



Guida per gli sviluppatori

Amazon Elastic Container Service



Amazon Elastic Container Service: Guida per gli sviluppatori

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cos'è Amazon ECS?	1
Terminologia e componenti	1
Funzionalità	3
Provisioning	3
Servizi correlati	4
Informazioni correlate	4
Nozioni di base	7
Configurazione	7
AWS Management Console	7
Iscriviti a un Account AWS	8
Crea un utente con accesso amministrativo	8
Crea un cloud privato virtuale	10
Creazione di un gruppo di sicurezza	10
Crea le credenziali per connetterti alla tua istanza EC2	14
Installa il AWS CLI	15
Fasi successive per l'utilizzo di Amazon ECS	16
Creazione di un'immagine di container	16
Prerequisiti	17
Creazione di un'immagine Docker	19
Invia l'immagine al registro del container di Amazon Elastic	21
Eliminazione	22
Passaggi successivi	23
Scopri come creare un'attività Linux per il tipo di lancio Fargate	23
Prerequisiti	23
Fase 1: Creazione del cluster	24
Fase 2: Creazione di una definizione di attività	25
Fase 3: Creazione del servizio	26
Fase 4: visualizzazione del servizio	27
Fase 5: rimozione	27
Scopri come creare un'attività Windows per il tipo di avvio Fargate	28
Prerequisiti	28
Fase 1: creazione di un cluster	29
Fase 2: Registrazione di una definizione di attività di Windows	30
Fase 3: creazione di un servizio con la definizione di attività	31

Fase 4: visualizzazione del servizio	32
Fase 5: pulizia	32
Scopri come creare un'attività Windows per il tipo di EC2 avvio	33
Prerequisiti	33
Fase 1: creazione di un cluster	34
Fase 2: Registrazione di una definizione di attività	36
Fase 3: creazione di un servizio	37
Fase 4: visualizzazione del servizio	37
Fase 5: pulizia	38
Utilizzando il AWS CDK	39
Fase 1: Configura il tuo progetto AWS CDK	40
Passaggio 2: utilizzare il AWS CDK per definire un server Web containerizzato su Fargate ...	42
Fase 3: Test del server Web	50
Fase 4: pulizia	50
Passaggi successivi	50
Creazione di risorse utilizzando la CLI di AWS Copilot	51
Installazione della CLI di AWS Copilot	52
Distribuzione di un'applicazione Amazon ECS di esempio utilizzando la AWS CLI Copilot	61
Amazon ECS con AWS CloudFormation	63
Modelli di esempio AWS CloudFormation	64
Definizioni di processi	64
Cluster	76
Servizi	89
Ruoli IAM per Amazon ECS	146
Creazione di risorse Amazon ECS utilizzando la console AWS CloudFormation	149
Prerequisiti	149
Fase 1: Creare un modello di stack	150
Fase 2: creare uno stack per le risorse Amazon ECS	155
Fase 3: Verifica	156
Fase 4: Eliminazione delle risorse	157
Creazione di risorse Amazon ECS utilizzando AWS CLI comandi per AWS CloudFormation	157
Prerequisiti	157
Fase 1: Creare uno stack	158
Fase 2: Verificare la creazione delle risorse	164
Fase 3: pulizia	165
Best practice	166

AWS Fargate	170
Procedure guidate	170
Provider di capacità	171
Definizioni di processi	171
Versioni della piattaforma	171
Bilanciamento del carico nel servizio	172
Parametri di utilizzo	172
Considerazioni sulla sicurezza relative all'utilizzo del tipo di lancio Fargate	173
Le migliori pratiche di sicurezza di Fargate	173
Utilizzato AWS KMS per crittografare lo storage temporaneo per Fargate	173
Funzionalità SYS_PTRACE per il tracciamento delle syscall del kernel con Fargate	174
Usa Amazon GuardDuty con Fargate Runtime Monitoring	174
Considerazioni sulla sicurezza di Fargate	175
Versioni della piattaforma Fargate	176
.....	176
Migrazione alla versione 1.4.0 della piattaforma Linux	177
Registro delle modifiche della versione della piattaforma Linux	178
Deprecazione della versione della piattaforma Linux	181
Registro delle modifiche della versione della piattaforma Windows	184
Contenitori Windows su Fargate: considerazioni per Amazon ECS	184
Archiviazione temporanea delle attività Fargate	185
Versioni della piattaforma container Linux Fargate	186
Versioni della piattaforma container Windows Fargate	187
Chiavi gestite dal cliente per lo storage temporaneo AWS Fargate	187
Ritiro e manutenzione delle attività	200
Panoramica dell'avviso di ritiro delle attività	202
Posso rinunciare al ritiro delle attività?	205
Posso ricevere notifiche di ritiro delle attività tramite altri servizi? AWS	205
Posso modificare il ritiro di un'attività dopo averla pianificata?	206
In che modo Amazon ECS gestisce le attività che fanno parte di un servizio?	206
Amazon ECS può gestire automaticamente le attività autonome?	206
Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS	206
AWS Regioni di Fargate	209
Contenitori Linux su AWS Fargate	209
Contenitori Windows su AWS Fargate	211
Progetta la tua soluzione per Amazon ECS	214

Capacità	215
Endpoint di servizio	216
Rete	216
Accesso alle funzionalità	217
Ruoli IAM	218
Registrazione	218
Tipi di avvio	219
Tipo di avvio di Fargate	219
EC2 tipo di lancio	224
Tipo di avvio esterno	226
Utilizzo degli endpoint dual-stack	227
Utilizzo degli endpoint dual-stack (AWS CLI)	228
Utilizzo degli endpoint dual-stack (AWS SDKs)	229
Utilizzo degli endpoint dual-stack dall'API REST	229
Applicazioni in sottoreti condivise, Local Zones e Wavelength Zones	230
Sottoreti condivise	230
Zone locali	232
Zone Wavelength	232
Amazon Elastic Container Service su AWS Outposts	233
Considerazioni	233
Prerequisiti	234
Panoramica della creazione di cluster su AWS Outposts	234
Ottimizza la capacità e la disponibilità	236
Massimizzazione della velocità di scalabilità	237
Gestione degli shock legati alla domanda	239
Best practice per la rete	241
Connect le applicazioni a Internet	241
Best practice per ricevere connessioni in entrata ad Amazon ECS	245
Le migliori pratiche per la connessione ai servizi AWS	250
Le migliori pratiche per connettere i servizi	253
Le migliori pratiche per i servizi di rete tra AWS account e VPCs	258
AWS servizi per la risoluzione dei problemi di rete	259
Accedi alle funzionalità con le impostazioni dell'account	260
Nomi di risorse Amazon (ARNs) e IDs	262
Sequenza temporale formato ARN e ID risorsa	264
Container Insights	264

AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)	266
Autorizzazione all'assegnazione di tag	267
Cronologia dell'autorizzazione all'assegnazione di tag	268
AWS Fargate tempo di attesa per il ritiro dell'attività	269
Aumenta le interfacce di rete delle istanze di container Linux	271
Monitoraggio del runtime (GuardDuty integrazione con Amazon)	271
VPC a doppio stack IPv6	272
Modalità predefinita del driver di registro	272
Visualizzazione delle impostazioni dell'account con la console	273
Modifica delle impostazioni dell'account	274
Tornare alle impostazioni predefinite dell'account	275
Gestione delle impostazioni dell'account utilizzando il AWS CLI	276
Ruoli IAM per Amazon ECS	278
Definizioni di processi	282
Stati delle definizioni di attività	283
Risorse Amazon ECS in grado di bloccare un'eliminazione	284
Progetta la tua applicazione	285
Le migliori pratiche per le immagini dei container	287
Le migliori pratiche per le dimensioni delle attività	289
Task networking per il tipo di EC2 lancio	290
Task networking per il tipo di lancio Fargate	302
Opzioni di archiviazione per le attività	306
Gestione dello spazio di memoria di swap dei container	394
Differenze nella definizione delle attività per il tipo di lancio di Fargate	396
Differenze nella definizione delle attività per le EC2 istanze che eseguono Windows	404
Creazione di una definizione di attività attraverso la nuova console	405
Convalida JSON	406
AWS CloudFormation pile	406
Procedura	406
Aggiornamento di una definizione di attività attraverso la console	439
Convalida JSON	440
Procedura	440
Annullamento della registrazione di una revisione di definizione dell'attività con la console	441
AWS CloudFormation pile	406
Procedura	442
Eliminazione di una revisione di definizione di attività attraverso la console	442

Risorse Amazon ECS in grado di bloccare un'eliminazione	284
Procedura	444
Casi d'uso delle definizioni di attività	444
Definizioni delle attività per i carichi di lavoro GPU	445
Definizioni delle attività per i carichi di lavoro di transcodifica video	455
Definizioni delle attività per i carichi di lavoro di machine learning di AWS Neuron	469
Definizioni delle attività per le istanze di deep learning	478
Definizioni delle attività per carichi di lavoro ARM a 64 bit	481
Invia i log a CloudWatch	483
Inviare i log a un servizio o AWS AWS Partner	487
Utilizzo di immagini non AWS containerizzate	500
Riavvia i singoli contenitori nelle attività	503
Passa dati sensibili a un contenitore	506
Parametri di definizione delle attività per il tipo di lancio Fargate	529
Family	529
Tipi di avvio	530
Ruolo del processo	530
Ruolo per l'esecuzione del processo	531
Modalità di rete	531
Piattaforma di runtime	532
Dimensioni processo	533
Definizioni del container	537
Nome dell'acceleratore di inferenza elastica	583
Configurazione del proxy	583
Volumi	585
Tag	592
Altri parametri di definizione di attività	593
Parametri di definizione delle attività per il tipo di EC2 avvio	596
Family	596
Tipi di avvio	597
Ruolo del processo	597
Ruolo per l'esecuzione del processo	598
Modalità di rete	598
Piattaforma di runtime	599
Dimensioni processo	600
Definizioni del container	602

Nome dell'acceleratore di inferenza elastica	652
Vincoli di posizionamento delle attività	652
Configurazione del proxy	653
Volumi	655
Tag	662
Altri parametri di definizione di attività	663
Modello di definizione di attività	665
Esempi di definizioni di attività	676
Server Web	677
splunkdriver di registro	679
fluentddriver di registro	680
gelfdriver di registro	680
Carichi di lavoro su istanze esterne	681
Immagine Amazon ECR e definizione delle attività, ruolo IAM	683
Punto di ingresso con comando	683
Dipendenze per i container	684
Volumi nelle definizioni delle attività	686
Definizioni di attività di esempio di Windows	686
Cluster	688
Provider di capacità	689
Cluster per Fargate	690
Avvisi di risoluzione di Fargate Spot	692
Creazione di un cluster per il tipo di lancio Fargate	694
Fornitori di capacità per il tipo di EC2 lancio	696
EC2 sicurezza delle istanze dei container	699
Creazione di un cluster per il tipo di EC2 lancio Amazon	699
Scalabilità automatica del cluster	705
Istanze di EC2 container Amazon	740
Cluster per il tipo di lancio esterno	880
Sistemi operativi e architetture di sistema supportati	881
Considerazioni	882
Creazione di un cluster per il tipo di avvio esterno	885
Registrazione di un'istanza esterna in un cluster Amazon ECS	888
Annullamento della registrazione di un'istanza esterna	893
Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS ..	899
Aggiornamento di un cluster	904

Eliminazione di un cluster	905
Annullamento della registrazione di un'istanza di container	906
Procedura	907
Esaurimento dell'istanza di container	908
Comportamento di svuotamento per i servizi	908
Comportamento di svuotamento per processi autonomi	909
Procedura	910
Agente del container	910
Ciclo di vita	911
AMI ottimizzate per Amazon ECS	912
Informazioni aggiuntive	912
Configurazione dell'agente del container	913
Installazione dell'agente del container Amazon ECS	915
Parametri di configurazione del registro dell'agente contenitore	921
Configurazione di istanze di container per immagini Docker private	925
Pulisci attività e immagini	930
Pianifica i tuoi contenitori	932
Opzioni di calcolo	934
Ciclo di vita del processo	935
Stati del ciclo di vita	936
In che modo Amazon ECS colloca le attività sulle istanze di container	938
EC2 tipo di lancio	938
Tipo di avvio di Fargate	939
Utilizza strategie per definire il posizionamento delle attività	940
Attività relative al gruppo	946
Definisci quali istanze di container vengono utilizzate per le attività	946
Attività autonome	958
Workflow delle attività	958
Ottimizza il tempo di avvio delle attività	959
Esecuzione di un'applicazione come attività	960
Utilizzo di Amazon EventBridge Scheduler per pianificare le attività	972
Interruzione di un'attività	980
Servizi	981
Strategia daemon	983
Strategia di replica	985
Ribilanciamento della zona di disponibilità	986

Creazione di un servizio	992
Aggiornamento di un servizio	1025
Aggiornamento di una distribuzione blu/verde	1045
Eliminazione di un servizio	1046
Implementazioni di aggiornamento continuo	1047
Distribuzioni blu/verde	1083
Implementazioni esterne	1103
Usa il bilanciamento del carico per distribuire il traffico di servizio	1111
Scalabilità automatica del servizio	1126
Servizi di interconnessione	1172
Protezione scalabile delle attività	1232
Iniezione dei guasti con Amazon ECS e Fargate	1240
Migrazione di un servizio con un ARN ridotto	1249
Logica di limitazione del servizio	1254
Parametri di definizione del servizio	1256
Applicazione di tag alle risorse	1292
Assegnazione di tag alle risorse	1293
Applicazione di tag alle risorse durante la creazione	1296
Restrizioni	1297
Tag gestiti da Amazon ECS	1297
Utilizza i tag per la fatturazione	1298
Aggiunta di tag alle risorse	1299
Aggiungere tag a un'istanza di contenitore	1301
Istanze di container esterni	1303
Report di utilizzo di	1303
Costi e utilizzo a livello di attività	1305
Monitoraggio	1307
Best practice per il monitoraggio di Amazon ECS	1308
Strumenti di monitoraggio	1308
Strumenti automatici	1308
Strumenti manuali	1310
Monitora Amazon ECS utilizzando CloudWatch	1311
Considerazioni	1312
Parametri consigliati	1312
Visualizzazione dei parametri Amazon ECS	1313
Metriche di Amazon ECS CloudWatch	1314

AWS Fargate metriche di utilizzo	1343
Metriche di prenotazione dei cluster Amazon ECS	1345
Metriche di utilizzo dei cluster Amazon ECS	1347
Metriche di utilizzo del servizio Amazon ECS	1349
Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge	1351
Eventi Amazon ECS	1352
Gestione degli eventi	1377
Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata	1381
Determina lo stato delle attività utilizzando i controlli dello stato dei container	1382
Come viene determinata la salute delle attività	1383
Controlli sanitari e disconnessioni degli agenti	1384
Visualizza lo stato del contenitore	1385
Monitora lo stato delle istanze dei container Amazon ECS	1385
Problemi di integrità dell'istanza del contenitore	1386
Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni	1387
Autorizzazioni IAM richieste per AWS Distro per l'integrazione con OpenTelemetry AWS X- Ray	1387
Specificare il AWS Distro for OpenTelemetry sidecar da AWS X-Ray integrare nella definizione dell'attività	1389
Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni	1390
Esportazione dei parametri delle applicazioni su Amazon CloudWatch	1391
Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus	1395
Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail	1399
Eventi di gestione di Amazon ECS in CloudTrail	1401
Esempi di eventi Amazon ECS	1401
Monitora i carichi di lavoro utilizzando i metadati	1403
Variabili di ambiente	1404
File di metadati del container	1405
Metadati delle attività disponibili per le attività di Amazon ECS su EC2	1412
Metadati delle attività disponibili per le attività su Fargate	1454
Introspezione dei contenitori	1478
Identifica i comportamenti non autorizzati utilizzando Runtime Monitoring	1481
Come funziona il monitoraggio del runtime con Amazon ECS	1482

Considerazioni	1483
Utilizzo delle risorse	1483
Monitoraggio del runtime per carichi di lavoro Fargate	1483
Monitoraggio del runtime per carichi di lavoro EC2	1487
Risoluzione dei problemi di monitoraggio del runtime	1492
Monitora i contenitori Amazon ECS con ECS Exec	1496
Considerazioni	1496
Prerequisiti	1499
Architettura	1499
Utilizzo di ECS Exec	1500
Registrazione tramite ECS Exec	1502
Utilizzo delle policy IAM per limitare l'accesso a ECS Exec	1506
Suggerimenti sul Compute Optimizer	1510
Suggerimenti sulle dimensioni delle attività per Fargate	1510
Risoluzione dei problemi	1511
Risolvi gli errori delle attività interrotte	1514
Interrotto l'aggiornamento dei messaggi di errore delle attività	1515
Visualizzazione degli errori delle attività interrotte	1520
Messaggi di errore relativi alle attività interrotte	1522
Verifica della connettività delle attività	1542
Visualizzazione delle richieste di ruoli IAM	1547
Visualizzazione dei messaggi relativi agli eventi di servizio	1548
Messaggi sugli eventi del servizio Amazon ECS	1549
Messaggi sugli eventi del servizio di ribilanciamento del servizio Amazon ECS Availability Zone	1560
Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS	1562
Risoluzione dei problemi relativi alla scalabilità automatica del servizio in Amazon ECS	1564
Risolvi gli errori di CPU o memoria non validi nella definizione delle attività	1564
Visualizzazione dei log degli agenti dei container	1567
Raccolta dei log dei container con Amazon ECS logs collector	1568
Introspezione degli agenti	1571
Diagnostica Docker in Amazon ECS	1573
Elenca i contenitori Docker in Amazon ECS	1573
Visualizza i log Docker in Amazon ECS	1574
Ispeziona i contenitori Docker in Amazon ECS	1575
Configurazione dell'output verboso dal demone Docker in Amazon ECS	1576

Risolvi i problemi relativi al Docker in Amazon API error (500): devmapper ECS	1578
Risolvi i problemi relativi a ECS Exec	1580
Verifica utilizzando Exec Checker	1580
Errore durante la chiamata a execute-command	1580
Risolvi i problemi relativi ad Amazon ECS Anywhere	1580
Problemi di registrazione delle istanze esterne	1581
Problemi di rete delle istanze esterne	1582
Problema di esecuzione dei processi	1582
AWS Fargate limitazione delle quote	1582
Limitazione dell'RunTaskAPI in Fargate	1583
Adeguamento delle quote tariffarie a Fargate	1584
Gestisci i problemi di limitazione	1584
Limitazione sincrona	1584
Throttling asincrono	1584
Monitora la limitazione	1585
CloudWatch Da utilizzare per monitorare la limitazione	1586
Motivi degli errori dell'API	1587
Sicurezza	1598
Identity and Access Management	1599
Destinatari	1599
Autenticazione con identità	1600
Gestione dell'accesso con policy	1604
Come funziona Amazon Elastic Container Service con IAM	1606
Esempi di policy basate su identità	1618
AWS politiche gestite per Amazon ECS	1630
Uso di ruoli collegati ai servizi	1642
Ruoli IAM per Amazon ECS	1646
Autorizzazioni necessarie per la console Amazon ECS	1704
Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS	1712
Assegnazione di tag alle risorse al momento della creazione	1713
Risoluzione dei problemi	1718
Best practice di IAM	1720
Registrazione di log e monitoraggio	1722
Convalida della conformità	1724
Migliori pratiche di conformità e sicurezza	1726
AWS Fargate Conformità FIPS-140	1728

AWS Fargate Considerazioni sullo standard FIPS-140	1728
Uso di FIPS su Fargate	1728
Uso CloudTrail per il controllo FIPS-140 di Fargate	1729
Sicurezza dell'infrastruttura	1730
Endpoint VPC di interfaccia (AWS PrivateLink)	1731
Modello di responsabilità condivisa	1737
Tipo di avvio di Fargate	1738
EC2 tipo di lancio	1739
Best practice per la sicurezza della rete	1740
Crittografia in transito	1740
Reti di attività	1741
AWS PrivateLink e Amazon ECS	1742
Impostazioni dell'agente Container	1743
Consigli sulla sicurezza della rete	1744
Best practice per la sicurezza di attività e contenitori	1745
Creazione di immagini di base o uso di immagini distroless	1745
Scansione delle immagini per individuare eventuali vulnerabilità	1747
Rimozione delle autorizzazioni speciali dalle immagini	1748
Creazione di un set di immagini curate	1748
Scansione dei pacchetti e delle librerie di applicazioni per individuare eventuali vulnerabilità	1747
Esecuzione dell'analisi statica del codice	1749
Esecuzione di container come utente non root	1749
Uso di un file system root di sola lettura	1750
Configura attività con limiti di CPU e memoria (Amazon EC2)	1750
Uso di tag immutabili con Amazon ECS	1751
Evita di utilizzare i container come privilegiati (Amazon EC2)	1751
Rimozione delle funzionalità di Linux non necessarie dal container	1751
Uso di una chiave gestita dal cliente (CMK) per la crittografia delle immagini inviate ad Amazon ECR	1752
Tutorial	1753
Creazione di un'attività Linux per il tipo di lancio Fargate con AWS CLI	1755
Prerequisiti	1756
Fase 1: creare un cluster	1757
Fase 2: Registra una definizione dei processi Linux	1758
Fase 3: Elenca le definizioni di attività	1759

Fase 4: Crea un servizio	1760
Fase 5: Elenca i servizi	1760
Fase 6: Descrivi il servizio in esecuzione	1761
Fase 7: Test	1763
Fase 8: elimina	1767
Creazione di un'attività Windows per il tipo di avvio Fargate con AWS CLI	1767
Prerequisiti	1768
Fase 1: creare un cluster	1769
Fase 2: Registrazione di una definizione di attività di Windows	1770
Fase 3: Elenca le definizioni di attività	1771
Fase 4: Crea un servizio	1771
Fase 5: Elenca i servizi	1772
Fase 6: Descrivi il servizio in esecuzione	1772
Fase 7: eliminare	1775
Creazione di un'attività per il tipo di EC2 avvio con AWS CLI	1775
Prerequisiti	1776
Fase 1: creare un cluster	1776
Fase 2: Avvia un'istanza con l'AMI Amazon ECS	1777
Fase 3: Elenca le istanze di container	1777
Fase 4: Descrivi la tua istanza di container	1778
Fase 5: Registra una definizione di attività	1781
Fase 6: Elenca le definizioni di attività	1782
Fase 7: Esegui un'attività	1783
Fase 8: Elenca le attività	1784
Fase 9: Descrivi l'attività in esecuzione	1784
Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events	1785
Prerequisito: configurazione di un cluster di verifica	1785
Fase 1: Creazione della funzione Lambda	1786
Fase 2: Registrazione di una regola di evento	1787
Fase 3: creazione di una definizione di attività	1787
Fase 4: Test della regola	1789
Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività ...	1790
Prerequisito: configurazione di un cluster di verifica	1790
Prerequisito: configurazione delle autorizzazioni per Amazon SNS	1790
Fase 1: Creazione e sottoscrizione a un argomento Amazon SNS	1790
Fase 2: Registrazione di una regola di evento	1791

Fase 3: Test del tuo articolo	1792
Concatenazione di messaggi di log multilinea o stack-trace	1794
Autorizzazioni IAM richieste	1794
Determinazione dei casi in cui utilizzare l'impostazione del log multilinea	1795
Opzioni di analisi e concatenazione	1797
Distribuzione di Fluent Bit su contenitori Windows	1816
Prerequisiti	1819
Fase 1: Creazione dei ruoli di accesso IAM	1820
Fase 2: Creazione di un'istanza di container Windows di Amazon ECS	1820
Fase 3: Configurazione di Fluent Bit	1822
Passaggio 4: Registrare una definizione di attività Windows Fluent Bit che indirizza i log a CloudWatch	1824
Fase 5: Esecuzione della definizione di attività <code>ecs-windows-fluent-bit</code> come servizio Amazon ECS utilizzando la strategia di pianificazione del daemon	1826
Fase 6: Registrazione di una definizione di attività di Windows che genera i log	1827
Fase 7: Esecuzione della definizione di attività <code>windows-app-task</code>	1828
Fase 8: Verificare gli accessi CloudWatch	1829
Fase 9: Pulizia	1830
Utilizzo gMSA per EC2 Linux container	1831
Considerazioni	1831
Prerequisiti	1832
Configurazione	1834
CredSpec file	1840
Utilizzo gMSA for Linux container su Fargate	1842
Considerazioni	1842
Prerequisiti	1842
Configurazione	1843
CredSpec file	1846
Utilizzo di contenitori Windows con domainless gMSA utilizzando il AWS CLI	1847
Prerequisiti	1848
Fase 1: Creare e configurare gMSA account su Active Directory Domain Services (AD DS)	1850
Fase 2: caricamento delle credenziali in Secrets Manager	1852
Passaggio 3: Modifica il tuo CredSpec JSON da includere senza dominio gMSA Informazioni	1852
Fase 4: Caricare CredSpec su Amazon S3	1854

Fase 5: creazione di un cluster Amazon ECS (facoltativo)	1854
Fase 6: creazione di un ruolo IAM per le istanze di container	1855
Fase 7: creazione di un ruolo di esecuzione dell'attività personalizzata	1855
Fase 8: creazione di un ruolo dell'attività per Amazon ECS Exec	1856
Fase 9: registrazione di una definizione di attività	1858
Fase 10: registrazione di un'istanza di container Windows	1860
Fase 11: verifica dell'istanza di container	1860
Fase 12: esecuzione di un'attività di Windows	1862
Passaggio 13: verifica che il contenitore abbia gMSA credenziali	1862
Fase 14: pulizia	1863
Debug	1864
Scopri come usare g MSAs per i contenitori EC2 Windows	1865
Considerazioni	1866
Prerequisiti	1867
Configurazione	1868
Utilizzo di Image Builder per creare modelli personalizzati ottimizzati per Amazon ECS AMIs ..	1874
Utilizzo dell'immagine ARN con infrastruttura come codice (IaC)	1875
Utilizzo dell'immagine ARN con AWS CloudFormation	1878
Utilizzo dell'immagine ARN con Terraform	1879
Utilizzo dei AWS Deep Learning Containers	1880
Quote del servizio	1881
Gestione delle quote di servizio in AWS Management Console	1881
Gestisci le quote di servizio e i limiti di limitazione delle API	1883
Sistema di bilanciamento del carico elastico	1884
Interfacce di rete elastiche	1885
AWS Cloud Map	1887
Documentazione di riferimento dell'API Amazon ECS	1889
Cronologia dei documenti	1890
.....	mcmxxxvi

Cos'è Amazon Elastic Container Service?

Amazon Elastic Container Service (Amazon ECS) è un servizio di orchestrazione di container completamente gestito che facilita l'implementazione, la gestione e il dimensionamento delle applicazioni containerizzate. Essendo un servizio completamente gestito, Amazon ECS include AWS configurazioni e best practice operative integrate. È integrato con entrambi AWS gli strumenti, come Amazon Elastic Container Registry, e con strumenti di terze parti, come Docker. Questa integrazione consente ai team di concentrarsi più facilmente sulla creazione delle applicazioni piuttosto che sull'ambiente. Puoi eseguire e scalare i carichi di lavoro dei container Regioni AWS nel cloud e in locale, senza la complessità della gestione di un piano di controllo.

Terminologia e componenti

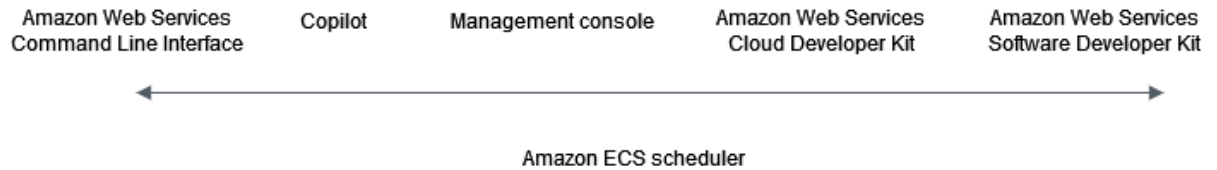
Amazon ECS si compone di tre livelli:

- Capacità: l'infrastruttura su cui vengono eseguiti i container
- Controller: per implementare e gestire le applicazioni in esecuzione sui container
- Provisioning: gli strumenti che puoi utilizzare per interfacciarti con il pianificatore al fine di implementare e gestire le applicazioni e i container

Il diagramma seguente mostra i livelli di Amazon ECS.

Amazon Elastic Container Service Layers

Provisioning



Controller



Capacity options



La capacità è l'infrastruttura in cui vengono eseguiti i container. Di seguito è riportata una panoramica delle opzioni di capacità:

- EC2 Istanze Amazon nel cloud AWS

Scegli il tipo di istanza e il numero di istanze per gestire la capacità.

- Serverless (AWS Fargate) nel cloud AWS

Fargate è un motore di elaborazione senza server. pay-as-you-go Con Fargate non è necessario gestire i server e la pianificazione della capacità o isolare i carichi di lavoro dei container per motivi di sicurezza.

- Macchine virtuali (VM) o server on-premise

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premise o una macchina virtuale (VM) nel cluster Amazon ECS.

Il pianificatore di Amazon ECS è il software che gestisce le applicazioni.

Funzionalità

Amazon ECS offre le seguenti funzionalità di alto livello:

Definizione di attività

Il modello per l'applicazione.

Cluster

L'infrastruttura su cui viene eseguita l'applicazione.

Attività

Un'applicazione, ad esempio un processo in batch, che esegue un lavoro e poi si interrompe.

Servizio

Un'applicazione stateless di lunga durata.

Impostazione dell'account

Consente l'accesso alle funzionalità.

Auto Scaling del cluster

Amazon ECS gestisce la scalabilità delle EC2 istanze Amazon registrate nel tuo cluster.

Auto Scaling dei servizi

Amazon ECS aumenta o diminuisce automaticamente il numero desiderato di attività nel tuo servizio.

Provisioning

Esistono diverse opzioni per il provisioning di Amazon ECS:

- AWS Management Console: fornisce un'interfaccia Web che può essere utilizzata per accedere alle risorse Amazon ECS.
- AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di AWS servizi, tra cui Amazon ECS. È supportata su Windows, Mac e Linux. Per ulteriori informazioni, consulta [AWS Command Line Interface](#).

- **AWS SDKs**— Fornisce informazioni specifiche per la lingua APIs e si occupa di molti dettagli di connessione. Questi includono il calcolo delle firme e la gestione di errori e di nuovi tentativi di richiesta. Per ulteriori informazioni, consulta [AWS SDKs](#).
- **AWS CDK**: offre un framework di sviluppo software open source che puoi utilizzare per modellare ed eseguire il provisioning delle risorse delle applicazioni cloud utilizzando linguaggi di programmazione familiari. Il AWS CDK effettua il provisioning delle tue risorse in modo sicuro e ripetibile tramite AWS CloudFormation.

Servizi correlati

Servizi da usare con Amazon ECS

Puoi usarne altri Servizi AWS per aiutarti a distribuire le tue attività e i tuoi servizi su Amazon ECS.

[Amazon EC2 Auto Scaling](#)

Ti aiuta a garantire il numero corretto di EC2 istanze Amazon disponibili per gestire il carico della tua applicazione.

[Amazon CloudWatch](#)

Monitora i tuoi servizi e le tue attività.

[Amazon Elastic Container Registry](#)

Archivia e gestisci le immagini dei container.

[Elastic Load Balancing](#)

Distribuisce automaticamente il traffico di servizio in entrata.

[Amazon GuardDuty](#)

Rileva l'uso potenzialmente non autorizzato o dannoso delle istanze e dei carichi di lavoro dei container.

Informazioni relative ad Amazon ECS

Le seguenti risorse correlate possono rivelarsi utili durante l'utilizzo di questo servizio.

- [AWS Fargate](#): panoramica delle funzionalità di Fargate.

- [Windows on AWS](#): panoramica di Windows su AWS carichi di lavoro e servizi.
- [Linux from AWS](#) - Portafoglio di moderni sistemi operativi basati su Linux di. AWS

Strumenti per sviluppatori

- [AWS App2Container](#): uno strumento a riga di comando per modernizzare le applicazioni.NET e Java in applicazioni containerizzate.
- [Strumenti per Amazon Web Services](#): AWS SDKs da utilizzare per gestire le risorse e le operazioni di Amazon ECS da una varietà di linguaggi di programmazione.
- [AWS CloudFormation](#)— Definisci e gestisci tutte le AWS risorse del tuo ambiente con una distribuzione automatizzata.
- [AWS Cloud Development Kit \(AWS CDK\)](#)— un framework Infrastructure-as-Code (IAC) che puoi utilizzare per definire l'infrastruttura AWS cloud utilizzando un linguaggio di programmazione di tua scelta.

Tutorial per sviluppatori

- [AWS Compute Blog](#): informazioni sulle nuove funzionalità e approfondimenti, esempi di codice e best practice.

AWS re:Post

- [AWS re:Post](#)— servizio AWS gestito di domande e risposte (Q & A) che offre risposte collettive e recensite da esperti alle tue domande tecniche.

Prezzi

- [Prezzi di Amazon ECS](#): informazioni sui prezzi di Amazon ECS.
- [AWS Fargate prezzi](#) — Informazioni sui prezzi di Fargate.

Risorse generali AWS

Le seguenti risorse generali possono aiutarti a lavorare con AWS.

- [Corsi e workshop](#): collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS

- [AWS Developer Center](#): esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- [AWS Strumenti per sviluppatori](#): collegamenti a strumenti di sviluppo SDKs, toolkit IDE e strumenti da riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- [Centro risorse introduttivo](#): scopri come configurare Account AWS, unirti alla AWS community e lanciare la tua prima applicazione.
- [Tutorial pratici: segui i tutorial](#) per avviare la step-by-step tua prima applicazione su. AWS
- [AWS Whitepaper](#): collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- [Supporto AWS Center](#): l'hub per la creazione e la gestione dei casi. Supporto AWS Include anche collegamenti ad altre risorse utili, come forum, informazioni tecniche FAQs, stato di salute del servizio e AWS Trusted Advisor.
- [Supporto](#)— La pagina web principale per informazioni su Supporto one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- [Contatti](#) - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- [AWS Termini del sito](#): informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Scopri come creare e utilizzare le risorse Amazon ECS

Le seguenti guide forniscono un'introduzione agli strumenti disponibili per accedere ad Amazon ECS e alle procedure introduttive per eseguire i container. Le nozioni di base su Docker consentono di eseguire le principali fasi di creazione di un'immagine del container Docker e del suo caricamento in un repository privato Amazon ECR. Le guide introduttive illustrano l'utilizzo dell'interfaccia a riga di comando di AWS Copilot e il AWS Management Console completamento delle attività comuni per eseguire i container su Amazon ECS e. AWS Fargate

Indice

- [Configurazione per l'uso di Amazon ECS](#)
- [Creazione di un'immagine di container da utilizzare su Amazon ECS](#)
- [Scopri come creare un'attività Amazon ECS Linux per il tipo di lancio Fargate](#)
- [Scopri come creare un'attività Amazon ECS Windows per il tipo di lancio Fargate](#)
- [Scopri come creare un'attività Amazon ECS Windows per il tipo di EC2 avvio](#)
- [Creazione di risorse Amazon ECS utilizzando AWS CDK](#)
- [Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot](#)

Configurazione per l'uso di Amazon ECS

Se ti sei già registrato ad Amazon Web Services (AWS) e utilizzi Amazon Elastic Compute Cloud (Amazon EC2), sarai quasi in grado di utilizzare Amazon ECS. Il processo di configurazione per i due servizi è molto simile. La seguente guida ti prepara per l'avvio del tuo primo cluster Amazon ECS.

Completa i seguenti processi per iniziare a usare Amazon ECS.

AWS Management Console

AWS Management Console È un'interfaccia basata su browser per la gestione delle risorse Amazon ECS. La console fornisce una panoramica visiva del servizio, semplificando l'esplorazione delle funzionalità e delle funzioni Amazon ECS senza dover utilizzare strumenti aggiuntivi. Sono disponibili numerosi tutorial e spiegazioni passo per passo correlate che possono guidare l'utente attraverso l'utilizzo della console.

Per un tutorial esplicativo della console, consulta [Scopri come creare e utilizzare le risorse Amazon ECS](#).

All'inizio, molti clienti preferiscono utilizzare la console perché fornisce un feedback visivo immediato sull'esito delle azioni intraprese. AWS i clienti che hanno familiarità con AWS Management Console, possono gestire facilmente le risorse correlate come i sistemi di bilanciamento del carico e le istanze Amazon EC2 .

Inizia con. AWS Management Console

Iscriviti a un Account AWS

Se non ne hai uno Account AWS, completa i seguenti passaggi per crearne uno.

Per iscriverti a un Account AWS

1. Apri la <https://portal.aws.amazon.com/billing/registrazione>.
2. Segui le istruzioni online.

Nel corso della procedura di registrazione riceverai una telefonata, durante la quale sarà necessario inserire un codice di verifica attraverso la tastiera del telefono.

Quando ti iscrivi a un Account AWS, Utente root dell'account AWS viene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come best practice di sicurezza, assegna l'accesso amministrativo a un utente e utilizza solo l'utente root per eseguire [attività che richiedono l'accesso di un utente root](#).

AWS ti invia un'email di conferma dopo il completamento della procedura di registrazione. In qualsiasi momento, puoi visualizzare l'attività corrente del tuo account e gestirlo accedendo a <https://aws.amazon.com/> e scegliendo Il mio account.

Crea un utente con accesso amministrativo

Dopo esserti registrato Account AWS, proteggi Utente root dell'account AWS AWS IAM Identity Center, abilita e crea un utente amministrativo in modo da non utilizzare l'utente root per le attività quotidiane.

Proteggi i tuoi Utente root dell'account AWS

1. Accedi [AWS Management Console](#) come proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

Per informazioni sull'accesso utilizzando un utente root, consulta la pagina [Signing in as the root user](#) della Guida per l'utente di Accedi ad AWS .

2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

Per istruzioni, consulta [Abilitare un dispositivo MFA virtuale per l'utente Account AWS root \(console\)](#) nella Guida per l'utente IAM.

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

Per istruzioni, consulta [Abilitazione di AWS IAM Identity Center](#) nella Guida per l'utente di AWS IAM Identity Center .

2. In IAM Identity Center, assegna l'accesso amministrativo a un utente.

Per un tutorial sull'utilizzo di IAM Identity Center directory come fonte di identità, consulta [Configurare l'accesso utente con l'impostazione predefinita IAM Identity Center directory](#) nella Guida per l'AWS IAM Identity Center utente.

Accesso come utente amministratore

- Per accedere con l'utente IAM Identity Center, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente IAM Identity Center.

Per informazioni sull'accesso utilizzando un utente IAM Identity Center, consulta [AWS Accedere al portale di accesso](#) nella Guida per l'Accedi ad AWS utente.

Assegna l'accesso a ulteriori utenti

1. In IAM Identity Center, crea un set di autorizzazioni conforme alla best practice dell'applicazione di autorizzazioni con il privilegio minimo.

Segui le istruzioni riportate nella pagina [Creazione di un set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .

2. Assegna al gruppo prima gli utenti e poi l'accesso con autenticazione unica (Single Sign-On).

Per istruzioni, consulta [Aggiungere gruppi](#) nella Guida per l'utente di AWS IAM Identity Center .

Crea un cloud privato virtuale

Puoi usare Amazon Virtual Private Cloud (Amazon VPC) per lanciare AWS risorse in una rete virtuale che hai definito. È consigliabile avviare le istanze di container in un VPC.

Se è presente un VPC di default, puoi ignorare questa sezione e passare all'operazione successiva, [Creazione di un gruppo di sicurezza](#). Per determinare se disponi di un VPC predefinito, consulta [Lavora con il tuo VPC predefinito e le sottoreti predefinite nella Amazon VPC](#) User Guide. In caso contrario, puoi creare un VPC non predefinito nell'account utilizzando le fasi seguenti.

Per informazioni su come creare un VPC, consulta [Create a VPC nella Amazon VPC](#) User Guide e usa la tabella seguente per determinare quali opzioni selezionare.

Opzione	Valore	
Risorse da creare	Solo VPC	
Nome	Se lo desideri, puoi fornire un nome per il VPC.	
IPv4 blocco CIDR	IPv4 Inserimento manuale CIDR La dimensione del blocco CIDR deve essere compresa tra /16 e /28.	
IPv6 blocco CIDR	Nessun blocco IPv6 CIDR	
Tenancy	Predefinita	

Per ulteriori informazioni sul servizio Amazon VPC, consulta [Cos'è Amazon VPC?](#) nella Guida per l'utente di Amazon VPC.

Creazione di un gruppo di sicurezza

I gruppi di sicurezza fungono da firewall per le istanze di container, controllando sia il traffico in entrata che quello in uscita a livello di istanza di container. È possibile aggiungere regole a un gruppo di sicurezza che consentono di connettersi all'istanza di container dall'indirizzo IP tramite SSH. È

inoltre possibile aggiungere regole che consentono il traffico HTTP e HTTPS in entrata e in uscita da qualsiasi posizione. Aggiungi le regole per aprire le porte necessarie per le attività. Le istanze di container richiedono un accesso di rete esterno per comunicare con l'endpoint del servizio Amazon ECS.

Se prevedi di avviare istanze di container in più regioni, devi creare un gruppo di sicurezza in ogni regione. Per ulteriori informazioni, consulta [Regioni e zone di disponibilità](#) nella Amazon EC2 User Guide.

 Tip

È necessario l'indirizzo IP pubblico del computer locale, che puoi ottenere usando un servizio. Ad esempio, forniamo il seguente servizio: <http://checkip.amazonaws.com/> o <https://checkip.amazonaws.com/>. Per individuare un altro servizio che fornisce l'indirizzo IP, utilizza la frase di ricerca "qual è il mio indirizzo IP". Se ti stai connettendo tramite un provider di servizi Internet (ISP) o da un firewall senza un indirizzo IP statico, è necessario individuare l'intervallo di indirizzi IP utilizzati dai computer client.

Per informazioni su come creare un gruppo di sicurezza, consulta [Creare un gruppo di sicurezza per la tua EC2 istanza Amazon](#) nella Amazon EC2 User Guide e usa la tabella seguente per determinare quali opzioni selezionare.

Opzione	Valore	
Regione	La stessa regione in cui è stata creata la coppia di chiavi.	
Nome	Un nome facile da ricordare , ad esempio ecs-instances-default-cluster.	
VPC	Il VPC predefinito (contrassegnato con "(predefinito)").	

 Note

Se il tuo account supporta Amazon EC2

Opzione	Valore	
	Classic, seleziona il VPC che hai creato nell'attività precedente.	

Per informazioni sulle regole in uscita da aggiungere per i tuoi casi d'uso, consulta [le regole dei gruppi di sicurezza per diversi casi d'uso](#) nella Amazon EC2 User Guide.

Le istanze di container Amazon ECS non richiedono l'apertura delle porte in entrata. Tuttavia, puoi aggiungere una regola SSH per poter accedere all'istanza di container ed esaminare le attività con comandi Docker. Puoi anche aggiungere regole per HTTP e HTTPS se desideri che l'istanza di container ospiti un'attività che esegue un server Web. Le istanze di container richiedono accesso di rete esterno per comunicare con l'endpoint del servizio Amazon ECS. Completa le fasi seguenti per aggiungere queste regole facoltative per i gruppi di sicurezza.

Aggiungi le seguenti tre regole in entrata al tuo gruppo di sicurezza. Per informazioni su come creare un gruppo di sicurezza, consulta [Configure security group rules](#) nella Amazon EC2 User Guide.

Opzione	Valore	
Regola HTTP	Tipo: HTTP Origine: Qualsiasi (0.0.0.0/0) Questa opzione aggiunge automaticamente il blocco CIDR IPv4 0.0.0.0/0 come origine. Questo è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, è preferibile autorizzare l'accesso a un'istanza solo di	

Opzione	Valore	
	un determinato indirizzo IP o a di un intervallo di indirizzi.	
Regola HTTPS	Tipo: HTTPS Origine: Qualsiasi (0.0.0.0/0) Questo è accettabile per un breve periodo di tempo in un ambiente di test, ma non è sicuro per gli ambienti di produzione. In produzione, è preferibile autorizzare l'accesso a un'istanza solo di un determinato indirizzo IP o a di un intervallo di indirizzi.	

Opzione	Valore	
Regola SSH	Tipo: SSH Origine: Personalizzato, specifica l'indirizzo IP pubblico del computer o della rete in base alla notazione CIDR. Per specificare un singolo indirizzo IP in notazione CIDR, aggiungi il prefisso di routing /32. Se, ad esempio, l'indirizzo IP è 203.0.113.25 , specifica 203.0.113.25/32 . Se l'azienda alloca gli indirizzi da un intervallo, specificare l'intero intervallo, ad esempio 203.0.113.0/24 .	
	 Important Per motivi di sicurezza , non è consigliabile consentire l'accesso SSH da tutti gli indirizzi IP (0.0.0.0/0) all'istanza, se non per scopi di test e solo per un breve intervallo di tempo.	

Crea le credenziali per connetterti alla tua istanza EC2

Per Amazon ECS, una coppia di key pair è necessaria solo se intendi utilizzare il tipo di EC2 avvio.

AWS utilizza la crittografia a chiave pubblica per proteggere le informazioni di accesso dell'istanza. Un'istanza di Linux, ad esempio un'istanza di container Amazon ECS, non prevede password per l'accesso a SSH. Viene utilizzata una coppia di chiavi per accedere in modo sicuro all'istanza. Devi specificare il nome della coppia di chiavi quando avvii l'istanza di container, quindi fornisci la chiave privata quando accedi utilizzando SSH.

Se non hai già creato una key pair, puoi crearne una utilizzando la EC2 console Amazon. Se prevedi di avviare istanze in più regioni, devi creare una coppia di chiavi in ogni regione. Per ulteriori informazioni sulle regioni, consulta [Regioni e zone di disponibilità](#) nella Amazon EC2 User Guide.

Per creare una coppia di chiavi

- Usa la EC2 console Amazon per creare una key pair. Per ulteriori informazioni sulla creazione di una coppia di chiavi, consulta [Create a key pair](#) nella Amazon EC2 User Guide.

Per informazioni su come connetterti alla tua istanza, consulta [Connect to your Linux instance](#) nella Amazon EC2 User Guide.

Installa il AWS CLI

AWS Management Console Può essere utilizzato per gestire tutte le operazioni manualmente con Amazon ECS. Tuttavia, puoi installarlo sul tuo desktop locale o AWS CLI su un box per sviluppatori in modo da poter creare script in grado di automatizzare le attività di gestione comuni in Amazon ECS.

Per utilizzarlo AWS CLI con Amazon ECS, installa la AWS CLI versione più recente. Per informazioni sull'installazione AWS CLI o sull'aggiornamento alla versione più recente, consulta [Installazione o aggiornamento all'ultima versione di AWS CLI nella Guida per l'utente](#). AWS Command Line Interface

AWS Command Line Interface (AWS CLI) è uno strumento unificato che puoi utilizzare per gestire i tuoi servizi. AWS Solo con questo strumento, puoi controllare più AWS servizi e automatizzarli tramite script. I comandi Amazon ECS in AWS CLI sono un riflesso dell'API Amazon ECS.

AWS CLI È adatto ai clienti che preferiscono e sono abituati a creare script e interfacciarsi con uno strumento da riga di comando e fanno esattamente quali azioni vogliono eseguire sulle proprie risorse Amazon ECS. AWS CLI È utile anche per i clienti che desiderano familiarizzare con Amazon ECS. APIs I clienti possono AWS CLI utilizzarlo per eseguire una serie di operazioni sulle risorse Amazon ECS, tra cui le operazioni di creazione, lettura, aggiornamento ed eliminazione, direttamente dall'interfaccia a riga di comando.

Utilizza il AWS CLI se conosci o desideri acquisire familiarità con Amazon ECS APIs e i comandi CLI corrispondenti e desideri scrivere script automatici ed eseguire azioni specifiche sulle risorse Amazon ECS.

AWS fornisce anche gli strumenti da riga di comando. [AWS Tools for Windows PowerShell](#) Per ulteriori informazioni, consulta la [AWS Tools for Windows PowerShell Guida per l'utente di](#).

Fasi successive per l'utilizzo di Amazon ECS

Dopo aver installato AWS CLI, ci sono molti strumenti diversi che puoi utilizzare mentre continui a utilizzare Amazon ECS. I seguenti link spiegano quali sono alcuni di questi strumenti e forniscono esempi di come utilizzarli con Amazon ECS.

- [Crea la tua prima immagine di contenitore](#) con Docker e inviala ad Amazon ECR per utilizzarla nelle definizioni delle attività di Amazon ECS.
- [Scopri come creare un'attività Amazon ECS Linux per il tipo di lancio Fargate](#).
- [Scopri come creare un'attività Amazon ECS Windows per il tipo di lancio Fargate](#).
- [Scopri come creare un'attività Amazon ECS Windows per il tipo di EC2 avvio](#).
- Usando il linguaggio di programmazione preferito, definisci l'infrastruttura o l'architettura come codice con l'opzione [Creazione di risorse Amazon ECS utilizzando AWS CDK](#).
- Definisci e gestisci tutte le AWS risorse del tuo ambiente con la distribuzione automatizzata utilizzando. [Utilizzo di Amazon ECS con AWS CloudFormation](#)
- Utilizza il flusso di lavoro completo [Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot](#) end-to-end degli sviluppatori per creare, rilasciare e gestire applicazioni container conformi alle AWS migliori pratiche per l'infrastruttura.

Creazione di un'immagine di container da utilizzare su Amazon ECS

Amazon ECS utilizza le immagini Docker nelle definizioni di attività per avviare i container. Docker è una tecnologia che fornisce gli strumenti per costruire, eseguire, testare e implementare applicazioni distribuite basate su container.

Lo scopo dei passaggi descritti qui è guidarti nella creazione della tua prima immagine Docker e nel suo invio ad Amazon ECR, che è un registro del container, per utilizzarla nelle definizioni delle attività Amazon ECS. Questa procedura guidata presuppone una conoscenza di base di Docker e del

suo funzionamento. Per ulteriori informazioni su Docker, vedi [Cos'è Docker?](#) e la [documentazione Docker](#).

Prerequisiti

Prima di iniziare, verifica che siano soddisfatti i seguenti requisiti preliminari:

- Assicurati di aver completato le fasi di configurazione di Amazon ECR. Per ulteriori informazioni, consulta [Spostare un'immagine attraverso il suo ciclo di vita in Amazon ECR nella Amazon Elastic Container Registry User Guide](#).
- L'utente dispone delle autorizzazioni IAM richieste per accedere al servizio Amazon ECR. Per ulteriori informazioni, consulta [Policy gestite da Amazon ECR](#).
- Docker è installato. Per le fasi di installazione di Docker per Amazon Linux 2, consulta la sezione [Installazione di Docker su 023 AL2](#). Per tutti gli altri sistemi operativi, consulta la documentazione su Docker all'indirizzo [Panoramica di Docker Desktop](#).
- Hai AWS CLI installato e configurato il. Per ulteriori informazioni, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

Se non disponi o non hai bisogno di un ambiente di sviluppo locale e preferisci utilizzare un' EC2 istanza Amazon per usare Docker, forniamo i seguenti passaggi per avviare un' EC2 istanza Amazon utilizzando Amazon Linux 2 e installare Docker Engine e Docker CLI.

Installazione di Docker su 023 AL2

Docker è disponibile per diversi sistemi operativi, compresa la maggior parte delle distribuzioni Linux, ad esempio Ubuntu, e persino per macOS e Windows. Per ulteriori informazioni sull'installazione di Docker sul tuo specifico sistema operativo, consulta la [guida all'installazione di Docker](#).

Per l'utilizzo di Docker non è necessario un sistema di sviluppo locale. Se utilizzi EC2 già Amazon, puoi avviare un'istanza Amazon Linux 2023 e installare Docker per iniziare.

Se hai già installato Docker, passa a [Creazione di un'immagine Docker](#).

Per installare Docker su un' EC2 istanza Amazon utilizzando un'AMI Amazon Linux 2023

1. Avvia un'istanza con l'ultima AMI Amazon Linux 2023. Per ulteriori informazioni, consulta [Avviare un' EC2 istanza utilizzando la procedura guidata di avvio dell'istanza nella console](#) nella Amazon EC2 User Guide.

2. Connettiti alla tua istanza. Per ulteriori informazioni, consulta [Connect to your EC2 instance](#) nella Amazon EC2 User Guide.
3. Aggiorna i pacchetti installati e la cache dei pacchetti sulla tua istanza.

```
sudo yum update -y
```

4. Installa il pacchetto Docker Community Edition più recente.

```
sudo yum install docker
```

5. Avvia il servizio Docker.

```
sudo service docker start
```

6. Aggiungi `ec2-user` al gruppo `docker` in modo da poter eseguire comandi Docker senza utilizzare `sudo`.

```
sudo usermod -a -G docker ec2-user
```

7. Esci e ripeti l'accesso per trovare il nuovo gruppo di autorizzazioni `docker`. A questo scopo, puoi chiudere la finestra del terminale SSH corrente e riconnetterti all'istanza in una nuova finestra. La nuova sessione SSH avrà le autorizzazioni del gruppo `docker` appropriate.
8. Verifica che `ec2-user` possa eseguire i comandi Docker senza `sudo`.

```
docker info
```

Note

In alcuni casi, l'assegnazione delle autorizzazioni necessarie a `ec2-user` per accedere al daemon Docker può richiedere il riavvio dell'istanza. Prova a riavviare l'istanza se visualizzi questo errore:

```
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
```


Creazione di un'immagine Docker

Le definizioni di processo di Amazon ECS utilizzano le immagini Docker per avviare i container sulle istanze di container nei tuoi cluster. In questa sezione, crei un'immagine Docker di una semplice applicazione Web e la testi sul tuo sistema locale o su un' EC2 istanza Amazon, quindi invii l'immagine al registro dei contenitori Amazon ECR in modo da poterla utilizzare in una definizione di attività Amazon ECS.

Per creare un'immagine Docker di una semplice applicazione Web

1. Crea un file denominato `Dockerfile`. Un Dockerfile è un file manifest che descrive l'immagine di base da utilizzare per l'immagine Docker, nonché gli elementi da installare ed eseguire su di essa. Per ulteriori informazioni sui Dockerfile, consulta la [documentazione di riferimento sui Dockerfile](#).

```
touch Dockerfile
```

2. Modifica il Dockerfile appena creato e aggiungi i seguenti contenuti.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Update installed packages and install Apache
RUN yum update -y && \
    yum install -y httpd

# Write hello world message
RUN echo 'Hello World!' > /var/www/html/index.html

# Configure Apache
RUN echo 'mkdir -p /var/run/httpd' >> /root/run_apache.sh && \
    echo 'mkdir -p /var/lock/httpd' >> /root/run_apache.sh && \
    echo '/usr/sbin/httpd -D FOREGROUND' >> /root/run_apache.sh && \
    chmod 755 /root/run_apache.sh

EXPOSE 80

CMD /root/run_apache.sh
```

Questo Dockerfile utilizza l'immagine pubblica di Amazon Linux 2 ospitata su Amazon ECR Public. Le istruzioni RUN aggiornano le cache dei pacchetti, installano alcuni pacchetti software per il server Web e infine scrivono il contenuto "Hello World!" nella root del documento del

server Web. L'EXPOSEistruzione indica che la porta 80 sul contenitore è quella in ascolto e l'CMDistruzione avvia il server Web.

3. Crea l'immagine Docker dal tuo Dockerfile.

 Note

In alcune versioni di Docker, il seguente comando potrebbe richiedere il percorso completo al Dockerfile anziché il percorso relativo mostrato di seguito.

Se esegui il comando su macOS 64" M1/Mx-chip, Use the --platform option "--platform linux/amd.

```
docker build -t hello-world .
```

4. Elenca l'immagine del container.

```
docker images --filter reference=hello-world
```

Output:

REPOSITORY SIZE	TAG	IMAGE ID	CREATED
hello-world 194MB	latest	e9ffedc8c286	4 minutes ago

5. Esegui l'immagine appena creata. L'opzione -p 80:80 mappa la porta 80 esposta sul container alla porta 80 sul sistema host.

```
docker run -t -i -p 80:80 hello-world
```

 Note

L'output dal server Web Apache viene visualizzato nella finestra del terminale. Puoi ignorare il messaggio "Could not reliably determine the fully qualified domain name".

6. Apri un browser e accedi al server su cui è in esecuzione Docker e che ospita il tuo container.

- Se stai usando un' EC2 istanza, questo è il valore DNS pubblico per il server, che è lo stesso indirizzo che usi per connetterti all'istanza con SSH. Assicurati che il gruppo di sicurezza per l'istanza consenta il traffico in entrata sulla porta 80.
- Se Docker è in esecuzione in locale, accedi con il browser a <http://localhost/>.
- Se lo utilizzi docker-machine su un computer Windows o Mac, trova l'indirizzo IP della macchina VirtualBox virtuale che ospita Docker con il docker-machine ip comando, sostituendolo *machine-name* con il nome della macchina docker che stai utilizzando.

```
docker-machine ip machine-name
```

Visualizzerai una pagina Web con il tuo contenuto "Hello World!" dichiarazione.

7. Interrompi il container Docker digitando Ctrl+c.

Invia l'immagine al registro del container di Amazon Elastic

Amazon ECR è un servizio di registro AWS Docker gestito. Puoi utilizzare la CLI di Docker per inviare, estrarre e gestire le immagini nei tuoi repository Amazon ECR. Per i dettagli sui prodotti Amazon ECR, i case study dei clienti in evidenza e FAQs, consulta le [pagine dei dettagli del prodotto Amazon Elastic Container Registry](#).

Come aggiungere un tag all'immagine e inviarla ad Amazon ECR

1. Crea un repository Amazon ECR per archiviare l'immagine hello-world. Annota il valore repositoryUri nell'output.

regionSostituiscilo, ad esempio Regione AWS, con il tuo. us-east-1

```
aws ecr create-repository --repository-name hello-repository --region region
```

Output:

```
{
  "repository": {
    "registryId": "aws_account_id",
    "repositoryName": "hello-repository",
```

```
"repositoryArn": "arn:aws:ecr:region:aws_account_id:repository/hello-  
repository",  
  "createdAt": 1505337806.0,  
  "repositoryUri": "aws_account_id.dkr.ecr.region.amazonaws.com/hello-  
repository"  
}
```

2. Applica un tag all'immagine hello-world con il valore repositoryUri ricavato nella fase precedente.

```
docker tag hello-world aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

3. Esegui il comando `aws ecr get-login-password`. Specificare l'URI del registro in cui si desidera eseguire l'autenticazione. Per maggiori informazioni, consulta [Autorizzazioni del registro](#) nella Guida per l'utente di Amazon Elastic Container Registry.

```
aws ecr get-login-password --region region | docker login --username AWS --  
password-stdin aws_account_id.dkr.ecr.region.amazonaws.com
```

Output:

```
Login Succeeded
```

Important

Se viene visualizzato un errore, installare o eseguire l'upgrade alla versione più recente dell' AWS CLI. Per ulteriori informazioni, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente](#).

4. Invia l'immagine ad Amazon ECR con il valore repositoryUri dalla fase precedente.

```
docker push aws_account_id.dkr.ecr.region.amazonaws.com/hello-repository
```

Eliminazione

Per continuare con la creazione di una definizione di attività di Amazon ECS e l'avvio di un'attività con l'immagine del container, vai al passaggio [Passaggi successivi](#). Una volta completato l'esperimento

con l'immagine Amazon ECR, puoi eliminare il repository in modo che non ti siano addebitati costi per l'archiviazione dell'immagine.

```
aws ecr delete-repository --repository-name hello-repository --region region --force
```

Passaggi successivi

Le definizioni di attività richiedono un ruolo per l'esecuzione dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Dopo aver creato e inviato l'immagine del contenitore ad Amazon ECR, puoi utilizzare quell'immagine in una definizione di attività. Per ulteriori informazioni, consultare uno dei seguenti argomenti:

- [the section called “Scopri come creare un'attività Linux per il tipo di lancio Fargate”](#)
- [the section called “Scopri come creare un'attività Windows per il tipo di avvio Fargate”](#)
- [Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI](#)

Scopri come creare un'attività Amazon ECS Linux per il tipo di lancio Fargate

Amazon Elastic Container Service (Amazon ECS) è un servizio di gestione dei container rapido e altamente dimensionabile che semplifica l'esecuzione, l'arresto e la gestione dei container. Puoi ospitare i container in un'infrastruttura serverless gestita da Amazon ECS avviando i servizi o i processi su AWS Fargate. Per ulteriori informazioni su Fargate, vedere [AWS Fargate per Amazon ECS](#)

Inizia a usare Amazon ECS AWS Fargate utilizzando il tipo di lancio Fargate per le tue attività nelle regioni in cui Amazon ECS AWS supporta Fargate.

Per cominciare a utilizzare Amazon ECS su AWS Fargate, completa le fasi descritte di seguito.

Prerequisiti

Prima di iniziare, completa i passaggi indicati [Configurazione per l'uso di Amazon ECS](#) e assicurati che il tuo utente IAM disponga delle autorizzazioni specificate nell'esempio di policy IAM.

AdministratorAccess

La console tenta di creare automaticamente il ruolo IAM per l'esecuzione dei processi, obbligatorio per le attività Fargate. Per garantire che la console sia in grado di creare questo ruolo IAM, uno dei seguenti requisiti deve essere true:

- L'utente ha accesso di amministratore. Per ulteriori informazioni, consulta [Configurazione per l'uso di Amazon ECS](#).
- L'utente dispone delle autorizzazioni IAM per creare un ruolo di servizio. Per ulteriori informazioni, consulta [Creazione di un ruolo per delegare le autorizzazioni a un servizio](#). AWS
- Un utente con accesso di amministratore ha creato manualmente il ruolo di esecuzione delle attività in modo che sia disponibile nell'account da essere utilizzato. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Important

Il gruppo di sicurezza selezionato durante la creazione di un servizio con la definizione delle attività deve avere la porta 80 aperta per il traffico in entrata. Aggiungi la seguente regola in entrata al gruppo di sicurezza. Per informazioni su come creare un gruppo di sicurezza, consulta [Creare un gruppo di sicurezza per la tua EC2 istanza Amazon](#) nella Amazon EC2 User Guide.

- Type (Tipo): HTTP
- Protocol (Protocollo): TCP
- Intervallo porte: 80
- Origine: Qualsiasi (0.0.0.0/0)

Fase 1: Creazione del cluster

Crea un cluster che utilizzi il VPC predefinito.

Prima di iniziare, assegna l'autorizzazione IAM appropriata. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#).

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.

4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Cluster configuration (Configurazione del cluster), inserisci un nome univoco in Cluster name (Nome del cluster).

Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.

6. (Facoltativo) Per attivare Container Insights, espandi Monitoring (Monitoraggio) e poi attiva Use Container Insights (Usa Container Insights).
7. (Facoltativo) Per identificare il tuo cluster, espandi la sezione Tags (Tag), quindi configura i tag.

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:

- In Chiave, immetti il nome della chiave.
- In Valore, immetti il valore della chiave.

[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.

8. Scegli Create (Crea).

Fase 2: Creazione di una definizione di attività

Una definizione di attività è come un modello per la tua applicazione. Ogni volta che avvii un processo in Amazon ECS, specifichi una definizione di attività. Il servizio quindi sa quale immagine Docker utilizzare per i container, quanti container utilizzare nell'attività e l'allocazione delle risorse per ciascun container.

1. Nel riquadro di navigazione, scegli Definizioni di attività.
2. Scegli Create new Task Definition (Crea nuova definizione di attività), Create new revision with JSON (Crea nuova revisione con JSON).
3. Copia e incolla la seguente definizione di attività di esempio nella casella, quindi scegli Save (Salva).

```
{
  "family": "sample-fargate",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "fargate-app",
      "image": "public.ecr.aws/docker/library/httpd:latest",
```

```

        "portMappings": [
            {
                "containerPort": 80,
                "hostPort": 80,
                "protocol": "tcp"
            }
        ],
        "essential": true,
        "entryPoint": [
            "sh",
            "-c"
        ],
        "command": [
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</
h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html &&
httpd-foreground\""
        ]
    },
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "256",
    "memory": "512"
}

```

4. Scegli Create (Crea).

Fase 3: Creazione del servizio

Crea un servizio utilizzando la definizione di attività.

1. Nel riquadro di navigazione scegli Clusters (Cluster), quindi seleziona il cluster creato in [Fase 1: Creazione del cluster](#).
2. Nella scheda Services (Servizi), scegli Create (Crea).
3. In Deployment configuration (Configurazione dell'implementazione), specifica come viene implementata l'applicazione.

- a. In Task Definition (Definizione di attività), scegli la definizione di attività creata in [Fase 2: Creazione di una definizione di attività](#).
 - b. In Service name (Nome servizio), specifica un nome per il servizio.
 - c. Per Desired tasks (Attività desiderate), immetti 1.
4. In Reti, puoi creare un nuovo gruppo di sicurezza o sceglierne uno esistente per l'attività. Assicurati che il gruppo di sicurezza utilizzato contenga la regola in entrata elencata in [Prerequisiti](#).
 5. Scegli Create (Crea).

Fase 4: visualizzazione del servizio

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Scegli il cluster in cui è stato eseguito il servizio.
4. Nella scheda Servizi, in Nome del servizio scegli il servizio creato in [Fase 3: Creazione del servizio](#).
5. Scegli la scheda Attività, quindi scegli l'attività nel tuo servizio.
6. Nella pagina dell'attività, nella sezione Configurazione, per IP pubblico, scegli Apri indirizzo.

Fase 5: rimozione

Dopo aver finito di utilizzare un cluster Amazon ECS, dovresti eliminare le risorse associate in modo da evitare costi aggiuntivi per le risorse non utilizzate.

Alcune risorse Amazon ECS, come i processi, i servizi, i cluster e le istanze di container, vengono eliminate dalla console Amazon ECS. Altre risorse, come EC2 istanze Amazon, sistemi di bilanciamento del carico Elastic Load Balancing e gruppi Auto Scaling, devono essere pulite manualmente nella console EC2 Amazon o eliminando lo stack che le ha create. AWS CloudFormation

1. Nel pannello di navigazione scegliere Clusters (Cluster).
2. Nella pagina Cluster, seleziona il cluster creato per questo tutorial.
3. Scegli la scheda Servizi.

4. Seleziona il servizio, quindi scegli Elimina.
5. Alla richiesta di conferma, immetti delete (elimina), quindi scegli Delete (Elimina). In alternativa, puoi utilizzare l'opzione Force delete che consente ad Amazon ECS di ridimensionare il servizio per tuo conto prima di eliminarlo.

Attendi che il servizio venga eliminato.

6. Scegli Elimina Cluster. Alla richiesta di conferma, inserisci delete **cluster-name**, quindi scegli Elimina. L'eliminazione del cluster pulisce le risorse associate che sono state create con il cluster, inclusi i gruppi di Auto Scaling o i sistemi di bilanciamento del VPCs carico.

Scopri come creare un'attività Amazon ECS Windows per il tipo di lancio Fargate

Inizia a usare Amazon ECS AWS Fargate utilizzando il tipo di lancio Fargate per le tue attività nelle regioni in cui Amazon ECS AWS supporta Fargate.

Per cominciare a utilizzare Amazon ECS su AWS Fargate, completa le fasi descritte di seguito.

Prerequisiti

Prima di iniziare, completa i passaggi indicati [Configurazione per l'uso di Amazon ECS](#) e assicurati che il tuo utente IAM disponga delle autorizzazioni specificate nell'esempio di policy IAM.

AdministratorAccess

La console tenta di creare automaticamente il ruolo IAM per l'esecuzione dei processi, obbligatorio per le attività Fargate. Per garantire che la console sia in grado di creare questo ruolo IAM, uno dei seguenti requisiti deve essere true:

- L'utente ha accesso di amministratore. Per ulteriori informazioni, consulta [Configurazione per l'uso di Amazon ECS](#).
- L'utente dispone delle autorizzazioni IAM per creare un ruolo di servizio. Per ulteriori informazioni, consulta [Creazione di un ruolo per delegare le autorizzazioni a un servizio](#). AWS
- Un utente con accesso di amministratore ha creato manualmente il ruolo di esecuzione delle attività in modo che sia disponibile nell'account da essere utilizzato. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

⚠ Important

Il gruppo di sicurezza selezionato durante la creazione di un servizio con la definizione delle attività deve avere la porta 80 aperta per il traffico in entrata. Aggiungi la seguente regola in entrata al gruppo di sicurezza. Per informazioni su come creare un gruppo di sicurezza, consulta [Creare un gruppo di sicurezza per la tua EC2 istanza Amazon](#) nella Amazon EC2 User Guide.

- Type (Tipo): HTTP
- Protocol (Protocollo): TCP
- Intervallo porte: 80
- Origine: Qualsiasi (0.0.0.0/0)

Fase 1: creazione di un cluster

Puoi creare un nuovo cluster denominato windows che utilizzi il VPC predefinito.

Per creare un cluster con AWS Management Console

1. Aprire la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Cluster configuration (Configurazione del cluster), in Cluster name (Nome del cluster) immetti windows.
6. (Facoltativo) Per attivare Container Insights, espandi Monitoring (Monitoraggio) e poi attiva Use Container Insights (Usa Container Insights).
7. (Facoltativo) Per identificare il tuo cluster, espandi la sezione Tags (Tag), quindi configura i tag.

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:

- In Chiave, immetti il nome della chiave.
- In Valore, immetti il valore della chiave.

[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.

8. Scegli Create (Crea).

Fase 2: Registrazione di una definizione di attività di Windows

Prima di eseguire i container Windows nel cluster Amazon ECS, devi registrare una definizione di attività. Il seguente esempio di definizione di attività mostra una semplice pagina Web sulla porta 8080 di un'istanza di container con l'immagine del container `mcr.microsoft.com/windows/servercore/iis`.

Per registrare la definizione di attività di esempio con AWS Management Console

1. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
2. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).
3. Copia e incolla la seguente definizione di attività di esempio nella casella, quindi scegli Save (Salva).

```
{
  "containerDefinitions": [
    {
      "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html
-Type file -Value '<html> <head> <title>Amazon ECS Sample App</title>
<style>body {margin-top: 40px; background-color: #333;} </style> </head><body>
<div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
<h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p>'; C:\\ServiceMonitor.exe w3svc"],
      "entryPoint": [
        "powershell",
        "-Command"
      ],
      "essential": true,
      "cpu": 2048,
      "memory": 4096,
      "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
      "name": "sample_windows_app",
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80,
          "protocol": "tcp"
        }
      ]
    }
  ]
}
```

```
        }
    ]
},
{
    "memory": "4096",
    "cpu": "2048",
    "networkMode": "awsvpc",
    "family": "windows-simple-iis-2019-core",
    "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
    "runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
    "requiresCompatibilities": ["FARGATE"]
}
```

4. Verifica le informazioni e scegli Crea.

Fase 3: creazione di un servizio con la definizione di attività

Dopo aver registrato la definizione di attività, tramite di essa puoi posizionare le attività nel cluster. La seguente procedura crea un servizio con la definizione di attività e posiziona un'attività nel cluster.

Per creare un servizio dalla definizione di attività con la console

1. Nel riquadro di navigazione scegli Clusters (Cluster), quindi seleziona il cluster creato in [Fase 1: creazione di un cluster](#).
2. Nella scheda Services (Servizi), scegli Create (Crea).
3. In Deployment configuration (Configurazione dell'implementazione), specifica come viene implementata l'applicazione.
 - a. In Task Definition (Definizione di attività), scegli la definizione di attività creata in [Fase 2: Registrazione di una definizione di attività di Windows](#).
 - b. In Service name (Nome servizio), specifica un nome per il servizio.
 - c. Per Desired tasks (Attività desiderate), immetti 1.
4. In Reti, puoi creare un gruppo di sicurezza o sceglierne uno esistente. Assicurati che il gruppo di sicurezza utilizzato contenga la regola in entrata elencata in [Prerequisiti](#).
5. Scegli Create (Crea).

Fase 4: visualizzazione del servizio

Una volta che il servizio ha lanciato un'attività nel cluster, puoi visualizzare il servizio e aprire la pagina di test di IIS in un browser per verificare che il container sia in esecuzione.

Note

Il download dell'istanza di container e l'estrazione dei livelli base del container Windows da parte di tale istanza possono richiedere fino a 15 minuti.

Come visualizzare il servizio

1. Aprire la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Scegli il cluster in cui è stato eseguito il servizio.
4. Nella scheda Servizi, in Nome del servizio scegli il servizio creato in [Fase 3: creazione di un servizio con la definizione di attività](#).
5. Scegli la scheda Attività, quindi scegli l'attività nel tuo servizio.
6. Nella pagina dell'attività, nella sezione Configurazione, per IP pubblico, scegli Apri indirizzo.

Fase 5: pulizia

Dopo aver finito di utilizzare un cluster Amazon ECS, dovresti eliminare le risorse associate in modo da evitare costi aggiuntivi per le risorse non utilizzate.

Alcune risorse Amazon ECS, come i processi, i servizi, i cluster e le istanze di container, vengono eliminate dalla console Amazon ECS. Altre risorse, come EC2 istanze Amazon, sistemi di bilanciamento del carico Elastic Load Balancing e gruppi Auto Scaling, devono essere pulite manualmente nella console EC2 Amazon o eliminando lo stack che le ha create. AWS CloudFormation

1. Nel pannello di navigazione scegliere Clusters (Cluster).
2. Nella pagina Cluster, seleziona il cluster creato per questo tutorial.
3. Scegli la scheda Servizi.
4. Seleziona il servizio, quindi scegli Elimina.

5. Alla richiesta di conferma, immetti delete (elimina), quindi scegli Delete (Elimina).

Attendi che il servizio venga eliminato.

6. Scegli Elimina Cluster. Alla richiesta di conferma, inserisci delete, quindi scegli Elimina.

cluster-name L'eliminazione del cluster pulisce le risorse associate che sono state create con il cluster, inclusi i gruppi di Auto Scaling o i sistemi di bilanciamento del VPCs carico.

Scopri come creare un'attività Amazon ECS Windows per il tipo di EC2 avvio

Inizia a usare Amazon ECS utilizzando il tipo di EC2 avvio registrando una definizione di attività, creando un cluster e creando un servizio nella console.

Completa i seguenti passaggi per iniziare a usare Amazon ECS utilizzando il tipo di EC2 avvio.

Prerequisiti

Prima di iniziare, completa i passaggi indicati [Configurazione per l'uso di Amazon ECS](#) e assicurati che il tuo utente IAM disponga delle autorizzazioni specificate nell'esempio di policy AdministratorAccess IAM.

La console tenta di creare automaticamente il ruolo IAM per l'esecuzione dei processi, obbligatorio per le attività Fargate. Per garantire che la console sia in grado di creare questo ruolo IAM, uno dei seguenti requisiti deve essere true:

- L'utente ha accesso di amministratore. Per ulteriori informazioni, consulta [Configurazione per l'uso di Amazon ECS](#).
- L'utente dispone delle autorizzazioni IAM per creare un ruolo di servizio. Per ulteriori informazioni, consulta [Creazione di un ruolo per delegare le autorizzazioni a un servizio](#). AWS
- Un utente con accesso di amministratore ha creato manualmente il ruolo di esecuzione delle attività in modo che sia disponibile nell'account da essere utilizzato. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Important

Il gruppo di sicurezza selezionato durante la creazione di un servizio con la definizione delle attività deve avere la porta 80 aperta per il traffico in entrata. Aggiungi la seguente regola

in entrata al gruppo di sicurezza. Per informazioni su come creare un gruppo di sicurezza, consulta [Creare un gruppo di sicurezza per la tua EC2 istanza Amazon](#) nella Amazon EC2 User Guide.

- Type (Tipo): HTTP
- Protocol (Protocollo): TCP
- Intervallo porte: 80
- Origine: Qualsiasi (0.0.0.0/0)

Fase 1: creazione di un cluster

Un cluster Amazon ECS è un raggruppamento logico di attività, servizi e istanze di container.

I passaggi seguenti illustrano la creazione di un cluster con un' EC2 istanza Amazon registrata che ci consentirà di eseguire un'attività su di esso. Se non completi un campo specifico, lascia il valore predefinito utilizzato dalla console.

Creazione di un nuovo cluster (console Amazon ECS)

Prima di iniziare, assegna l'autorizzazione IAM appropriata. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#).

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Cluster configuration (Configurazione del cluster), inserisci un nome univoco in Cluster name (Nome del cluster).

Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.

6. (Facoltativo) Per modificare il VPC e le sottoreti in cui vengono avviati i processi e i servizi, in Networking (Reti), esegui una qualunque di queste operazioni:
 - Per rimuovere una sottorete, in Subnets (Sottoreti), scegli X per ogni sottorete da rimuovere.
 - Per passare a un VPC diverso da quello di default, in VPC, scegli un VPC esistente, poi in Subnets (Sottoreti), seleziona ciascuna sottorete.

7. Per aggiungere EC2 istanze Amazon al tuo cluster, espandi l'infrastruttura, quindi seleziona EC2 istanze Amazon. Successivamente, configura il gruppo Auto Scaling che funge da provider di capacità:
 - a. Per utilizzare un gruppo Auto Scaling esistente, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona il gruppo.
 - b. Per creare un gruppo Auto Scaling, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona Create new group (Crea nuovo gruppo) e quindi fornisci i seguenti dettagli sul gruppo:
 - Per Operating system/Architecture (Sistema operativo/architettura), scegli l'AMI ottimizzata per Amazon ECS per le istanze del gruppo Auto Scaling.
 - Ad EC2 esempio, scegli il tipo di istanza per i tuoi carichi di lavoro. Per ulteriori informazioni sui diversi tipi di istanze, consulta [Amazon EC2 Instances](#).

La scalabilità gestita funziona meglio se il gruppo Auto Scaling utilizza tipi di istanza uguali o simili.

- In SSH key pair (Coppia di chiavi SSH), scegli la coppia che dimostra la tua identità quando ti connetti all'istanza.
 - In Capacity (Capacità), inserisci il numero minimo e massimo di istanze da avviare nel gruppo Auto Scaling. EC2 Le istanze Amazon comportano dei costi finché sono disponibili nelle tue risorse. AWS Per ulteriori informazioni, consulta la pagina [EC2 dei prezzi di Amazon](#).
8. (Facoltativo) Per attivare Container Insights, espandi Monitoring (Monitoraggio) e poi attiva Use Container Insights (Usa Container Insights).
 9. (Facoltativo) Per gestire i tag cluster, espandi Tags (Tag), quindi esegui una delle seguenti operazioni:

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:

 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.

[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.
 10. Scegli Create (Crea).

Fase 2: Registrazione di una definizione di attività

Per registrare la definizione di attività di esempio con AWS Management Console

1. Nel riquadro di navigazione, scegli Definizioni di attività.
2. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).
3. Copia e incolla la seguente definizione di attività di esempio nella casella, quindi scegli Salva.

```
{
  "containerDefinitions": [
    {
      "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html
-Type file -Value '<html> <head> <title>Amazon ECS Sample App</title>
<style>body {margin-top: 40px; background-color: #333;} </style> </head><body>
<div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
<h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p>'; C:\\ServiceMonitor.exe w3svc"],
      "entryPoint": [
        "powershell",
        "-Command"
      ],
      "essential": true,
      "cpu": 2048,
      "memory": 4096,
      "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
      "name": "sample_windows_app",
      "portMappings": [
        {
          "hostPort": 443,
          "containerPort": 80,
          "protocol": "tcp"
        }
      ]
    }
  ],
  "memory": "4096",
  "cpu": "2048",
  "family": "windows-simple-iis-2019-core",
  "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
  "runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
}
```

```
"requiresCompatibilities": ["EC2"]
}
```

4. Verifica le informazioni e scegli Crea.

Fase 3: creazione di un servizio

Un servizio Amazon ECS consente di eseguire e mantenere simultaneamente un determinato numero di istanze di una definizione di attività in un cluster Amazon ECS. Se uno dei processi ha esito negativo o viene interrotto per un qualsiasi motivo, il pianificatore del servizio Amazon ECS lancia un'altra istanza della definizione di attività per sostituirla e mantenere il numero desiderato di processi nel servizio. Per ulteriori informazioni sui servizi, consulta [Servizi Amazon ECS](#).

Come creare un servizio

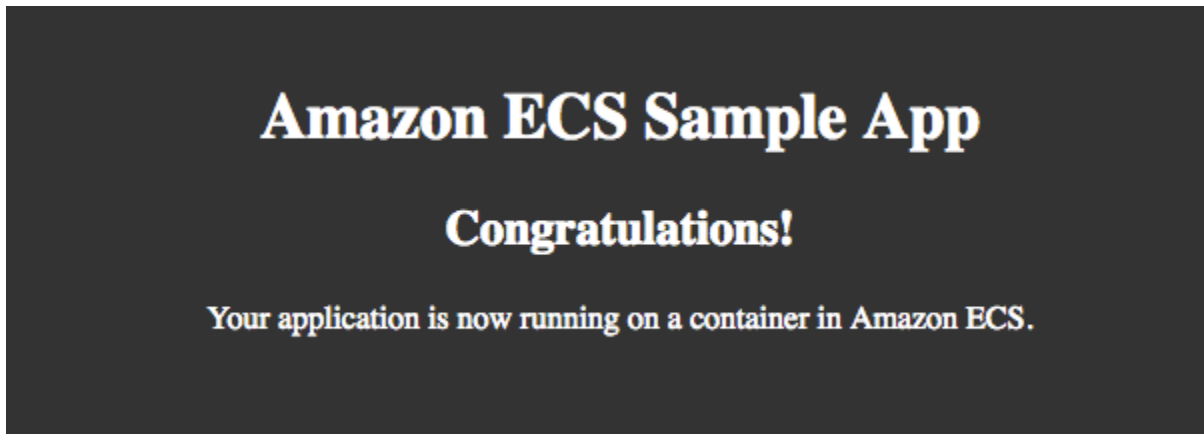
1. Nel pannello di navigazione scegliere Clusters (Cluster).
2. Seleziona il cluster creato in [Fase 1: creazione di un cluster](#).
3. Nella scheda Servizi, scegli Crea.
4. Nella sezione Environment (Ambiente), completa le operazioni descritte di seguito.
 - a. Per Compute options (Opzioni di calcolo), scegli Launch type (Tipo di avvio).
 - b. Per Tipo di avvio, selezionare EC2
5. Nella sezione Deployment configuration (Configurazione implementazione), procedi come segue.
 - a. In Family (Famiglia), scegli la definizione di attività creata in [Fase 2: Registrazione di una definizione di attività](#).
 - b. In Service name (Nome servizio), specifica un nome per il servizio.
 - c. Per Desired tasks (Attività desiderate), immetti 1.
6. Controlla le opzioni e seleziona Crea.
7. Scegli Visualizza servizio per rivedere il servizio.

Fase 4: visualizzazione del servizio

Il servizio è un'applicazione basata sul Web che consente di visualizzare i relativi container con un browser Web.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.

2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Scegli il cluster in cui è stato eseguito il servizio.
4. Nella scheda Servizi, in Nome del servizio scegli il servizio creato in [Fase 3: creazione di un servizio](#).
5. Scegli la scheda Attività, quindi scegli l'attività nel tuo servizio.
6. Nella pagina dell'attività, nella sezione Configurazione, per IP pubblico, scegli Apri indirizzo. La schermata riportata di seguito è l'output previsto.



Fase 5: pulizia

Dopo aver finito di utilizzare un cluster Amazon ECS, dovresti eliminare le risorse associate in modo da evitare costi aggiuntivi per le risorse non utilizzate.

Alcune risorse Amazon ECS, come i processi, i servizi, i cluster e le istanze di container, vengono eliminate dalla console Amazon ECS. Altre risorse, come EC2 istanze Amazon, sistemi di bilanciamento del carico Elastic Load Balancing e gruppi Auto Scaling, devono essere pulite manualmente nella console EC2 Amazon o eliminando lo stack che le ha create. AWS CloudFormation

1. Nel pannello di navigazione scegliere Clusters (Cluster).
2. Nella pagina Cluster, seleziona il cluster creato per questo tutorial.
3. Scegli la scheda Servizi.
4. Seleziona il servizio, quindi scegli Elimina.
5. Alla richiesta di conferma, immetti delete (elimina), quindi scegli Delete (Elimina).

Attendi che il servizio venga eliminato.

6. Scegli Elimina Cluster. Alla richiesta di conferma, inserisci delete, quindi scegli Elimina.

cluster-name L'eliminazione del cluster pulisce le risorse associate che sono state create con il cluster, inclusi i gruppi di Auto Scaling o i sistemi di bilanciamento del VPCs carico.

Creazione di risorse Amazon ECS utilizzando AWS CDK

AWS Cloud Development Kit (AWS CDK) È un framework Infrastructure-as-Code (IAC) che puoi utilizzare per definire l'infrastruttura AWS cloud utilizzando un linguaggio di programmazione a tua scelta. Per definire la tua infrastruttura cloud, devi innanzitutto scrivere un'app (in uno dei linguaggi supportati dal CDK) contenente uno o più stack. Quindi, lo sintetizzi in un AWS CloudFormation modello e distribuisce le tue risorse sul tuo Account AWS. Segui i passaggi descritti in questo argomento per distribuire un server Web containerizzato con Amazon Elastic Container Service (Amazon ECS) e su Fargate. AWS CDK

La AWS Construct Library, inclusa nel CDK, fornisce moduli che puoi usare per modellare le risorse che forniscono. Servizi AWS Per i servizi più diffusi, la libreria fornisce costrutti curati con impostazioni predefinite intelligenti e best practice. Uno di questi moduli, [aws-ecs-patterns](#), fornisce astrazioni di alto livello che possono essere utilizzate per definire il servizio containerizzato e tutte le risorse di supporto necessarie in poche righe di codice.

In questo argomento viene utilizzato il costrutto [ApplicationLoadBalancedFargateService](#). Questo costrutto implementa un servizio Amazon ECS su Fargate dietro un Application Load Balancer. Il `aws-ecs-patterns` modulo include anche costrutti che utilizzano un sistema di bilanciamento del carico di rete e vengono eseguiti su Amazon. EC2

Prima di iniziare questa attività, configura il tuo ambiente di AWS CDK sviluppo e installalo AWS CDK eseguendo il comando seguente. Per istruzioni su come configurare l'ambiente di AWS CDK sviluppo, consulta [Getting Started With the AWS CDK - Prerequisiti](#).

```
npm install -g aws-cdk
```

Note

Queste istruzioni presuppongono che tu stia usando AWS CDK v2.

Argomenti

- [Fase 1: Configura il tuo progetto AWS CDK](#)

- [Passaggio 2: utilizzare il AWS CDK per definire un server Web containerizzato su Fargate](#)
- [Fase 3: Test del server Web](#)
- [Fase 4: pulizia](#)
- [Passaggi successivi](#)

Fase 1: Configura il tuo progetto AWS CDK

Crea una directory per la tua nuova AWS CDK app e inizializza il progetto.

TypeScript

```
mkdir hello-ecs
cd hello-ecs
cdk init --language typescript
```

JavaScript

```
mkdir hello-ecs
cd hello-ecs
cdk init --language javascript
```

Python

```
mkdir hello-ecs
cd hello-ecs
cdk init --language python
```

Dopo l'avvio del progetto, attiva l'ambiente virtuale del progetto e installa le dipendenze AWS CDK di base.

```
source .venv/bin/activate
python -m pip install -r requirements.txt
```

Java

```
mkdir hello-ecs
cd hello-ecs
cdk init --language java
```

Importa questo progetto Maven nel tuo IDE Java. Ad esempio, in Eclipse, usa File > Import (Importa) > Maven > Existing Maven Projects (Progetti Maven esistenti).

C#

```
mkdir hello-ecs
cd hello-ecs
cdk init --language csharp
```

Go

```
mkdir hello-ecs
cd hello-ecs
cdk init --language go
```

Note

Il modello di AWS CDK applicazione utilizza il nome della directory del progetto per generare nomi per i file e le classi di origine. In questo esempio, la directory è denominata hello-ecs. Se utilizzi un nome di directory del progetto diverso, l'app non corrisponderà a queste istruzioni.

AWS CDK v2 include costrutti stabili per tutti Servizi AWS in un unico pacchetto chiamato. `aws-cdk-lib` Questo pacchetto viene installato come dipendenza quando iniziizzi il progetto. Quando lavori con determinati linguaggi di programmazione, il pacchetto viene installato quando crei il progetto per la prima volta. In questo argomento viene descritto come utilizzare un costrutto Amazon ECS Patterns, che fornisce astrazioni di alto livello per lavorare con Amazon ECS. Questo modulo si basa sui costrutti Amazon ECS e su altri costrutti per il provisioning delle risorse necessarie per l'applicazione Amazon ECS.

I nomi utilizzati per importare queste librerie nell'applicazione CDK potrebbero differire leggermente a seconda del linguaggio di programmazione utilizzato. Come riferimento, ecco i nomi utilizzati in ogni linguaggio di programmazione CDK supportato.

TypeScript

```
aws-cdk-lib/aws-ecs
aws-cdk-lib/aws-ecs-patterns
```

JavaScript

```
aws-cdk-lib/aws-ecs  
aws-cdk-lib/aws-ecs-patterns
```

Python

```
aws_cdk.aws_ecs  
aws_cdk.aws_ecs_patterns
```

Java

```
software.amazon.awscdk.services.ecs  
software.amazon.awscdk.services.ecs.patterns
```

C#

```
Amazon.CDK.AWS.ECS  
Amazon.CDK.AWS.ECS.Patterns
```

Go

```
github.com/aws/aws-cdk-go/awscdk/v2/awsecs  
github.com/aws/aws-cdk-go/awscdk/v2/awsecspatterns
```

Passaggio 2: utilizzare il AWS CDK per definire un server Web containerizzato su Fargate

Usa l'immagine del contenitore. [amazon-ecs-sample](#) Questa immagine contiene un'app web PHP che funziona su Nginx.

Nel AWS CDK progetto che avete creato, modificate il file che contiene la definizione dello stack in modo che sia simile a uno degli esempi seguenti.

Note

Uno stack è un'unità di implementazione. Tutte le risorse devono trovarsi in uno stack e tutte le risorse in uno stack vengono implementate contemporaneamente. Se una risorsa non viene implementata, verrà eseguito il rollback di tutte le altre risorse già implementate. Un'

AWS CDK app può contenere più stack e le risorse in uno stack possono fare riferimento a risorse in un altro stack.

TypeScript

Aggiorna `lib/hello-ecs-stack.ts` in modo che assomigli a quanto segue.

```
import * as cdk from 'aws-cdk-lib';
import { Construct } from 'constructs';
import * as ecs from 'aws-cdk-lib/aws-ecs';
import * as ecsp from 'aws-cdk-lib/aws-ecs-patterns';

export class HelloEcsStack extends cdk.Stack {
  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    new ecsp.ApplicationLoadBalancedFargateService(this, 'MyWebServer', {
      taskImageOptions: {
        image: ecs.ContainerImage.fromRegistry('public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest'),
      },
      publicLoadBalancer: true
    });
  }
}
```

JavaScript

Aggiorna `lib/hello-ecs-stack.js` in modo che assomigli a quanto segue.

```
const cdk = require('aws-cdk-lib');
const { Construct } = require('constructs');
const ecs = require('aws-cdk-lib/aws-ecs');
const ecsp = require('aws-cdk-lib/aws-ecs-patterns');

class HelloEcsStack extends cdk.Stack {
  constructor(scope = Construct, id = string, props = cdk.StackProps) {
    super(scope, id, props);

    new ecsp.ApplicationLoadBalancedFargateService(this, 'MyWebServer', {
      taskImageOptions: {
        image: ecs.ContainerImage.fromRegistry('amazon/amazon-ecs-sample'),
      },
    });
  }
}
```

```
    },  
    publicLoadBalancer: true  
  });  
}  
}  
  
module.exports = { HelloEcsStack }
```

Python

Aggiorna `hello-ecs/hello_ecs_stack.py` in modo che assomigli a quanto segue.

```
import aws_cdk as cdk  
from constructs import Construct  
  
import aws_cdk.aws_ecs as ecs  
import aws_cdk.aws_ecs_patterns as ecsp  
  
class HelloEcsStack(cdk.Stack):  
  
    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:  
        super().__init__(scope, construct_id, **kwargs)  
  
        ecsp.ApplicationLoadBalancedFargateService(self, "MyWebServer",  
            task_image_options=ecsp.ApplicationLoadBalancedTaskImageOptions(  
                image=ecs.ContainerImage.from_registry("amazon/amazon-ecs-sample")),  
                public_load_balancer=True  
            )
```

Java

Aggiorna `src/main/java/com.myorg/HelloEcsStack.java` in modo che assomigli a quanto segue.

```
package com.myorg;  
  
import software.constructs.Construct;  
import software.amazon.awscdk.Stack;  
import software.amazon.awscdk.StackProps;  
  
import software.amazon.awscdk.services.ecs.ContainerImage;  
import  
    software.amazon.awscdk.services.ecs.patterns.ApplicationLoadBalancedFargateService;
```

```

import
software.amazon.awscdk.services.ecs.patterns.ApplicationLoadBalancedTaskImageOptions;

public class HelloEcsStack extends Stack {
    public HelloEcsStack(final Construct scope, final String id) {
        this(scope, id, null);
    }

    public HelloEcsStack(final Construct scope, final String id, final StackProps
props) {
        super(scope, id, props);

        ApplicationLoadBalancedFargateService.Builder.create(this, "MyWebServer")
            .taskImageOptions(ApplicationLoadBalancedTaskImageOptions.builder()
                .image(ContainerImage.fromRegistry("amazon/amazon-ecs-sample"))
                .build())
            .publicLoadBalancer(true)
            .build();
    }
}

```

C#

Aggiorna `src/HelloEcs/HelloEcsStack.cs` in modo che assomigli a quanto segue.

```

using Amazon.CDK;
using Constructs;
using Amazon.CDK.AWS.ECS;
using Amazon.CDK.AWS.ECS.Patterns;
namespace HelloEcs
{
    public class HelloEcsStack : Stack
    {
        internal HelloEcsStack(Construct scope, string id, IStackProps props =
null) : base(scope, id, props)
        {
            new ApplicationLoadBalancedFargateService(this, "MyWebServer",
                new ApplicationLoadBalancedFargateServiceProps
                {
                    TaskImageOptions = new ApplicationLoadBalancedTaskImageOptions
                    {
                        Image = ContainerImage.FromRegistry("amazon/amazon-ecs-
sample")
                    },

```

```

        PublicLoadBalancer = true
    });
}
}
}

```

Go

Aggiorna `hello-ecs.go` in modo che assomigli a quanto segue.

```

package main

import (
    "github.com/aws/aws-cdk-go/awscdk/v2"
    // "github.com/aws/aws-cdk-go/awscdk/v2/awssqs"
    "github.com/aws/aws-cdk-go/awscdk/v2/awsecs"
    "github.com/aws/aws-cdk-go/awscdk/v2/awsecspatterns"
    "github.com/aws/constructs-go/constructs/v10"
    "github.com/aws/jsii-runtime-go"
)

type HelloEcsStackProps struct {
    awscdk.StackProps
}

func NewHelloEcsStack(scope constructs.Construct, id string, props
    *HelloEcsStackProps) awscdk.Stack {
    var sprops awscdk.StackProps
    if props != nil {
        sprops = props.StackProps
    }
    stack := awscdk.NewStack(scope, &id, &sprops)

    // The code that defines your stack goes here

    // example resource
    // queue := awssqs.NewQueue(stack, jsii.String("HelloEcsQueue"),
    &awssqs.QueueProps{
    // VisibilityTimeout: awscdk.Duration_Seconds(jsii.Number(300)),
    // })
    res := awsecspatterns.NewApplicationLoadBalancedFargateService(stack,
    jsii.String("MyWebServer"),
    &awsecspatterns.ApplicationLoadBalancedFargateServiceProps{
        TaskImageOptions: &awsecspatterns.ApplicationLoadBalancedTaskImageOptions{

```

```

    Image: awsecs.ContainerImage_FromRegistry(jsii.String("amazon/amazon-ecs-
sample"), &awsecs.RepositoryImageProps{}),
    },
    },
)
awscdk.NewCfnOutput(stack, jsii.String("LoadBalancerDNS"),
&awscdk.CfnOutputProps{Value: res.LoadBalancer().LoadBalancerDnsName()})

return stack
}

func main() {
defer jsii.Close()

app := awscdk.NewApp(nil)

NewHelloEcsStack(app, "HelloEcsStack", &HelloEcsStackProps{
    awscdk.StackProps{
        Env: env(),
    },
})

app.Synth(nil)
}

// env determines the AWS environment (account+region) in which our stack is to
// be deployed. For more information see: https://docs.aws.amazon.com/cdk/latest/guide/environments.html
func env() *awscdk.Environment {
    // If unspecified, this stack will be "environment-agnostic".
    // Account/Region-dependent features and context lookups will not work, but a
    // single synthesized template can be deployed anywhere.
    //-----
    return nil

    // Uncomment if you know exactly what account and region you want to deploy
    // the stack to. This is the recommendation for production stacks.
    //-----
    // return &awscdk.Environment{
    //     Account: jsii.String("123456789012"),
    //     Region:  jsii.String("us-east-1"),
    // }

    // Uncomment to specialize this stack for the AWS Account and Region that are

```

```
// implied by the current CLI configuration. This is recommended for dev
// stacks.
//-----
// return &awsdk.Environment{
//   Account: jsii.String(os.Getenv("CDK_DEFAULT_ACCOUNT")),
//   Region:  jsii.String(os.Getenv("CDK_DEFAULT_REGION")),
// }
}
```

Il frammento breve precedente include quanto segue:

- Il nome logico del servizio, `MyWebServer`.
- L'immagine del contenitore ottenuta dalla Amazon ECR Public Gallery: `amazon/amazon-ecs-sample`.
- Altre informazioni pertinenti, ad esempio il fatto che il load balancer ha un indirizzo pubblico ed è accessibile da Internet.

AWS CDK Creerà tutte le risorse necessarie per distribuire il server Web, incluse le seguenti risorse. Queste risorse sono state omesse in questo esempio.

- Cluster Amazon ECS
- Amazon VPC e istanze Amazon EC2
- Gruppo con scalabilità automatica
- Application Load Balancer
- Ruoli IAM e policy

Alcune risorse con provisioning automatico verranno condivise da tutti i servizi Amazon ECS definiti nello stack.

Salva il file di origine, quindi esegui il comando `cdk synth` nella directory principale dell'applicazione. AWS CDK Esegue l'app e sintetizza un AWS CloudFormation modello a partire da essa, quindi visualizza il modello. Il modello è un file YAML di circa 600 righe. L'inizio del file è mostrato di seguito. Il modello potrebbe differire da questo esempio.

```
Resources:
  MyWebServerLB3B5FD3AB:
```

```

Type: AWS::ElasticLoadBalancingV2::LoadBalancer
Properties:
  LoadBalancerAttributes:
    - Key: deletion_protection.enabled
      Value: "false"
  Scheme: internet-facing
  SecurityGroups:
    - Fn::GetAtt:
        - MyWebServerLBSecurityGroup01B285AA
        - GroupId
  Subnets:
    - Ref: EcsDefaultClusterMnL3mNNYNVpcPublicSubnet1Subnet3C273B99
    - Ref: EcsDefaultClusterMnL3mNNYNVpcPublicSubnet2Subnet95FF715A
  Type: application
DependsOn:
  - EcsDefaultClusterMnL3mNNYNVpcPublicSubnet1DefaultRouteFF4E2178
  - EcsDefaultClusterMnL3mNNYNVpcPublicSubnet2DefaultRouteB1375520
Metadata:
  aws:cdk:path: HelloEcsStack/MyWebServer/LB/Resource
MyWebServerLBSecurityGroup01B285AA:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Automatically created Security Group for ELB
HelloEcsStackMyWebServerLB06757F57
  SecurityGroupIngress:
    - CidrIp: 0.0.0.0/0
      Description: Allow from anyone on port 80
      FromPort: 80
      IpProtocol: tcp
      ToPort: 80
  VpcId:
    Ref: EcsDefaultClusterMnL3mNNYNVpc7788A521
  Metadata:
    aws:cdk:path: HelloEcsStack/MyWebServer/LB/SecurityGroup/Resource
# and so on for another few hundred lines

```

Per distribuire il servizio nella tua applicazione Account AWS, esegui il `cdk deploy` comando nella directory principale dell'applicazione. Ti viene chiesto di approvare le policy IAM che hanno generato. AWS CDK

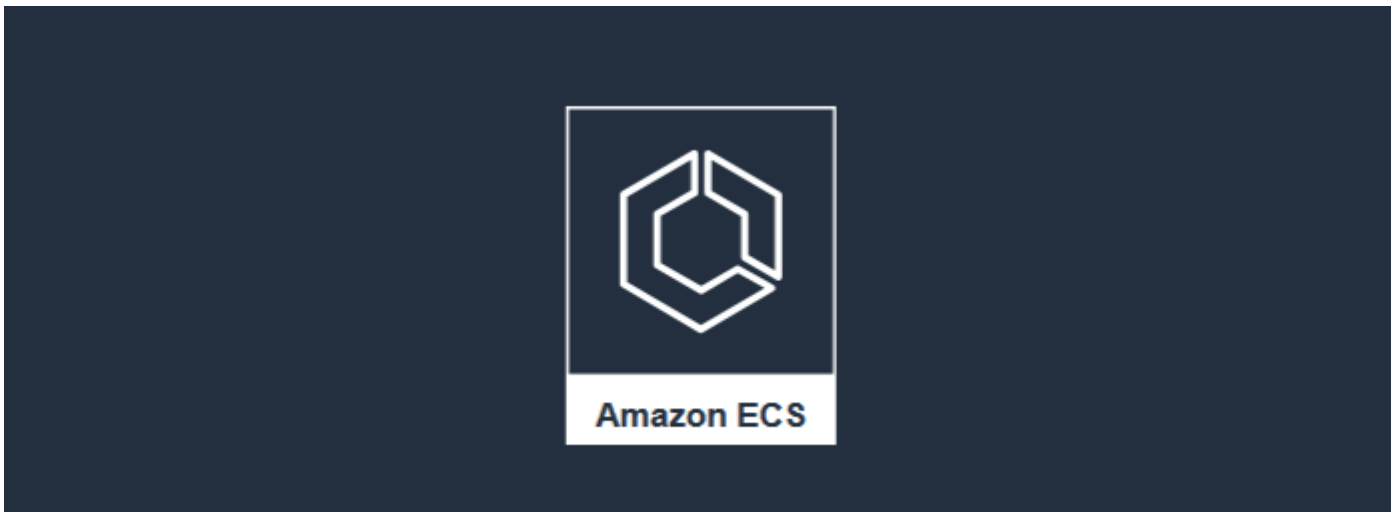
L'implementazione richiede diversi minuti, durante i quali vengono AWS CDK create diverse risorse. Le ultime righe dell'output dell'implementazione includono il nome host pubblico del load balancer e un URL del nuovo server Web. Sono i seguenti:

Outputs:

```
HelloEcsStack.MyWebServerLoadBalancerDNSXXXXXXXX = Hello-MyWeb-ZZZZZZZZZZZZ-  
ZZZZZZZZZZ.us-west-2.elb.amazonaws.com  
HelloEcsStack.MyWebServerServiceURLYYYYYYYYY = http://Hello-MyWeb-ZZZZZZZZZZZZ-  
ZZZZZZZZZZ.us-west-2.elb.amazonaws.com
```

Fase 3: Test del server Web

Copia l'URL dall'output dell'implementazione e incollalo nel browser Web. Viene visualizzato il seguente messaggio di benvenuto dal server Web.



Fase 4: pulizia

Dopo aver finito con il server Web, termina il servizio utilizzando il CDK eseguendo il comando `cdk destroy` nella directory principale dell'applicazione. In questo modo si evita di incorrere in addebiti non intenzionali in futuro.

Passaggi successivi

Per ulteriori informazioni su come sviluppare l' AWS infrastruttura utilizzando AWS CDK, consulta la [Guida per gli AWS CDK sviluppatori](#).

Per informazioni sulla scrittura di AWS CDK app nella lingua che preferisci, consulta quanto segue:

TypeScript

[Lavorare con AWS CDK in TypeScript](#)

JavaScript

[Lavorare con l' AWS CDK interno JavaScript](#)

Python

[Lavorare con il AWS CDK in Python](#)

Java

[Lavorare con il AWS CDK in Java](#)

C#

[Lavorare con il AWS CDK in C#](#)

Go

[Lavorare con AWS CDK in Go](#)

Per ulteriori informazioni sui moduli AWS Construct Library utilizzati in questo argomento, consultate le seguenti panoramiche di riferimento sulle AWS CDK API.

- [aws-ecs](#)
- [aws-ecs-patterns](#)

Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot

I comandi dell'interfaccia a riga di comando (CLI) di AWS Copilot semplificano la creazione, il rilascio e il funzionamento di applicazioni containerizzate pronte per la produzione su Amazon ECS da un ambiente di sviluppo locale. La AWS CLI di Copilot si allinea ai flussi di lavoro degli sviluppatori che supportano le migliori pratiche applicative moderne: dall'utilizzo dell'infrastruttura come codice alla creazione di una pipeline CI/CD fornita per conto di un utente. Utilizza la AWS CLI di Copilot come parte del tuo ciclo quotidiano di sviluppo e test come alternativa a. AWS Management Console

AWS Attualmente Copilot supporta i sistemi Linux, macOS e Windows. [Per ulteriori informazioni sulla versione più recente della AWS CLI di Copilot, vedere Releases.](#)

Note

Il codice sorgente per la AWS CLI di Copilot è disponibile su [GitHub](#). Consigliamo di inviare i problemi e le richieste pull per le modifiche che desideri siano incluse. Tuttavia, Amazon Web Services al momento non supporta l'esecuzione di copie modificate del codice di AWS Copilot. Segnala problemi con AWS Copilot collegandoti con noi su [Gitter](#) o [GitHub](#) dove puoi aprire problemi, fornire feedback e segnalare bug.

Argomenti

- [Installazione della CLI di AWS Copilot](#)
- [Distribuzione di un'applicazione Amazon ECS di esempio utilizzando la AWS CLI Copilot](#)

[La documentazione aggiuntiva per la AWS CLI di Copilot è disponibile sul sito Web di Copilot AWS](#)

Installazione della CLI di AWS Copilot

È possibile installare la AWS CLI di Copilot utilizzando Homebrew o scaricando manualmente il file binario con i seguenti passaggi.

Usa Homebrew

Il comando seguente viene utilizzato per installare la AWS CLI di Copilot sul sistema macOS o Linux utilizzando Homebrew. Prima dell'installazione, dovresti avere installato Homebrew. Per ulteriori informazioni, consulta [Homebrew](#).

```
brew install aws/tap/copilot-cli
```

Scarica il file binario

In alternativa a Homebrew, puoi installare manualmente la CLI di AWS Copilot sul tuo sistema macOS, Windows o Linux. Usa il seguente comando per il tuo sistema operativo per scaricare il file binario. Gli esempi macOS e Linux includono anche comandi che applicano le autorizzazioni di esecuzione al file binario ed elencano il menu di aiuto per verificare che l'installazione funzioni.

macOS

Per macOS:

```
sudo curl -Lo /usr/local/bin/copilot https://github.com/aws/copilot-cli/releases/
latest/download/copilot-darwin \
    && sudo chmod +x /usr/local/bin/copilot \
    && copilot --help
```

Per sistemi macOS ARM:

```
sudo curl -Lo /usr/local/bin/copilot https://github.com/aws/copilot-cli/releases/
latest/download/copilot-darwin-arm64 \
    && sudo chmod +x /usr/local/bin/copilot \
    && copilot --help
```

Linux

Per sistemi Linux x86 (64 bit):

```
sudo curl -Lo /usr/local/bin/copilot https://github.com/aws/copilot-cli/releases/
latest/download/copilot-linux \
    && sudo chmod +x /usr/local/bin/copilot \
    && copilot --help
```

Per i sistemi Linux ARM:

```
sudo curl -Lo /usr/local/bin/copilot https://github.com/aws/copilot-cli/releases/
latest/download/copilot-linux-arm64 \
    && sudo chmod +x /usr/local/bin/copilot \
    && copilot --help
```

Windows

Tramite Powershell, esegui il comando riportato:

```
New-Item -Path 'C:\copilot' -ItemType directory; `
Invoke-WebRequest -OutFile 'C:\copilot\copilot.exe' https://github.com/aws/
copilot-cli/releases/latest/download/copilot-windows.exe
```

(Facoltativo) Verifica della CLI di AWS Copilot installata manualmente tramite firme PGP

Gli eseguibili dell'interfaccia AWS CLI di Copilot sono firmati crittograficamente utilizzando firme PGP. Le firme PGP possono essere utilizzate per verificare la validità dell'eseguibile CLI di Copilot AWS . Puoi completare le fasi seguenti per verificare le firme tramite lo strumento GnuPG.

1. Scarica e installa GnuPG. Per ulteriori informazioni, consulta il [sito Web GnuPG](#) .

macOS

Consigliamo di usare Homebrew. Installa Homebrew seguendo le istruzioni fornite nel sito Web dello strumento. Per ulteriori informazioni, consulta [Homebrew](#). Dopo aver installato Homebrew, usa il comando seguente dal terminale macOS.

```
brew install gnupg
```

Linux

Installa gpg usando il programma di gestione dei pacchetti che preferisci per Linux.

Windows

Scarica il programma di installazione semplice i Windows dal sito Web GnuPG ed esegui l'installazione come amministratore. Dopo aver installato GnuPG, chiudi e riapri l'amministratore. PowerShell

Per ulteriori informazioni, consulta la pagina relativa al [download di GnuPG](#).

2. Verifica che il percorso GnuPG sia stato aggiunto al percorso dell'ambiente.

macOS

```
echo $PATH
```

Se non vedi il percorso GnuPG nell'output, esegui il seguente comando per aggiungerlo al percorso.

```
PATH=$PATH:<path to GnuPG executable files>
```

Linux

```
echo $PATH
```

Se non vedi il percorso GnuPG nell'output, esegui il seguente comando per aggiungerlo al percorso.

```
export PATH=$PATH:<path to GnuPG executable files>
```

Windows

```
Write-Output $Env:PATH
```

Se non vedi il percorso GnuPG nell'output, esegui il seguente comando per aggiungerlo al percorso.

```
$Env:PATH += "<path to GnuPG executable files>"
```

3. Crea un file di testo locale normale.

macOS

Sul terminale, inserisci:

```
touch <public_key_filename.txt>
```

Apri il file con. TextEdit

Linux

Crea un file di testo in un editor di testo, ad esempio gedit. Salva come `public_key_filename.txt`

Windows

Crea un file di testo in un editor di testo, ad esempio Notepad. Salva come `public_key_filename.txt`

4. Aggiungi i contenuti seguenti della chiave pubblica PGP di Amazon ECS e salva il file.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
```

```
Version: GnuPG v2
```

```
mQINBFq1SasBEADliGcT1NVJ1ydfN8DqebYYe9ne3dt6jqKFmKowLmm6LLGJe7HU
jGtqhCWRDkN+qPpHqdArRgDZAtn2pXY5fEipHgar4CP8QgRnRM02f174lmavr4Vg
7K/KH8VHlq2uRw32/B94XLEgRbGTMDwFdKuxoPCttBQaMj3LGn6Pe+6xVWRkChQu
BoQAjhjBQ+bEm0kNy0LjNgjNlnL3UMAG56t8E3LANIgGgEnpNsB1UwfwluPoGZoTx
N+6pHBjRkIL/1v/ETU4FXpYw2zvhwNahxeNRnoYj3uyCHkeliCw4kj0+skizBg0
2K7oVX80c3j5+Zilhl/qDLXmUCb2az5cMM1m0oF8EKX5HaNuq1KfwJxqXE6NNIc0
lFTTrT7QwD5fMNld3FanLgv/ZnIrsSaqJ0L6zRSq804LN10WBVBndExk2Kr+5kFxn
5lBPgfPgrJ5hQ+KTHMa9Y8Z7yUc64BJn6F9N17FJuSsfqbdkvRLsQRbcBG9qxX3
rJAEhieJzVMEUNl+EgeCkxj5xuSkNU7zw2c3hQZqEcRADLV+hvFJkt0z9Gm6xzbq
lTnWWCz4xrIWtuEBA2qE+MlDheVd78a3gIsEaSTfQq0osYXaQbvlnSW0oc1y/5Zb
zizHTJIhltUyls9WisP2s0emeHZicVMfw61EgPrJAiupgc7kyZvFt4YwfwARAQAB
tCRBbWF6b24gRUNTIDx1Y3Mtc2VjdXJpdHlAYW1hem9uLmNvbT6JAhwEEAECAAYF
AlrjL0YACgkQHivRXs0TaQrg1g/+JppwPqHn1VPmv7lessB8I5UqZeD6p6uVpHd7
Bs3pcPp8BV7BdRbs3sPlt5bV1+rkq0lw+0gZ4Q/ue/YbWt0At4qY00cEo0HgcnaX
lsB827QIfZIVtGWMhuh94xzm/SJkvngml6KB3YJNnWP61A9qJ37/VbVVLzvcmaZA
McWB4HUMNrh0JgBCo0gIpqCbpJEvUc02Bjn23eEJsS9kC70UAHyQkVnx4d9UzXF
40oISF6hmQKIBoLnRrAlj5Qvs3GhvHQ0ThYq0Grk/KMJJX2CSqt7tWJ8gk1n3H3Y
SReRXJRnv7DsDDbWfGt6r5Q2HW1TBUvaoZy5hF6maD09nHcNnvBjqADzeT8Tr/Qu
bBCLzkNSYqqkpgtwv7seoD2P4n1giRvDA0EFmZpVkuR+C252IaH1HZFEz+TvBVQM
Y80WWxmIJW+J6evjo3N1e019UHv71jvoF8z1jbI4bsL2c+QTJm0v7nRqzDQgCWyp
Id/v2dUVVTK1j9omuLBBwNjzQCB+72LcIzJhYmaP1HC4LcKQG+/f41exuItenatK
lEJQhYtyVXcBlh6Yn/wzNg2NW0wb3vqY/F7m6u9ixAwgtIMgPCDE4aJ86zrrXYFz
N2HqkTSQh77Z8KPKmyGopsmN/reMuilPdINb249nA0dzoN+nj+tTF0YCIaLaFyjs
Z0r1QA0JAjkeEwECACMFAlq1SasCGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIEAQIX
gAAKCRC86dmkLVF4T9iFEACEnkm1dNXsWUx34R3c0vamHrPxvfkyI1F1EUen8D1h
uX9xy6jCER0HWEp0rjGK4QDPgM93sWJ+s1UAKg214QRVzft0y9/DdR+twApA0fzy
uavIthGd6+03jAAo6udYDE+cZC3P7XBbDiYEWk4XAF9I1JjB8hTZUgvXBL046JhG
eM17+crgUyQeetki0QemLbsbXQ40Bd9V7zf7XJraFd8VrwNUwNb+9KFtgAsc9rk+
YIT/PEf+Y0PysgcxI4sTWghtyCulVnuGoskgDv4v73PALU0ieUrvvQVqWMRvhVx1
0X90J7cC1K0yh1EQQ1aFTgmQjmXexVTwIBm8LvysFK6YXM41Kj0r1z3+6xBIm/qe
bFyLUnf4WoiuOp1AaJhK9pRY+XENGNxdTn4D26Kd0F+PLkm3Tr3Hy3b10k34F1Gr
KVHUq1TZD7cvMnnKEELTUcKX+1mV3an16nmAg/my1JSUt6BNK2rJpY1s/kkSGSE
XQ4zuF2IGCpvBFhYAlT5Un5zwqkwwQR3/n2kwAoDzonJcehdw/C/cGos5D0aIU7I
K2X2aTD3+pA7Mx3IME2hqmYqRt9X42yF1PIEVRneBRJ3HDezAgJrNh0GQWRQkhIx
gz6/cTR+ekr5TptVsZS9few2GpI5bCgBKBisZIst89aw7mAKWut0Gcm4qM9/yK6
1bkCDQRatUmrARAaxNPvVwreJ2yAiFcUpdRlVhsu0gnxvs1QgsIw3H7+Pacr9Hpe
8uftYZqdC82KeSKhpHq7c8gMTMucIINTH25x9BCc73E33EjCL9Lqov1TL7+QkgHe
T+JIhZwdD8Mx2K+LVVVu/aWkNrfMuNwyDUciSI4D5QHa8T+F8fgN40TpWYjirzel
5yoICMr9hVcbzDNv/ozKCxjx+XKgnFc3wrnDfJfntfDAT7ecwbUTL+viQKJ646s+
psiqXRYtVvYInEhLVrJ0aV6zHFOigE/Bils6/g7ru1Q6CEHqEw++APs5CcE8VzJu
```

WAGSVHZgun5Y9N4quR/M9Vm+IPMhTxrAg7r0vyRN9cAXfeSMf77I+XTifigNna8x
t/M0djXr1fjF4pThEi5u6WsuRdFwjY2azEv3vevodTi4HoJReH6dFRa6y8c+UDg1
2iHi0KIpQqLbHEfQmHcDd2fix+AAJKMnPGNku9qCFEMbgSRJpXz6BfwnY1QuKE+I
R6jA0frUNT2jhiGG/F8RceXzohaaC/Cx7LUCUFwc0n7z32C9/Dtj7I1PM0acdZzz
bjJzRK0/ZDv+UN/c9dwAk11zAyPMwGBkUaY68EBstnIliW34aWm6IiHhxioVPKSp
VJfyiXP00EXqujtHLAeChfjcns3I12YshT1dv2PafG53fp33ZdzeUgsBo+EAEQEA
AYkCHwQYAIACQUCWrVJqwIbDAAKCRC86dmkLVF4T+ZdD/9x/8APzgNJF3o3STrF
jvnV1ycyhWYGaEBJiu7wjsNWwzMF0v15tLjB7AqeVxZn+WKDD/mIOQ450ZvnYZuy
X7DR0Jszah9wrYTzXLVruAu+t6UL0y/XQ4L1GZ9QR6+r+7t1Mvbfy7B1HbvX/gYt
Rwe/uwdibI0CagEzyX+2D3kT0LH05XThbXaNf8AN8zha91Jt2Q2UR2X5T6JcwtMz
FBvZn13LSmZyE0EQehS2iUurU4uW0pGppuqVnbi0jbCvCHKgDGrqZ0smKNAQng54
F365W3g8AFy48s8XQwzmccliowYX9bT8PZiEi0J4QmQh0aXkpqZyFefuWe0L2R94S
XKzr+gRh3BAULoqF+qK+IUMxTip9KTPNvYDpiC66yBiT6gFDji5Ca9pGpJXrC3xe
TXiKQ8DBWDhBPVPrruLIaenTtZE0sPc4I85yt5U9RoPTStc0r34s3w5yEaJagt6S
Gc5r9ysjkfH6+6rbi1ujxMgR0Sqtqr+RyB+V9A5/OgtNZc811K6u4Uo0Cde8jUuW
vqWKvjJB/Kz3u4zaeNu2ZyyHa0q0uH+TETcW+jsY9IhbEzqN5yQYGi4pVmDkY5vu
lXbJnbqPKPRXgM9BecV9AMBpgbDq/5LhNJXg+G8YQ0gp4lR/hC1TEFdIp5wM8AK
CWsENyt2o1rjgMXiZOMF8A5oBLkCDQRatUuSARAAr77kj7j2QR2SZe0S1FBvV7oS
mFeSNnz9xZssqrsm6bTwSHM6YLDwc7Sdf2esDdyz0NETwqrVCg+FxgL8hmo9hS4c
rR6tmrP0m0mptr+xlLsKcaP7ogIXsyZnrEAEsvW8PnfayoiPCdc3cMCR/1TnHFGA
7EuR/XLBmi7Qg9tByVYQ5Yj5wB9V4B2yeCt3XtzPqeLKvaxl7PNelAHGJQY/xo+m
V0bndxf9IY+4oFJ4b1D32WqvYxESo7vW6WBh7oqv3Zbm0yQrr8a6mDBpqLkvWwNI
3kpJR974tg5o5LfDu1BeeyHWPsgm4U/G4JB+JIG1ADy+RmoWEt4BqTCZ/knnoGvw
D5sTCxbKdmu0mhGyTssoG+300cGYHV7pWYPPhazKHMPm201xKCjH1RfzRULzGKjD+
yMLT1I3AXFmLmZJXika01vE3/wgMqCXscbycbLjLD/bXIuFwo3rzoeeXjgi/DJx
jKBAyBTY05nMctH109oaFd9d0Hbs0UDkIMnsgGBE766Piro6MHo0T0rXl07Tp4pI
rwuS0sc6XzCzdImj0Wc6axS/HeUKRXWdXJwno5awTwXKRJMXGfhCvSvbcbc2Wx+L
IKvmB7EB4K3fmjFFE67yolmiw2qRcUBfygtH3eL5XZU28MiCpue8Y8GKJoBAUyvF
KeM1r08Jm3iRac5a/D0AEQEAAYkEPgQYAIACQUCWrVLkgIbAgIpCRC86dmkLVF4
T8FdIAQZAQIABgUCWrVLkgAKCRDePL1hra+LjtHYD/9MucxdFe6bX01dQR4tKhhQ
P0LRqy6z1BY9ILCLowNdGZdqorogUiUymgn3VhEhVtxT0oHcN7q0uM01PNsRn0eS
EYjf8Xrb1clzkD6xULwm0clTb9bBxnBc/4PFvHAbZW3QzusaZniNgkuxt6BTfloS
Of4inq71kjmGK+TlzQ6mUMQUg228NUQC+a84EPqYyAeY1sgvgB7hJBhYL0QAxhcW
6m20Rd8iEc6HyZJ3yC0CsKip/nRWAbf00vfHfRBP0+m0ZwnJM8cPRFj0qqzFpKH9
HpDmTrC4wKP1+TL52LyEqNh4yZitXmZNV7giSRikk0eDSko+bFy6VbMzKUMkUJK3
D3eHFAMkujmbfJmSMTJ0PGn5SB1HyjCZNx6bhIIBQyEUB9gKCMUfaXKwKpF6rj0
iQXAJxLR/shZ5Rk96Vxz0phU17T90m/PnUEEPwq8KsBhnMRgxa0RFidDP+n9fgtv
HLmr0qX9zBCVXh0mdWYlRwvmzQFWzG7AoE55fkf8nAEPsalrCdtaNUBHRXA00QxG
AHM0dJQqVbSmqMvuAdjKdWpFu5y0MY5ddU+hiUzUyQLjL5Hhd5LOUddewlZgIw1j
xrEAUzDKetnemM8GkHxDgg8koev5frmShJuce7vSjKpCNg3EIJSGqM0PFjJuLWtZ
vjHeDNbJy6uNL65ckJy6WhGjEADS2WAW1D6Tfekkc21SsIXk/LqEpLMR/0g50Uif
wcEN1rS9IJXBwIy8Me1N9qr5KcKQLmfdFBNeyyceBhyVl0MDyH0KC+7PofMtkGBq
13QierHv5GJ8LB3fclqHV8pwTTo3Bc8z2g0TjmUYAN/ixETdReDoKavWJYSE9yoM
aaJu279ioVTrwpECse0XkiRyKToTjw0b73CGkBZZpJyqux/rmCV/fp4ALdSW8zbz

```

FJVORaivhoWwzjpfQKhwcU9lABXi2UvVm14v0AfeI7oiJPSU1zM4fEny4oiIBX1R
zhFNih1UjIu82X16mTm3BwbIga/s1fnQRGzyhqUIMii+mWra23EwjChaxpvjjcUH
5i1Lc5Zq781aCYRygYQw+hu5nFk0H1R+Z50Ubxjd/aqUfnGIAx7kPMD3Lof4K1dD
Q8ppQriUvxVo+4nPv6rpTy/PyqCLWDjkguHpJsEFsMkwajrAz0QNSAU5CJ0G2Zu4
yxvYlumHCE17nbFrm0vIiA75Sa8KnywTDsyZsu3Xc0cf3g+g1xWtpjJqy2bYX1qz
9uD0WtArWH0is6bq8l9RE6xr1RBVXS6uqqQIZFBGyq66b0dIq4D2JdsUvgEMaHbc
e7tBfeB1CMBdA64e9Rq7bFR7Tvt8gasCZY1Nr3lydh+dFHIEkH53HzQe6l88HEic
+0jVnLkCDQRa55wJARAAYLya2Lx6gyoWoJN1a6740q3o8e9d4KggQ0fGMTCf1meq
ivuzgN+3DZHN+9ty2KxXMtn0mhHBerZdbNJyjMNT1gAgrhPNB4HtXBxum2wS57WK
DNmade914L7FWTPAWBG2Wn4480EHTqsClICXXWy9IICgclAEyIq0Yq5mAdTEgRJS
Z8t4GpwtDL9gNQyFXaWQmDmkAsCygQMvha1mu9x0IzQG5CxSnZFk7zcuL60k14Z3
Cmt49k4T/7ZU8goWi8tt+rU78/IL3J/ff9+1civ10wuUidgfPCSv0UW1JojsdCQA
L+RZJcoXq71f0Fj/eNje0SstCTDPfTCL+kThE6E5neDtbQHBKEX1BRiTedsV4+M
ucgiTrdQFWKf89G72xdv8ut9AAYYQ2BbEYU+JAYhUH8rYYui2dHKJIgJNvJscuUWb
+QEjJIRleJRhr0+/CHgMs4fZAKWF1VFhKBkcKmEjLn1f7EJJUW84ZhKXj0/AUPX
1CHsNjziRceujCJYox1cwsoq6jTE50GiNzcIxTn9xUc0UMKFeggNAFys1K+TDTm3
Bzo8H5ucjCUEmUm9lhkGwqTZg0lRX5eqPX+JBoSa0bqhggCa5IPinKRa6MgoFPHK
6sYKqroYwBGgZm6Js5chpNchvJMs/3WXN0EVg0J3z3vP0DMhxqWm+r+n9z1W8qsA
EQEAAYKEPgQYAQgACQUCWuecCQIbAgIpCRC86dmkLVF4T8FdIAQZAQgABgUCWuec
CQAKCRBQ3szEcQ5hr+ykD/4t0LRHFHXuKUcxgGaubUcVtsFrwBKma1cYjqaPms8u
6Sk0wfGRI32G/Gh0rp0Ts/M0kb0bq6VLTh8N5Yc/53ME18zQFw9Y5AmRow4PZXER
uj5s57p4oR7xHMihMjCCBn1bvrR+34YPfgzTcgLi0EFHYT8UTxwnGmX0vNkMM7md
xD3CV5q6VAte8WKBo/220II3fcQ1c9r/owX4kXXkb0v9hoGwKbDJ1tzqTPrp/xFt
yohqnvImpnlz+Q9zXmbrWYL9/g8VCmW/NN2gju2G3Lu/TlFUWIT4v/50PK6TdeNb
VKJ04+S8bTayqSG9CML1S57KSgCo5HUHQWeSNHI+fpe5oX6FALPT9JLDce80Zz1i
cZZ0MELP37m00Qun0AlmHm/hVzf0f311PtzbzqWaE51tJvgUR/nZFo6Ta305Ezhs
3V1EJNQ1Ijf/6DH87SxvAoRIARCuZd0qxBCDK0avpFzUtbJd24lRA3WJpkEiMqKv
RDVZkE4b6TW61f0o+LaVfK6E8oLpixegS4fiqC16mFr0dyRk+RJJfIUyz0WTDVmt
g0U1C01ezokMSqkJ7724pyjr2xf/r9/sC6a0JwB/1KgZkJfC6NqL7TlxVA31dUga
LE0vEJTTE4gl+tYtfsCDvALCtqL0jduSkUo+RXcBItmXhA+tShW0pbS2Rtx/ixua
KohVD/0R4QxiSwQmICntm9mw9ydI1lyjYXX5a9x4wMJracNY/LBybJPFnZnT4dYR
z4XjqysDwvvYZByaWoIe3QxjX84V6M1I2IdAT/xImu8gbaCI8tmyfpIrLnPKiR9D
VFYfGBXuAX7+HgPPSFtrHQ0NCALxxz1bNpS+zxt9r0MiLgcLyspWxSdmoYGZ6nQP
R05Nm/ZVS+u2imPCRzNUZEMa+dLE6kHx0rS0dPiuJ407NtPeYDKkoQtNagspsDvh
cK7CSqAiKMq06UBTxqlTSRkm62e0Ctcs3p30eHu5GRZF1uzTET0ZxYkaPgdrQknx
ozjP5mC7X+451cCfmcVt94TFNL5HwEUVJpm0gmzILCI8yoDTWzloo+i+fPFsXX4f
kynhE83mSEcr5VHFYrTY3mQXGmNJ3bCLuc/jq7ysGq69xiKmTlUeXFm+aojCR05i
zyShIRJZ0GZfuzDYFDbMV9amA/YQGygLw//zP5ju5SW26dNx1f3MdFQE5JJ86rn9
MgZ4gcpazHEVUusbZsgkLizRp9imUiH8ymLqAXnFRG1U/LpNSefnvDFTtEIRcp0Hc
bhayG0bk51Bd4mio0XnIsKy4j63nJXA27x5EVVHQ1sYRN8Ny4Fdr2tMAmj20+X+J
qX2yy/UX5nSPU492e2CdZ1UhoU0SRFY3bxKHKb7SDbVeav+K5g==
=Gi5D
-----END PGP PUBLIC KEY BLOCK-----

```


Ecco i dettagli della chiave pubblica PGP di Amazon ECS come riferimento:

```
Key ID: BCE9D9A42D51784F
Type: RSA
Size: 4096/4096
Expires: Never
User ID: Amazon ECS
Key fingerprint: F34C 3DDA E729 26B0 79BE AEC6 BCE9 D9A4 2D51 784F
```

5. Importa il file con la chiave pubblica PGP di Amazon ECS con il seguente comando nel terminale.

```
gpg --import <public_key_filename.txt>
```

6. Scarica le firme dell' AWS interfaccia CLI di Copilot. Le firme sono firme PGP ASCII disconnesse archiviate in file con estensione .asc. Il file delle firme ha lo stesso nome dell'eseguibile corrispondente, con l'aggiunta di .asc.

macOS

Per i sistemi macOS, utilizza il seguente comando.

```
sudo curl -Lo copilot.asc https://github.com/aws/copilot-cli/releases/latest/download/copilot-darwin.asc
```

Linux

Per i sistemi Linux x86 (64 bit), esegui il comando riportato.

```
sudo curl -Lo copilot.asc https://github.com/aws/copilot-cli/releases/latest/download/copilot-linux.asc
```

Per i sistemi ARM Linux, esegui il comando riportato.

```
sudo curl -Lo copilot.asc https://github.com/aws/copilot-cli/releases/latest/download/copilot-linux-arm64.asc
```

Windows

Tramite Powershell, esegui il comando riportato.

```
Invoke-WebRequest -OutFile 'C:\copilot\copilot.asc' https://github.com/aws/copilot-cli/releases/latest/download/copilot-windows.exe.asc
```

7. Verifica la firma utilizzando il seguente comando:

- Per sistemi macOS e Linux:

```
gpg --verify copilot.asc /usr/local/bin/copilot
```

- Per i sistemi Windows:

```
gpg --verify 'C:\copilot\copilot.asc' 'C:\copilot\copilot.exe'
```

Output previsto:

```
gpg: Signature made Tue Apr  3 13:29:30 2018 PDT
gpg:                using RSA key DE3CBD61ADAF8B8E
gpg: Good signature from "Amazon ECS <ecs-security@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the owner.
Primary key fingerprint: F34C 3DDA E729 26B0 79BE  AEC6 BCE9 D9A4 2D51 784F
Subkey fingerprint: EB3D F841 E2C9 212A 2BD4  2232 DE3C BD61 ADAF 8B8E
```

Important

L'avviso nell'output è normale e non indica un problema. Si verifica perché non è presente una catena di attendibilità tra la chiave PGP personale (se disponibile) e la chiave PGP di Amazon ECS. Per ulteriori informazioni, consulta [Web of trust](#).

8. Per le installazioni Windows, esegui il seguente comando su Powershell per aggiungere la directory AWS Copilot al percorso.

```
$Env:PATH += ";<path to Copilot executable files>"
```

Distribuzione di un'applicazione Amazon ECS di esempio utilizzando la AWS CLI Copilot

Dopo aver installato la AWS CLI di Copilot, puoi seguire questi passaggi per distribuire un'app di esempio, verificare la distribuzione e ripulire le risorse.

Prerequisiti

Prima di cominciare, assicurati che i seguenti requisiti preliminari siano soddisfatti:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta la sezione [Interfaccia a riga di comando di AWS](#).
- Esegui `aws configure` per configurare un profilo predefinito che la AWS CLI di Copilot utilizzerà per gestire l'applicazione e i servizi.
- Installa ed esegui il Docker. Per ulteriori informazioni, consulta [Nozioni di base su Docker](#).

Implementa un'applicazione Amazon ECS di esempio utilizzando un solo comando

1. Implementa un'applicazione web di esempio clonata da un repository utilizzando il seguente comando. GitHub [Per ulteriori informazioni su AWS Copilot init e i relativi flag, consultate la documentazione di Copilot.AWS](#)

```
git clone https://github.com/aws-samples/aws-copilot-sample-service.git demo-app && \
cd demo-app && \
copilot init --app demo \
--name api \
--type 'Load Balanced Web Service' \
--dockerfile './Dockerfile' \
--port 80 \
--tag latest \
--deploy
```

2. Una volta completata la distribuzione, la AWS CLI di Copilot restituirà un URL che è possibile utilizzare per verificare la distribuzione. Puoi anche utilizzare i seguenti comandi per verificare lo stato dell'app.
 - Elenca tutte le tue applicazioni AWS Copilot.

```
copilot app ls
```

- Mostra informazioni sugli ambienti e sui servizi dell'applicazione.

```
copilot app show
```

- Mostra informazioni sugli ambienti in uso.

```
copilot env ls
```

- Mostra informazioni sul servizio, inclusi endpoint, capacità e risorse correlate.

```
copilot svc show
```

- Elenca tutti i servizi in un'applicazione.

```
copilot svc ls
```

- Mostra i log di un servizio implementato.

```
copilot svc logs
```

- Mostra lo stato del servizio.

```
copilot svc status
```

3. Al termine di questa demo, esegui il seguente comando per ripulire le risorse associate ed evitare di incorrere in addebiti per le risorse non utilizzate.

```
copilot app delete
```

Utilizzo di Amazon ECS con AWS CloudFormation

Amazon ECS è integrato con AWS CloudFormation, un servizio che puoi utilizzare per modellare e configurare AWS risorse con modelli definiti dall'utente. AWS CloudFormation utilizza modelli che sono un file YAML di testo JSON formattato. I modelli sono come progetti per la AWS risorsa che desideri creare. Quando crei e invii un modello, AWS CloudFormation crea una pila. Gestisci le risorse che hai definito nel modello tramite lo stack. Quando desideri creare, aggiornare o eliminare una risorsa, crei, aggiorni o elimini lo stack creato da quella risorsa. Per aggiornare gli stack, devi prima creare un set di modifiche. I set di modifiche mostrano cosa è influenzato dalla modifica prima di apportarla. In questo modo si evita di eliminare accidentalmente i database, ad esempio modificando il nome del database. Per ulteriori informazioni su modelli, pile e set di modifiche, consulta [How AWS CloudFormation works](#) in the AWS CloudFormation User Guide.

Utilizzando AWS CloudFormation, puoi dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Puoi creare un modello che descriva tutte le AWS risorse che desideri, come i cluster Amazon ECS, le definizioni delle attività, i servizi. Quindi, AWS CloudFormation si occupa del provisioning e della configurazione di tali risorse per te.

AWS CloudFormation consente inoltre di riutilizzare il modello per configurare le risorse Amazon ECS in modo coerente e ripetibile. Descrivi le tue risorse una sola volta e poi esegui nuovamente il provisioning delle stesse risorse su più e. Account AWS Regioni AWS

AWS CloudFormation i modelli possono essere utilizzati con AWS Management Console o AWS Command Line Interface per creare risorse.

Per ulteriori informazioni AWS CloudFormation, consulta le seguenti risorse:

- [AWS CloudFormation](#)
- [AWS CloudFormation Guida per l'utente](#)
- [AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando](#)

Argomenti

- [AWS CloudFormation modelli di esempio per Amazon ECS](#)
- [Creazione di risorse Amazon ECS utilizzando la console AWS CloudFormation](#)
- [Creazione di risorse Amazon ECS utilizzando AWS CLI comandi per AWS CloudFormation](#)

AWS CloudFormation modelli di esempio per Amazon ECS

Per effettuare il provisioning e configurare le risorse per Amazon ECS e i servizi correlati, assicurati di conoscere i AWS CloudFormation modelli. Per ulteriori informazioni, consulta i [AWS CloudFormation modelli](#) nella Guida per l'AWS CloudFormation utente. AWS CloudFormation i modelli sono file di testo in formato JSON o YAML che descrivono le risorse da inserire negli stack. AWS CloudFormation Se non conosci il formato JSON o YAML o entrambi, puoi utilizzarlo per iniziare a utilizzare i modelli. AWS Infrastructure Composer AWS CloudFormation Per ulteriori informazioni, consulta [Creare modelli visivamente con Infrastructure Composer](#) nella Guida per l'utente AWS CloudFormation

Puoi creare cluster Amazon ECS, definizioni di attività e servizi utilizzando. AWS CloudFormation I seguenti argomenti includono modelli che dimostrano come creare risorse con diverse configurazioni. È possibile creare queste risorse con questi modelli utilizzando la AWS CloudFormation console o il AWS CLI. Per ulteriori informazioni su come creare risorse utilizzando la AWS CloudFormation console, consulta la [Guida per l'AWS CloudFormation utente](#).

I seguenti argomenti elencano modelli di esempio per le definizioni di attività, i cluster e i servizi di Amazon ECS.

Argomenti

- [Definizioni di processi](#)
- [Cluster](#)
- [Servizi](#)
- [Ruoli IAM per Amazon ECS](#)

Definizioni di processi

Una definizione di attività è un modello per l'applicazione che descrive i parametri e uno o più contenitori che costituiscono l'applicazione. Di seguito sono riportati AWS CloudFormation modelli di esempio per le definizioni delle attività di Amazon ECS. Per ulteriori informazioni sulle definizioni delle attività di Amazon ECS, consulta [Definizioni dei processi di Amazon ECS](#).

Definizione delle attività Fargate Linux

È possibile utilizzare il seguente modello per creare un task Fargate Linux di esempio.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "ECS Task Definition with parameterized values",
  "Parameters": {
    "ContainerImage": {
      "Type": "String",
      "Default": "public.ecr.aws/docker/library/httpd:2.4",
      "Description": "The container image to use for the task"
    },
    "ContainerCpu": {
      "Type": "Number",
      "Default": 256,
      "Description": "The number of CPU units to reserve for the container",
      "AllowedValues": [256, 512, 1024, 2048, 4096]
    },
    "ContainerMemory": {
      "Type": "Number",
      "Default": 512,
      "Description": "The amount of memory (in MiB) to reserve for the container",
      "AllowedValues": [512, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192]
    },
    "TaskFamily": {
      "Type": "String",
      "Default": "task-definition-cfn",
      "Description": "The name of the task definition family"
    },
    "ContainerName": {
      "Type": "String",
      "Default": "sample-fargate-app",
      "Description": "The name of the container"
    },
    "ContainerPort": {
      "Type": "Number",
      "Default": 80,
      "Description": "The port number on the container"
    },
    "HostPort": {
      "Type": "Number",
      "Default": 80,
      "Description": "The port number on the host"
    },
    "ExecutionRoleArn": {
```

```

    "Type": "String",
    "Default": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole",
    "Description": "The ARN of the task execution role"
  },
  "LogGroup": {
    "Type": "String",
    "Default": "/ecs/fargate-task-definition",
    "Description": "The CloudWatch log group for container logs"
  },
  "NetworkMode": {
    "Type": "String",
    "Default": "awsvpc",
    "Description": "The Docker networking mode to use",
    "AllowedValues": ["awsvpc", "bridge", "host", "none"]
  },
  "OperatingSystemFamily": {
    "Type": "String",
    "Default": "LINUX",
    "Description": "The operating system for the task",
    "AllowedValues": ["LINUX", "WINDOWS_SERVER_2019_FULL",
"WINDOWS_SERVER_2019_CORE", "WINDOWS_SERVER_2022_FULL", "WINDOWS_SERVER_2022_CORE"]
  }
},
"Resources": {
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "ContainerDefinitions": [
        {
          "Command": [
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && -foreground\""]
          ],
          "EntryPoint": [
            "sh",
            "-c"
          ],
          "Essential": true,
          "Image": {"Ref": "ContainerImage"},
          "LogConfiguration": {

```



```

        "LogDriver": "awslogs",
        "Options": {
            "mode": "non-blocking",
            "max-buffer-size": "25m",
            "awslogs-create-group": "true",
            "awslogs-group": {"Ref": "LogGroup"},
            "awslogs-region": {"Ref": "AWS::Region"},
            "awslogs-stream-prefix": "ecs"
        }
    },
    "Name": {"Ref": "ContainerName"},
    "PortMappings": [
        {
            "ContainerPort": {"Ref": "ContainerPort"},
            "HostPort": {"Ref": "HostPort"},
            "Protocol": "tcp"
        }
    ]
}
],
"Cpu": {"Ref": "ContainerCpu"},
"ExecutionRoleArn": {"Ref": "ExecutionRoleArn"},
"Family": {"Ref": "TaskFamily"},
"Memory": {"Ref": "ContainerMemory"},
"NetworkMode": {"Ref": "NetworkMode"},
"RequiresCompatibilities": [
    "FARGATE"
],
"RuntimePlatform": {
    "OperatingSystemFamily": {"Ref": "OperatingSystemFamily"}
}
}
},
"Outputs": {
    "TaskDefinitionArn": {
        "Description": "The ARN of the created task definition",
        "Value": {"Ref": "ECSTaskDefinition"}
    }
}
}

```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: 'ECS Task Definition to deploy a sample app'
Parameters:
  ContainerImage:
    Type: String
    Default: 'public.ecr.aws/docker/library/httpd:2.4'
    Description: The container image to use for the task
  ContainerCpu:
    Type: Number
    Default: 256
    Description: The number of CPU units to reserve for the container
    AllowedValues: [256, 512, 1024, 2048, 4096]
  ContainerMemory:
    Type: Number
    Default: 512
    Description: The amount of memory (in MiB) to reserve for the container
    AllowedValues: [512, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192]
  TaskFamily:
    Type: String
    Default: 'task-definition-cfn'
    Description: The name of the task definition family
  ContainerName:
    Type: String
    Default: 'sample-fargate-app'
    Description: The name of the container
  ContainerPort:
    Type: Number
    Default: 80
    Description: The port number on the container
  HostPort:
    Type: Number
    Default: 80
    Description: The port number on the host
  ExecutionRoleArn:
    Type: String
    Default: 'arn:aws:iam::111122223333:role/ecsTaskExecutionRole'
    Description: The ARN of the task execution role
  LogGroup:
    Type: String
    Default: '/ecs/fargate-task-definition'
    Description: The CloudWatch log group for container logs
  NetworkMode:
```

```

Type: String
Default: 'awsvpc'
Description: The Docker networking mode to use
AllowedValues: ['awsvpc', 'bridge', 'host', 'none']
OperatingSystemFamily:
  Type: String
  Default: 'LINUX'
  Description: The operating system for the task
  AllowedValues: ['LINUX', 'WINDOWS_SERVER_2019_FULL', 'WINDOWS_SERVER_2019_CORE',
'WINDOWS_SERVER_2022_FULL', 'WINDOWS_SERVER_2022_CORE']
Resources:
  ECSTaskDefinition:
    Type: 'AWS::ECS::TaskDefinition'
    Properties:
      ContainerDefinitions:
        - Command:
            - >-
              /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color:
#333;} </style> </head><body> <div
style=color:white;text-align:center> <h1>Amazon ECS Sample
App</h1> <h2>Congratulations!</h2> <p>Your application is now
running on a container in Amazon ECS.</p> </div></body></html>' >
              /usr/local/apache2/htdocs/index.html && httpd-foreground"
        EntryPoint:
          - sh
          - '-c'
        Essential: true
        Image: !Ref ContainerImage
        LogConfiguration:
          LogDriver: awslogs
          Options:
            mode: non-blocking
            max-buffer-size: 25m
            awslogs-create-group: 'true'
            awslogs-group: !Ref LogGroup
            awslogs-region: !Ref AWS::Region
            awslogs-stream-prefix: ecs
        Name: !Ref ContainerName
        PortMappings:
          - ContainerPort: !Ref ContainerPort
            HostPort: !Ref HostPort
            Protocol: tcp
        Cpu: !Ref ContainerCpu

```

```

    ExecutionRoleArn: !Ref ExecutionRoleArn
    Family: !Ref TaskFamily
    Memory: !Ref ContainerMemory
    NetworkMode: !Ref NetworkMode
    RequiresCompatibilities:
      - FARGATE
    RuntimePlatform:
      OperatingSystemFamily: !Ref OperatingSystemFamily
  Outputs:
    TaskDefinitionArn:
      Description: The ARN of the created task definition
      Value: !Ref ECSTaskDefinition

```

Definizione delle attività di Amazon EFS

Puoi utilizzare il seguente modello per creare un'attività che utilizza un file system Amazon EFS creato da te. Per ulteriori informazioni sull'utilizzo dei volumi Amazon EFS con Amazon ECS, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "Create a task definition for a web server with parameterized values.",
  "Parameters": {
    "ExecutionRoleArn": {
      "Type": "String",
      "Default": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
      "Description": "The ARN of the task execution role"
    },
    "NetworkMode": {
      "Type": "String",
      "Default": "awsvpc",
      "Description": "The Docker networking mode to use",
      "AllowedValues": ["awsvpc", "bridge", "host", "none"]
    },
    "TaskFamily": {
      "Type": "String",
      "Default": "my-ecs-task",
      "Description": "The name of the task definition family"
    }
  },

```

```

"ContainerCpu": {
  "Type": "String",
  "Default": "256",
  "Description": "The number of CPU units to reserve for the container",
  "AllowedValues": ["256", "512", "1024", "2048", "4096"]
},
"ContainerMemory": {
  "Type": "String",
  "Default": "512",
  "Description": "The amount of memory (in MiB) to reserve for the container",
  "AllowedValues": ["512", "1024", "2048", "3072", "4096", "5120", "6144",
"7168", "8192"]
},
"ContainerName": {
  "Type": "String",
  "Default": "nginx",
  "Description": "The name of the container"
},
"ContainerImage": {
  "Type": "String",
  "Default": "public.ecr.aws/nginx/nginx:latest",
  "Description": "The container image to use for the task"
},
"ContainerPort": {
  "Type": "Number",
  "Default": 80,
  "Description": "The port number on the container"
},
"InitProcessEnabled": {
  "Type": "String",
  "Default": "true",
  "Description": "Whether to enable the init process inside the container",
  "AllowedValues": ["true", "false"]
},
"EfsVolumeName": {
  "Type": "String",
  "Default": "efs-volume",
  "Description": "The name of the EFS volume"
},
"EfsContainerPath": {
  "Type": "String",
  "Default": "/usr/share/nginx/html",
  "Description": "The path in the container where the EFS volume will be
mounted"

```

```

},
"LogGroup": {
  "Type": "String",
  "Default": "LogGroup",
  "Description": "The CloudWatch log group for container logs"
},
"LogStreamPrefix": {
  "Type": "String",
  "Default": "efs-task",
  "Description": "The prefix for the log stream"
},
"EfsFilesystemId": {
  "Type": "String",
  "Default": "fs-1234567890abcdef0",
  "Description": "The ID of the EFS filesystem"
},
"EfsRootDirectory": {
  "Type": "String",
  "Default": "/",
  "Description": "The root directory in the EFS filesystem"
},
"EfsTransitEncryption": {
  "Type": "String",
  "Default": "ENABLED",
  "Description": "Whether to enable transit encryption for EFS",
  "AllowedValues": ["ENABLED", "DISABLED"]
}
},
"Resources": {
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "ExecutionRoleArn": {"Ref": "ExecutionRoleArn"},
      "NetworkMode": {"Ref": "NetworkMode"},
      "RequiresCompatibilities": ["FARGATE"],
      "Family": {"Ref": "TaskFamily"},
      "Cpu": {"Ref": "ContainerCpu"},
      "Memory": {"Ref": "ContainerMemory"},
      "ContainerDefinitions": [
        {
          "Name": {"Ref": "ContainerName"},
          "Image": {"Ref": "ContainerImage"},
          "Essential": true,
          "PortMappings": [

```

```

        {
            "ContainerPort": {"Ref": "ContainerPort"},
            "Protocol": "tcp"
        }
    ],
    "LinuxParameters": {
        "InitProcessEnabled": {"Ref": "InitProcessEnabled"}
    },
    "MountPoints": [
        {
            "SourceVolume": {"Ref": "EfsVolumeName"},
            "ContainerPath": {"Ref": "EfsContainerPath"}
        }
    ],
    "LogConfiguration": {
        "LogDriver": "awslogs",
        "Options": {
            "mode": "non-blocking",
            "max-buffer-size": "25m",
            "awslogs-group": {"Ref": "LogGroup"},
            "awslogs-region": {"Ref": "AWS::Region"},
            "awslogs-create-group": "true",
            "awslogs-stream-prefix": {"Ref": "LogStreamPrefix"}
        }
    }
},
"Volumes": [
    {
        "Name": {"Ref": "EfsVolumeName"},
        "EFSVolumeConfiguration": {
            "FileSystemId": {"Ref": "EfsFileSystemId"},
            "RootDirectory": {"Ref": "EfsRootDirectory"},
            "TransitEncryption": {"Ref": "EfsTransitEncryption"}
        }
    }
]
}
},
"Outputs": {
    "TaskDefinitionArn": {
        "Description": "The ARN of the created task definition",
        "Value": {"Ref": "ECSTaskDefinition"}
    }
}

```

```

    }
  }
}
```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: Create a task definition for a web server with parameterized values.
Parameters:
  ExecutionRoleArn:
    Type: String
    Default: arn:aws:iam::123456789012:role/ecsTaskExecutionRole
    Description: The ARN of the task execution role
  NetworkMode:
    Type: String
    Default: awsvpc
    Description: The Docker networking mode to use
    AllowedValues: [awsvpc, bridge, host, none]
  TaskFamily:
    Type: String
    Default: my-ecs-task
    Description: The name of the task definition family
  ContainerCpu:
    Type: String
    Default: "256"
    Description: The number of CPU units to reserve for the container
    AllowedValues: ["256", "512", "1024", "2048", "4096"]
  ContainerMemory:
    Type: String
    Default: "512"
    Description: The amount of memory (in MiB) to reserve for the container
    AllowedValues: ["512", "1024", "2048", "3072", "4096", "5120", "6144", "7168",
"8192"]
  ContainerName:
    Type: String
    Default: nginx
    Description: The name of the container
  ContainerImage:
    Type: String
    Default: public.ecr.aws/nginx/nginx:latest
    Description: The container image to use for the task
  ContainerPort:
    Type: Number
```



```
    Default: 80
    Description: The port number on the container
InitProcessEnabled:
    Type: String
    Default: "true"
    Description: Whether to enable the init process inside the container
    AllowedValues: ["true", "false"]
EfsVolumeName:
    Type: String
    Default: efs-volume
    Description: The name of the EFS volume
EfsContainerPath:
    Type: String
    Default: /usr/share/nginx/html
    Description: The path in the container where the EFS volume will be mounted
LogGroup:
    Type: String
    Default: LogGroup
    Description: The CloudWatch log group for container logs
LogStreamPrefix:
    Type: String
    Default: efs-task
    Description: The prefix for the log stream
EfsFilesystemId:
    Type: String
    Default: fs-1234567890abcdef0
    Description: The ID of the EFS filesystem
EfsRootDirectory:
    Type: String
    Default: /
    Description: The root directory in the EFS filesystem
EfsTransitEncryption:
    Type: String
    Default: ENABLED
    Description: Whether to enable transit encryption for EFS
    AllowedValues: [ENABLED, DISABLED]
Resources:
    ECSTaskDefinition:
        Type: AWS::ECS::TaskDefinition
        Properties:
            ExecutionRoleArn: !Ref ExecutionRoleArn
            NetworkMode: !Ref NetworkMode
            RequiresCompatibilities:
                - FARGATE
```

```

Family: !Ref TaskFamily
Cpu: !Ref ContainerCpu
Memory: !Ref ContainerMemory
ContainerDefinitions:
  - Name: !Ref ContainerName
    Image: !Ref ContainerImage
    Essential: true
    PortMappings:
      - ContainerPort: !Ref ContainerPort
        Protocol: tcp
    LinuxParameters:
      InitProcessEnabled: !Ref InitProcessEnabled
    MountPoints:
      - SourceVolume: !Ref EfsVolumeName
        ContainerPath: !Ref EfsContainerPath
    LogConfiguration:
      LogDriver: awslogs
      Options:
        mode: non-blocking
        max-buffer-size: 25m
        awslogs-group: !Ref LogGroup
        awslogs-region: !Ref AWS::Region
        awslogs-create-group: "true"
        awslogs-stream-prefix: !Ref LogStreamPrefix
Volumes:
  - Name: !Ref EfsVolumeName
    EFSVolumeConfiguration:
      FilesystemId: !Ref EfsFilesystemId
      RootDirectory: !Ref EfsRootDirectory
      TransitEncryption: !Ref EfsTransitEncryption
Outputs:
  TaskDefinitionArn:
    Description: The ARN of the created task definition
    Value: !Ref ECSTaskDefinition

```

Cluster

Un cluster Amazon ECS è un raggruppamento logico di processi o servizi. Puoi utilizzare i seguenti modelli per creare cluster con configurazioni diverse. Per ulteriori informazioni sui cluster Amazon ECS, consulta [Cluster Amazon ECS](#)

Crea un cluster vuoto con impostazioni predefinite

È possibile utilizzare il seguente modello per creare un cluster vuoto con impostazioni predefinite.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "ECSCluster": {
      "Type": "AWS::ECS::Cluster",
      "Properties": {
        "ClusterName": "MyEmptyCluster"
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ECSCluster:
    Type: 'AWS::ECS::Cluster'
    Properties:
      ClusterName: MyEmptyCluster
```

Crea un cluster vuoto con crittografia dello storage gestita e Container Insights avanzato

Puoi utilizzare il seguente modello per creare un cluster con storage gestito a livello di cluster e Container Insights avanzato abilitato. La crittografia a livello di cluster si applica ai volumi di dati gestiti di Amazon ECS come i volumi Amazon EBS. Per ulteriori informazioni sulla crittografia Amazon EBS, consulta [Crittografia i dati archiviati nei volumi Amazon EBS collegati alle attività di Amazon ECS](#). Per ulteriori informazioni sull'utilizzo di Container Insights con osservabilità avanzata, consulta. [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#)

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "Cluster": {
      "Type": "AWS::ECS::Cluster",
      "Properties": {
        "ClusterName": "EncryptedEnhancedCluster",
        "ClusterSettings": [
          {
            "Name": "containerInsights",
            "Value": "enhanced"
          }
        ],
        "Configuration": {
          "ManagedStorageConfiguration": {
            "KmsKeyId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
          }
        }
      }
    }
  }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  Cluster:
    Type: AWS::ECS::Cluster
    Properties:
      ClusterName: EncryptedEnhancedCluster
      ClusterSettings:
        - Name: containerInsights
          Value: enhanced
      Configuration:
        ManagedStorageConfiguration:
          KmsKeyId: a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

Crea un cluster con Amazon AL2 023 ECS-Optimized-AMI

Puoi utilizzare il seguente modello per creare un cluster che utilizza un provider di capacità che avvia AL2 023 istanze su Amazon. EC2

Important

Per l'AMI più recente IDs, consulta l'AMI ottimizzata per [Amazon ECS](#) nella Amazon Elastic Container Service Developer Guide.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.",
  "Parameters": {
    "InstanceType": {
      "Type": "String",
      "Description": "EC2 instance type",
      "Default": "t2.medium",
      "AllowedValues": [
        "t1.micro",
        "t2.2xlarge",
        "t2.large",
        "t2.medium",
        "t2.micro",
        "t2.nano",
        "t2.small",
        "t2.xlarge",
        "t3.2xlarge",
        "t3.large",
        "t3.medium",
        "t3.micro",
        "t3.nano",
        "t3.small",
        "t3.xlarge"
      ]
    },
    "DesiredCapacity": {
```

```

        "Type": "Number",
        "Default": "0",
        "Description": "Number of EC2 instances to launch in your ECS cluster."
    },
    "MaxSize": {
        "Type": "Number",
        "Default": "100",
        "Description": "Maximum number of EC2 instances that can be launched in
your ECS cluster."
    },
    "ECSAMI": {
        "Description": "The Amazon Machine Image ID used for the cluster",
        "Type": "AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>",
        "Default": "/aws/service/ecs/optimized-ami/amazon-linux-2023/
recommended/image_id"
    },
    "VpcId": {
        "Type": "AWS::EC2::VPC::Id",
        "Description": "VPC ID where the ECS cluster is launched",
        "Default": "vpc-1234567890abcdef0"
    },
    "SubnetIds": {
        "Type": "List<AWS::EC2::Subnet::Id>",
        "Description": "List of subnet IDs where the EC2 instances will be
launched",
        "Default": "subnet-021345abcdef67890"
    }
},
"Resources": {
    "ECSCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {
            "ClusterSettings": [
                {
                    "Name": "containerInsights",
                    "Value": "enabled"
                }
            ]
        }
    },
    "ECSAutoScalingGroup": {
        "Type": "AWS::AutoScaling::AutoScalingGroup",
        "DependsOn": [
            "ECSCluster",

```

```

        "EC2Role"
    ],
    "Properties": {
        "VPCZoneIdentifier": {
            "Ref": "SubnetIds"
        },
        "LaunchTemplate": {
            "LaunchTemplateId": {
                "Ref": "ContainerInstances"
            },
            "Version": {
                "Fn::GetAtt": [
                    "ContainerInstances",
                    "LatestVersionNumber"
                ]
            }
        },
        "MinSize": 0,
        "MaxSize": {
            "Ref": "MaxSize"
        },
        "DesiredCapacity": {
            "Ref": "DesiredCapacity"
        },
        "NewInstancesProtectedFromScaleIn": true
    },
    "UpdatePolicy": {
        "AutoScalingReplacingUpdate": {
            "WillReplace": "true"
        }
    }
},
"ContainerInstances": {
    "Type": "AWS::EC2::LaunchTemplate",
    "Properties": {
        "LaunchTemplateName": "asg-launch-template-2",
        "LaunchTemplateData": {
            "ImageId": {
                "Ref": "ECSAMI"
            },
            "InstanceType": {
                "Ref": "InstanceType"
            },
            "IamInstanceProfile": {

```

```

        "Name": {
            "Ref": "EC2InstanceProfile"
        },
        "SecurityGroupIds": [
            {
                "Ref": "ContainerHostSecurityGroup"
            }
        ],
        "UserData": {
            "Fn::Base64": {
                "Fn::Sub": "#!/bin/bash -xe\n echo ECS_CLUSTER=
${ECSCluster} >> /etc/ecs/ecs.config\n yum install -y aws-cfn-bootstrap\n /opt/aws/
bin/cfn-init -v --stack ${AWS::StackId} --resource ContainerInstances --configsets
full_install --region ${AWS::Region} &\n"
            }
        },
        "MetadataOptions": {
            "HttpEndpoint": "enabled",
            "HttpTokens": "required"
        }
    }
},
"EC2InstanceProfile": {
    "Type": "AWS::IAM::InstanceProfile",
    "Properties": {
        "Path": "/",
        "Roles": [
            {
                "Ref": "EC2Role"
            }
        ]
    }
},
"CapacityProvider": {
    "Type": "AWS::ECS::CapacityProvider",
    "Properties": {
        "AutoScalingGroupProvider": {
            "AutoScalingGroupArn": {
                "Ref": "ECSAutoScalingGroup"
            },
            "ManagedScaling": {
                "InstanceWarmupPeriod": 60,

```



```

        "MinimumScalingStepSize": 1,
        "MaximumScalingStepSize": 100,
        "Status": "ENABLED",
        "TargetCapacity": 100
    },
    "ManagedTerminationProtection": "ENABLED"
}
},
"CapacityProviderAssociation": {
    "Type": "AWS::ECS::ClusterCapacityProviderAssociations",
    "Properties": {
        "CapacityProviders": [
            {
                "Ref": "CapacityProvider"
            }
        ],
        "Cluster": {
            "Ref": "ECSCluster"
        },
        "DefaultCapacityProviderStrategy": [
            {
                "Base": 0,
                "CapacityProvider": {
                    "Ref": "CapacityProvider"
                },
                "Weight": 1
            }
        ]
    }
},
"ContainerHostSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
        "GroupDescription": "Access to the EC2 hosts that run containers",
        "VpcId": {
            "Ref": "VpcId"
        }
    }
},
"EC2Role": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "AssumeRolePolicyDocument": {

```

```

        "Statement": [
            {
                "Effect": "Allow",
                "Principal": {
                    "Service": [
                        "ec2.amazonaws.com"
                    ]
                },
                "Action": [
                    "sts:AssumeRole"
                ]
            }
        ],
        "Path": "/",
        "ManagedPolicyArns": [
            "arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role",
            "arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore"
        ]
    },
    "ECSTaskExecutionRole": {
        "Type": "AWS::IAM::Role",
        "Properties": {
            "AssumeRolePolicyDocument": {
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": [
                                "ecs-tasks.amazonaws.com"
                            ]
                        },
                        "Action": [
                            "sts:AssumeRole"
                        ],
                        "Condition": {
                            "ArnLike": {
                                "aws:SourceArn": {
                                    "Fn::Sub": "arn:${AWS::Partition}:ecs:
${AWS::Region}:${AWS::AccountId}:*"
                                }
                            }
                        }
                    }
                ]
            }
        }
    },

```

```

        "StringEquals": {
            "aws:SourceAccount": {
                "Fn::Sub": "${AWS::AccountId}"
            }
        }
    },
    "Path": "/",
    "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
    ]
},
"Outputs": {
    "ClusterName": {
        "Description": "The ECS cluster into which to launch resources",
        "Value": "ECSCluster"
    },
    "ECSTaskExecutionRole": {
        "Description": "The role used to start up a task",
        "Value": "ECSTaskExecutionRole"
    },
    "CapacityProvider": {
        "Description": "The cluster capacity provider that the service should use to request capacity when it wants to start up a task",
        "Value": "CapacityProvider"
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Description: EC2 ECS cluster that starts out empty, with no EC2 instances yet. An ECS capacity provider automatically launches more EC2 instances as required on the fly when you request ECS to launch services or standalone tasks.
Parameters:
  InstanceType:
    Type: String

```

Description: EC2 instance type

Default: t2.medium

AllowedValues:

- t1.micro
- t2.2xlarge
- t2.large
- t2.medium
- t2.micro
- t2.nano
- t2.small
- t2.xlarge
- t3.2xlarge
- t3.large
- t3.medium
- t3.micro
- t3.nano
- t3.small
- t3.xlarge

DesiredCapacity:

Type: Number

Default: '0'

Description: Number of EC2 instances to launch in your ECS cluster.

MaxSize:

Type: Number

Default: '100'

Description: Maximum number of EC2 instances that can be launched in your ECS cluster.

ECSAMI:

Description: The Amazon Machine Image ID used for the cluster

Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>

Default: /aws/service/ecs/optimized-ami/amazon-linux-2023/recommended/image_id

VpcId:

Type: AWS::EC2::VPC::Id

Description: VPC ID where the ECS cluster is launched

Default: vpc-1234567890abcdef0

SubnetIds:

Type: List<AWS::EC2::Subnet::Id>

Description: List of subnet IDs where the EC2 instances will be launched

Default: subnet-021345abcdef67890

Resources:

ECSCluster:

Type: AWS::ECS::Cluster

Properties:

ClusterSettings:

```

    - Name: containerInsights
      Value: enabled
ECSAutoScalingGroup:
  Type: AWS::AutoScaling::AutoScalingGroup
  DependsOn:
    - ECSCluster
    - EC2Role
  Properties:
    VPCZoneIdentifier: !Ref SubnetIds
    LaunchTemplate:
      LaunchTemplateId: !Ref ContainerInstances
      Version: !GetAtt ContainerInstances.LatestVersionNumber
    MinSize: 0
    MaxSize: !Ref MaxSize
    DesiredCapacity: !Ref DesiredCapacity
    NewInstancesProtectedFromScaleIn: true
  UpdatePolicy:
    AutoScalingReplacingUpdate:
      WillReplace: 'true'
ContainerInstances:
  Type: AWS::EC2::LaunchTemplate
  Properties:
    LaunchTemplateName: asg-launch-template-2
    LaunchTemplateData:
      ImageId: !Ref ECSAMI
      InstanceType: !Ref InstanceType
      IamInstanceProfile:
        Name: !Ref EC2InstanceProfile
      SecurityGroupIds:
        - !Ref ContainerHostSecurityGroup
      UserData: !Base64
        Fn::Sub: |
          #!/bin/bash -xe
          echo ECS_CLUSTER=${ECSCluster} >> /etc/ecs/ecs.config
          yum install -y aws-cfn-bootstrap
          /opt/aws/bin/cfn-init -v --stack ${AWS::StackId} --resource
ContainerInstances --configsets full_install --region ${AWS::Region} &
      MetadataOptions:
        HttpEndpoint: enabled
        HttpTokens: required
EC2InstanceProfile:
  Type: AWS::IAM::InstanceProfile
  Properties:
    Path: /

```

```

Roles:
  - !Ref EC2Role
CapacityProvider:
  Type: AWS::ECS::CapacityProvider
  Properties:
    AutoScalingGroupProvider:
      AutoScalingGroupArn: !Ref ECSAutoScalingGroup
    ManagedScaling:
      InstanceWarmupPeriod: 60
      MinimumScalingStepSize: 1
      MaximumScalingStepSize: 100
      Status: ENABLED
      TargetCapacity: 100
    ManagedTerminationProtection: ENABLED
CapacityProviderAssociation:
  Type: AWS::ECS::ClusterCapacityProviderAssociations
  Properties:
    CapacityProviders:
      - !Ref CapacityProvider
    Cluster: !Ref ECSCluster
    DefaultCapacityProviderStrategy:
      - Base: 0
      CapacityProvider: !Ref CapacityProvider
      Weight: 1
ContainerHostSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Access to the EC2 hosts that run containers
    VpcId: !Ref VpcId
EC2Role:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ec2.amazonaws.com
          Action:
            - sts:AssumeRole
    Path: /
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
      - arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore

```

```

ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Statement:
        - Effect: Allow
          Principal:
            Service:
              - ecs-tasks.amazonaws.com
          Action:
            - sts:AssumeRole
          Condition:
            ArnLike:
              aws:SourceArn: !Sub arn:${AWS::Partition}:ecs:${AWS::Region}:
${AWS::AccountId}:*
            StringEquals:
              aws:SourceAccount: !Sub ${AWS::AccountId}
      Path: /
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
Outputs:
  ClusterName:
    Description: The ECS cluster into which to launch resources
    Value: ECSCluster
  ECSTaskExecutionRole:
    Description: The role used to start up a task
    Value: ECSTaskExecutionRole
  CapacityProvider:
    Description: The cluster capacity provider that the service should use to
request capacity when it wants to start up a task
    Value: CapacityProvider

```

Servizi

Puoi utilizzare un servizio Amazon ECS per eseguire e mantenere simultaneamente un determinato numero di istanze di una definizione di attività in un cluster Amazon ECS. Se una delle tue attività non riesce o si interrompe, il pianificatore del servizio Amazon ECS avvia un'altra istanza della definizione di attività per sostituirla. Ciò consente di mantenere il numero desiderato di attività nel servizio. I seguenti modelli possono essere utilizzati per distribuire servizi. Per ulteriori informazioni sui servizi Amazon ECS, consulta [Servizi Amazon ECS](#).

Distribuisci un'applicazione

Il modello seguente crea un servizio Amazon ECS denominato `cfn-service` che avvia e gestisce l'attività definita dalla definizione dell'attività. I modelli creano un cluster denominato `CFNCluster`. Il cluster contiene una definizione di attività Linux Fargate che configura un server Web. Il modello crea anche le risorse di rete e di registrazione necessarie. Il modello crea anche un ruolo di esecuzione delle attività di Amazon ECS. Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "A template that deploys an application that is built on an
  Apache web server Docker image by creating an Amazon ECS cluster, task definition,
  and service. The template also creates networking and logging resources, and an
  Amazon ECS task execution role.",
  "Parameters": {
    "ClusterName": {
      "Type": "String",
      "Default": "CFNCluster",
      "Description": "Name of the ECS Cluster"
    },
    "TaskFamily": {
      "Type": "String",
      "Default": "task-definition-cfn",
      "Description": "Family name for the Task Definition"
    },
    "ServiceName": {
      "Type": "String",
      "Default": "cfn-service",
      "Description": "Name of the ECS Service"
    },
    "ContainerImage": {
      "Type": "String",
      "Default": "public.ecr.aws/docker/library/httpd:2.4",
      "Description": "Container image to use for the task"
    },
    "TaskCpu": {
      "Type": "Number",
      "Default": 256,
      "AllowedValues": [
        256,
```



```
        512,  
        1024,  
        2048,  
        4096  
    ],  
    "Description": "CPU units for the task"  
},  
"TaskMemory": {  
    "Type": "Number",  
    "Default": 512,  
    "AllowedValues": [  
        512,  
        1024,  
        2048,  
        4096,  
        8192,  
        16384  
    ],  
    "Description": "Memory (in MiB) for the task"  
},  
"DesiredCount": {  
    "Type": "Number",  
    "Default": 1,  
    "Description": "Desired number of tasks to run"  
},  
"LogGroupName": {  
    "Type": "String",  
    "Default": "/ecs/fargate-task-definition",  
    "Description": "CloudWatch Log Group name"  
},  
"VpcCidr": {  
    "Type": "String",  
    "Default": "10.0.0.0/16",  
    "Description": "CIDR block for the VPC"  
},  
"PublicSubnet1Cidr": {  
    "Type": "String",  
    "Default": "10.0.0.0/24",  
    "Description": "CIDR block for public subnet 1"  
},  
"PublicSubnet2Cidr": {  
    "Type": "String",  
    "Default": "10.0.1.0/24",  
    "Description": "CIDR block for public subnet 2"
```

```

    }
  },
  "Resources": {
    "VPC": {
      "Type": "AWS::EC2::VPC",
      "Properties": {
        "CidrBlock": {
          "Ref": "VpcCidr"
        },
        "EnableDnsSupport": true,
        "EnableDnsHostnames": true,
        "Tags": [
          {
            "Key": "Name",
            "Value": {
              "Fn::Sub": "${AWS::StackName}-VPC"
            }
          }
        ]
      }
    },
    "InternetGateway": {
      "Type": "AWS::EC2::InternetGateway",
      "Properties": {
        "Tags": [
          {
            "Key": "Name",
            "Value": {
              "Fn::Sub": "${AWS::StackName}-IGW"
            }
          }
        ]
      }
    },
    "InternetGatewayAttachment": {
      "Type": "AWS::EC2::VPCGatewayAttachment",
      "Properties": {
        "InternetGatewayId": {
          "Ref": "InternetGateway"
        },
        "VpcId": {
          "Ref": "VPC"
        }
      }
    }
  }
}

```

```

    },
    "PublicSubnet1": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "VpcId": {
          "Ref": "VPC"
        },
        "AvailabilityZone": {
          "Fn::Select": [
            0,
            {
              "Fn::GetAZs": ""
            }
          ]
        },
        "CidrBlock": {
          "Ref": "PublicSubnet1Cidr"
        },
        "MapPublicIpOnLaunch": true,
        "Tags": [
          {
            "Key": "Name",
            "Value": {
              "Fn::Sub": "${AWS::StackName}-PublicSubnet1"
            }
          }
        ]
      }
    },
    "PublicSubnet2": {
      "Type": "AWS::EC2::Subnet",
      "Properties": {
        "VpcId": {
          "Ref": "VPC"
        },
        "AvailabilityZone": {
          "Fn::Select": [
            1,
            {
              "Fn::GetAZs": ""
            }
          ]
        },
        "CidrBlock": {

```

```

        "Ref": "PublicSubnet2Cidr"
    },
    "MapPublicIpOnLaunch": true,
    "Tags": [
        {
            "Key": "Name",
            "Value": {
                "Fn::Sub": "${AWS::StackName}-PublicSubnet2"
            }
        }
    ]
}
},
"PublicRouteTable": {
    "Type": "AWS::EC2::RouteTable",
    "Properties": {
        "VpcId": {
            "Ref": "VPC"
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-PublicRouteTable"
                }
            }
        ]
    }
},
"DefaultPublicRoute": {
    "Type": "AWS::EC2::Route",
    "DependsOn": "InternetGatewayAttachment",
    "Properties": {
        "RouteTableId": {
            "Ref": "PublicRouteTable"
        },
        "DestinationCidrBlock": "0.0.0.0/0",
        "GatewayId": {
            "Ref": "InternetGateway"
        }
    }
},
"PublicSubnet1RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",

```

```

    "Properties": {
      "RouteTableId": {
        "Ref": "PublicRouteTable"
      },
      "SubnetId": {
        "Ref": "PublicSubnet1"
      }
    }
  },
  "PublicSubnet2RouteTableAssociation": {
    "Type": "AWS::EC2::SubnetRouteTableAssociation",
    "Properties": {
      "RouteTableId": {
        "Ref": "PublicRouteTable"
      },
      "SubnetId": {
        "Ref": "PublicSubnet2"
      }
    }
  },
  "ECSSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Security group for ECS tasks",
      "VpcId": {
        "Ref": "VPC"
      },
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 443,
          "ToPort": 443,
          "CidrIp": "0.0.0.0/0"
        }
      ]
    }
  },
  "ECSTaskExecutionRole": {

```

```

    "Type": "AWS::IAM::Role",
    "Properties": {
      "AssumeRolePolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
          {
            "Effect": "Allow",
            "Principal": {
              "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
          }
        ]
      },
      "ManagedPolicyArns": [
        "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
      ]
    }
  },
  "TaskLogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "DeletionPolicy": "Retain",
    "UpdateReplacePolicy": "Retain",
    "Properties": {
      "LogGroupName": {
        "Ref": "LogGroupName"
      },
      "RetentionInDays": 30
    }
  },
  "ECSCluster": {
    "Type": "AWS::ECS::Cluster",
    "Properties": {
      "ClusterName": {
        "Ref": "ClusterName"
      }
    }
  },
  "ECSTaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
      "ContainerDefinitions": [
        {

```

```

        "Command": [
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</
h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html &&
httpd-foreground\""]
    ],
    "EntryPoint": [
        "sh",
        "-c"
    ],
    "Essential": true,
    "Image": {
        "Ref": "ContainerImage"
    },
    "LogConfiguration": {
        "LogDriver": "awslogs",
        "Options": {
            "mode": "non-blocking",
            "max-buffer-size": "25m",
            "awslogs-create-group": "true",
            "awslogs-group": {
                "Ref": "LogGroupName"
            },
            "awslogs-region": {
                "Ref": "AWS::Region"
            },
            "awslogs-stream-prefix": "ecs"
        }
    },
    "Name": "sample-fargate-app",
    "PortMappings": [
        {
            "ContainerPort": 80,
            "HostPort": 80,
            "Protocol": "tcp"
        }
    ]
},
"Cpu": {
    "Ref": "TaskCpu"
},

```

```

        "ExecutionRoleArn": {
            "Fn::GetAtt": [
                "ECSTaskExecutionRole",
                "Arn"
            ]
        },
        "Family": {
            "Ref": "TaskFamily"
        },
        "Memory": {
            "Ref": "TaskMemory"
        },
        "NetworkMode": "awsvpc",
        "RequiresCompatibilities": [
            "FARGATE"
        ],
        "RuntimePlatform": {
            "OperatingSystemFamily": "LINUX"
        }
    },
    "ECSService": {
        "Type": "AWS::ECS::Service",
        "DependsOn": [
            "PublicSubnet1RouteTableAssociation",
            "PublicSubnet2RouteTableAssociation"
        ],
        "Properties": {
            "ServiceName": {
                "Ref": "ServiceName"
            },
            "Cluster": {
                "Ref": "ECSCluster"
            },
            "DesiredCount": {
                "Ref": "DesiredCount"
            },
            "LaunchType": "FARGATE",
            "NetworkConfiguration": {
                "AwsVpcConfiguration": {
                    "AssignPublicIp": "ENABLED",
                    "SecurityGroups": [
                        {
                            "Ref": "ECSSecurityGroup"
                        }
                    ]
                }
            }
        }
    }
}

```



```

        }
    ],
    "Subnets": [
        {
            "Ref": "PublicSubnet1"
        },
        {
            "Ref": "PublicSubnet2"
        }
    ]
},
"TaskDefinition": {
    "Ref": "ECSTaskDefinition"
}
}
},
"Outputs": {
    "ClusterName": {
        "Description": "The name of the ECS cluster",
        "Value": {
            "Ref": "ECSCluster"
        }
    },
    "TaskDefinitionArn": {
        "Description": "The ARN of the task definition",
        "Value": {
            "Ref": "ECSTaskDefinition"
        }
    },
    "ServiceName": {
        "Description": "The name of the ECS service",
        "Value": {
            "Ref": "ECSService"
        }
    },
    "VpcId": {
        "Description": "The ID of the VPC",
        "Value": {
            "Ref": "VPC"
        }
    },
    "PublicSubnet1": {

```

```

        "Description": "The ID of public subnet 1",
        "Value": {
            "Ref": "PublicSubnet1"
        }
    },
    "PublicSubnet2": {
        "Description": "The ID of public subnet 2",
        "Value": {
            "Ref": "PublicSubnet2"
        }
    },
    "SecurityGroup": {
        "Description": "The ID of the security group",
        "Value": {
            "Ref": "ECSSecurityGroup"
        }
    },
    "ExecutionRoleArn": {
        "Description": "The ARN of the task execution role",
        "Value": {
            "Fn::GetAtt": [
                "ECSTaskExecutionRole",
                "Arn"
            ]
        }
    }
}
}
}
}
}

```

YAML

AWSTemplateFormatVersion: 2010-09-09

Description: A template that deploys an application that is built on an Apache web server Docker image by creating an Amazon ECS cluster, task definition, and service. The template also creates networking and logging resources, and an Amazon ECS task execution role.

Parameters:

ClusterName:

Type: String

Default: CFNCluster

Description: Name of the ECS Cluster

TaskFamily:

Type: String

```
    Default: task-definition-cfn
    Description: Family name for the Task Definition
  ServiceName:
    Type: String
    Default: cfn-service
    Description: Name of the ECS Service
  ContainerImage:
    Type: String
    Default: public.ecr.aws/docker/library/httpd:2.4
    Description: Container image to use for the task
  TaskCpu:
    Type: Number
    Default: 256
    AllowedValues: [256, 512, 1024, 2048, 4096]
    Description: CPU units for the task
  TaskMemory:
    Type: Number
    Default: 512
    AllowedValues: [512, 1024, 2048, 4096, 8192, 16384]
    Description: Memory (in MiB) for the task
  DesiredCount:
    Type: Number
    Default: 1
    Description: Desired number of tasks to run
  LogGroupName:
    Type: String
    Default: /ecs/fargate-task-definition
    Description: CloudWatch Log Group name
  VpcCidr:
    Type: String
    Default: 10.0.0.0/16
    Description: CIDR block for the VPC
  PublicSubnet1Cidr:
    Type: String
    Default: 10.0.0.0/24
    Description: CIDR block for public subnet 1
  PublicSubnet2Cidr:
    Type: String
    Default: 10.0.1.0/24
    Description: CIDR block for public subnet 2
  Resources:
    # VPC and Networking Resources
    VPC:
      Type: AWS::EC2::VPC
```

```
Properties:
  CidrBlock: !Ref VpcCidr
  EnableDnsSupport: true
  EnableDnsHostnames: true
  Tags:
    - Key: Name
      Value: !Sub ${AWS::StackName}-VPC
InternetGateway:
  Type: AWS::EC2::InternetGateway
  Properties:
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-IGW
InternetGatewayAttachment:
  Type: AWS::EC2::VPCGatewayAttachment
  Properties:
    InternetGatewayId: !Ref InternetGateway
    VpcId: !Ref VPC
PublicSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Ref PublicSubnet1Cidr
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicSubnet1
PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PublicSubnet2Cidr
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicSubnet2
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
```

```

    Value: !Sub ${AWS::StackName}-PublicRouteTable
DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1
PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2
# Security Group
ECSSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: Security group for ECS tasks
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
      - IpProtocol: tcp
        FromPort: 443
        ToPort: 443
        CidrIp: 0.0.0.0/0

# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com

```

```

    Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
# CloudWatch Logs
TaskLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    LogGroupName: !Ref LogGroupName
    RetentionInDays: 30
# ECS Resources
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterName: !Ref ClusterName
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Command:
          - >-
            /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
            App</title> <style>body {margin-top: 40px; background-color:
            #333;} </style> </head><body> <div
            style=color:white;text-align:center> <h1>Amazon ECS Sample
            App</h1> <h2>Congratulations!</h2> <p>Your application is now
            running on a container in Amazon ECS.</p> </div></body></html>' >
            /usr/local/apache2/htdocs/index.html && httpd-foreground"s
    EntryPoint:
      - sh
      - '-c'
    Essential: true
    Image: !Ref ContainerImage
    LogConfiguration:
      LogDriver: awslogs
      Options:
        mode: non-blocking
        max-buffer-size: 25m
        awslogs-create-group: 'true'
        awslogs-group: !Ref LogGroupName
        awslogs-region: !Ref 'AWS::Region'
        awslogs-stream-prefix: ecs
    Name: sample-fargate-app

```

```
    PortMappings:
      - ContainerPort: 80
        HostPort: 80
        Protocol: tcp
    Cpu: !Ref TaskCpu
    ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
    Family: !Ref TaskFamily
    Memory: !Ref TaskMemory
    NetworkMode: awsvpc
    RequiresCompatibilities:
      - FARGATE
    RuntimePlatform:
      OperatingSystemFamily: LINUX
  ECSService:
    Type: AWS::ECS::Service
    DependsOn:
      - PublicSubnet1RouteTableAssociation
      - PublicSubnet2RouteTableAssociation
    Properties:
      ServiceName: !Ref ServiceName
      Cluster: !Ref ECSCluster
      DesiredCount: !Ref DesiredCount
      LaunchType: FARGATE
      NetworkConfiguration:
        AwsvpcConfiguration:
          AssignPublicIp: ENABLED
          SecurityGroups:
            - !Ref ECSSecurityGroup
          Subnets:
            - !Ref PublicSubnet1
            - !Ref PublicSubnet2
      TaskDefinition: !Ref ECSTaskDefinition
  Outputs:
    ClusterName:
      Description: The name of the ECS cluster
      Value: !Ref ECSCluster
    TaskDefinitionArn:
      Description: The ARN of the task definition
      Value: !Ref ECSTaskDefinition
    ServiceName:
      Description: The name of the ECS service
      Value: !Ref ECSService
    VpcId:
      Description: The ID of the VPC
```

```

Value: !Ref VPC
PublicSubnet1:
  Description: The ID of public subnet 1
  Value: !Ref PublicSubnet1
PublicSubnet2:
  Description: The ID of public subnet 2
  Value: !Ref PublicSubnet2
SecurityGroup:
  Description: The ID of the security group
  Value: !Ref ECSSecurityGroup
ExecutionRoleArn:
  Description: The ARN of the task execution role
  Value: !GetAtt ECSTaskExecutionRole.Arn

```

Implementa un servizio con ECS Exec abilitato

È possibile utilizzare il seguente modello per distribuire un servizio con ECS Exec abilitato. Il servizio viene eseguito in un cluster con una chiave KMS per la crittografia delle sessioni ECS Exec e una configurazione di registro per il reindirizzamento dei log delle sessioni di comando di esecuzione verso un bucket Amazon S3. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS con ECS Exec](#).

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "ECS Cluster with Fargate Service and Task Definition and ECS Exec enabled.",
  "Parameters": {
    "ClusterName": {
      "Type": "String",
      "Default": "CFNCluster",
      "Description": "Name of the ECS Cluster"
    },
    "S3BucketName": {
      "Type": "String",
      "Default": "amzn-s3-demo-bucket",
      "Description": "S3 bucket for ECS execute command logs"
    },
    "KmsKeyId": {
      "Type": "String",
      "Default": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",

```



```

        "Description": "KMS Key ID for ECS execute command encryption"
    },
    "ContainerImage": {
        "Type": "String",
        "Default": "public.ecr.aws/docker/library/httpd:2.4",
        "Description": "Container image to use for the task"
    },
    "ContainerCpu": {
        "Type": "Number",
        "Default": 256,
        "AllowedValues": [256, 512, 1024, 2048, 4096],
        "Description": "CPU units for the container (256 = 0.25 vCPU)"
    },
    "ContainerMemory": {
        "Type": "Number",
        "Default": 512,
        "AllowedValues": [512, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192],
        "Description": "Memory for the container (in MiB)"
    },
    "DesiredCount": {
        "Type": "Number",
        "Default": 1,
        "Description": "Desired count of tasks in the service"
    },
    "SecurityGroups": {
        "Type": "List<AWS::EC2::SecurityGroup::Id>",
        "Description": "Security Group IDs for the ECS Service"
    },
    "Subnets": {
        "Type": "List<AWS::EC2::Subnet::Id>",
        "Description": "Subnet IDs for the ECS Service"
    },
    "ServiceName": {
        "Type": "String",
        "Default": "cfn-service",
        "Description": "Name of the ECS service"
    },
    "TaskFamily": {
        "Type": "String",
        "Default": "task-definition-cfn",
        "Description": "Family name for the task definition"
    },
    "TaskExecutionRoleArn": {
        "Type": "String",

```

```

        "Description": "ARN of an existing IAM role for ECS task execution",
        "Default": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole"
    },
    "TaskRoleArn": {
        "Type": "String",
        "Description": "ARN of an existing IAM role for ECS tasks",
        "Default": "arn:aws:iam::111122223333:role/execTaskRole"
    }
},
"Resources": {
    "ECSCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {
            "ClusterName": {"Ref": "ClusterName"},
            "Configuration": {
                "ExecuteCommandConfiguration": {
                    "Logging": "OVERRIDE",
                    "LogConfiguration": {
                        "S3BucketName": {"Ref": "S3BucketName"}
                    },
                    "KmsKeyId": {"Ref": "KmsKeyId"}
                }
            },
            "Tags": [
                {
                    "Key": "Environment",
                    "Value": {"Ref": "AWS::StackName"}
                }
            ]
        }
    },
    "ECSTaskDefinition": {
        "Type": "AWS::ECS::TaskDefinition",
        "Properties": {
            "ContainerDefinitions": [
                {
                    "Command": [
                        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS
Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style>
</head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</
h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html &&
httpd-foreground\""
                    ]
                }
            ]
        }
    }
}

```

```

        "EntryPoint": [
            "sh",
            "-c"
        ],
        "Essential": true,
        "Image": {"Ref": "ContainerImage"},
        "LogConfiguration": {
            "LogDriver": "awslogs",
            "Options": {
                "mode": "non-blocking",
                "max-buffer-size": "25m",
                "awslogs-create-group": "true",
                "awslogs-group": {"Fn::Sub": "/ecs/"
${AWS::StackName}"},
                "awslogs-region": {"Ref": "AWS::Region"},
                "awslogs-stream-prefix": "ecs"
            }
        },
        "Name": "sample-fargate-app",
        "PortMappings": [
            {
                "ContainerPort": 80,
                "HostPort": 80,
                "Protocol": "tcp"
            }
        ]
    }
],
    "Cpu": {"Ref": "ContainerCpu"},
    "ExecutionRoleArn": {"Ref": "TaskExecutionRoleArn"},
    "TaskRoleArn": {"Ref": "TaskRoleArn"},
    "Family": {"Ref": "TaskFamily"},
    "Memory": {"Ref": "ContainerMemory"},
    "NetworkMode": "awsvpc",
    "RequiresCompatibilities": [
        "FARGATE"
    ],
    "RuntimePlatform": {
        "OperatingSystemFamily": "LINUX"
    },
    "Tags": [
        {
            "Key": "Name",
            "Value": {"Fn::Sub": "${AWS::StackName}-TaskDefinition"}
        }
    ]
}

```

```

    }
  ]
}
},
"ECSService": {
  "Type": "AWS::ECS::Service",
  "Properties": {
    "ServiceName": {"Ref": "ServiceName"},
    "Cluster": {"Ref": "ECSCluster"},
    "DesiredCount": {"Ref": "DesiredCount"},
    "EnableExecuteCommand": true,
    "LaunchType": "FARGATE",
    "NetworkConfiguration": {
      "AwsvpcConfiguration": {
        "AssignPublicIp": "ENABLED",
        "SecurityGroups": {"Ref": "SecurityGroups"},
        "Subnets": {"Ref": "Subnets"}
      }
    },
    "TaskDefinition": {"Ref": "ECSTaskDefinition"},
    "Tags": [
      {
        "Key": "Name",
        "Value": {"Fn::Sub": "${AWS::StackName}-Service"}
      }
    ]
  }
}
},
"Outputs": {
  "ClusterName": {
    "Description": "The name of the ECS cluster",
    "Value": {"Ref": "ECSCluster"}
  },
  "ServiceName": {
    "Description": "The name of the ECS service",
    "Value": {"Ref": "ServiceName"}
  },
  "TaskDefinitionArn": {
    "Description": "The ARN of the task definition",
    "Value": {"Ref": "ECSTaskDefinition"}
  },
  "ClusterArn": {
    "Description": "The ARN of the ECS cluster",

```

```

        "Value": {"Fn::GetAtt": ["ECSCluster", "Arn"]}]
    },
    "StackName": {
        "Description": "The name of this stack",
        "Value": {"Ref": "AWS::StackName"}
    },
    "StackId": {
        "Description": "The unique identifier for this stack",
        "Value": {"Ref": "AWS::StackId"}
    },
    "Region": {
        "Description": "The AWS Region where the stack is deployed",
        "Value": {"Ref": "AWS::Region"}
    },
    "AccountId": {
        "Description": "The AWS Account ID",
        "Value": {"Ref": "AWS::AccountId"}
    }
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Description: ECS Cluster with Fargate Service and Task Definition and ECS Exec
enabled.
Parameters:
  ClusterName:
    Type: String
    Default: CFNCluster
    Description: Name of the ECS Cluster
  S3BucketName:
    Type: String
    Default: amzn-s3-demo-bucket
    Description: S3 bucket for ECS execute command logs
  KmsKeyId:
    Type: String
    Default: a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
    Description: KMS Key ID for ECS execute command encryption
  ContainerImage:
    Type: String
    Default: public.ecr.aws/docker/library/httpd:2.4
    Description: Container image to use for the task

```

```

ContainerCpu:
  Type: Number
  Default: 256
  AllowedValues: [256, 512, 1024, 2048, 4096]
  Description: CPU units for the container (256 = 0.25 vCPU)
ContainerMemory:
  Type: Number
  Default: 512
  AllowedValues: [512, 1024, 2048, 3072, 4096, 5120, 6144, 7168, 8192]
  Description: Memory for the container (in MiB)
DesiredCount:
  Type: Number
  Default: 1
  Description: Desired count of tasks in the service
SecurityGroups:
  Type: List<AWS::EC2::SecurityGroup::Id>
  Description: Security Group IDs for the ECS Service
Subnets:
  Type: List<AWS::EC2::Subnet::Id>
  Description: Subnet IDs for the ECS Service
ServiceName:
  Type: String
  Default: cfn-service
  Description: Name of the ECS service
TaskFamily:
  Type: String
  Default: task-definition-cfn
  Description: Family name for the task definition
TaskExecutionRoleArn:
  Type: String
  Description: ARN of an existing IAM role for ECS task execution
  Default: 'arn:aws:iam::111122223333:role/ecsTaskExecutionRole'
TaskRoleArn:
  Type: String
  Description: ARN of an existing IAM role for ECS tasks
  Default: 'arn:aws:iam::111122223333:role/execTaskRole'
Resources:
  ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
      ClusterName: !Ref ClusterName
      Configuration:
        ExecuteCommandConfiguration:
          Logging: OVERRIDE

```

```

    LogConfiguration:
      S3BucketName: !Ref S3BucketName
      KmsKeyId: !Ref KmsKeyId
    Tags:
      - Key: Environment
        Value: !Ref AWS::StackName
  ECSTaskDefinition:
    Type: AWS::ECS::TaskDefinition
    Properties:
      ContainerDefinitions:
        - Command:
            - >-
              /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
              App</title> <style>body {margin-top: 40px; background-color:
              #333;} </style> </head><body> <div
              style=color:white;text-align:center> <h1>Amazon ECS Sample
              App</h1> <h2>Congratulations!</h2> <p>Your application is now
              running on a container in Amazon ECS.</p> </div></body></html>' >
              /usr/local/apache2/htdocs/index.html && httpd-foreground"
          EntryPoint:
            - sh
            - '-c'
          Essential: true
          Image: !Ref ContainerImage
          LogConfiguration:
            LogDriver: awslogs
            Options:
              mode: non-blocking
              max-buffer-size: 25m
              awslogs-create-group: 'true'
              awslogs-group: !Sub /ecs/${AWS::StackName}
              awslogs-region: !Ref AWS::Region
              awslogs-stream-prefix: ecs
          Name: sample-fargate-app
          PortMappings:
            - ContainerPort: 80
              HostPort: 80
              Protocol: tcp
          Cpu: !Ref ContainerCpu
          ExecutionRoleArn: !Ref TaskExecutionRoleArn
          TaskRoleArn: !Ref TaskRoleArn
          Family: !Ref TaskFamily
          Memory: !Ref ContainerMemory
          NetworkMode: awsvpc

```

```
RequiresCompatibilities:
  - FARGATE
RuntimePlatform:
  OperatingSystemFamily: LINUX
Tags:
  - Key: Name
    Value: !Sub ${AWS::StackName}-TaskDefinition
ECSService:
  Type: AWS::ECS::Service
  Properties:
    ServiceName: !Ref ServiceName
    Cluster: !Ref ECSCluster
    DesiredCount: !Ref DesiredCount
    EnableExecuteCommand: true
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups: !Ref SecurityGroups
        Subnets: !Ref Subnets
    TaskDefinition: !Ref ECSTaskDefinition
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-Service
Outputs:
  ClusterName:
    Description: The name of the ECS cluster
    Value: !Ref ECSCluster
  ServiceName:
    Description: The name of the ECS service
    Value: !Ref ServiceName
  TaskDefinitionArn:
    Description: The ARN of the task definition
    Value: !Ref ECSTaskDefinition
  ClusterArn:
    Description: The ARN of the ECS cluster
    Value: !GetAtt ECSCluster.Arn
  StackName:
    Description: The name of this stack
    Value: !Ref AWS::StackName
  StackId:
    Description: The unique identifier for this stack
    Value: !Ref AWS::StackId
  Region:
```



```
Description: The AWS Region where the stack is deployed
Value: !Ref AWS::Region
AccountId:
Description: The AWS Account ID
Value: !Ref AWS::AccountId
```

Implementa un servizio che utilizza Amazon VPC Lattice

Puoi utilizzare il seguente modello per iniziare a creare un servizio Amazon ECS con VPC Lattice. Potrebbe essere necessario completare i seguenti passaggi aggiuntivi per configurare VPC Lattice:

- Aggiorna le regole in entrata del tuo gruppo di sicurezza per VPC Lattice per consentire il prefisso delle `vpc-lattice` regole in entrata e consentire il traffico sulla porta 80.
- Associa il VPC per il servizio a una rete di servizi VPC Lattice.
- Configura una zona ospitata privata o pubblica con Amazon Route 53.
- Configura i listener e le regole degli ascoltatori in un servizio VPC Lattice.
- Verifica le configurazioni dei controlli sanitari del gruppo target.

Per ulteriori informazioni sull'utilizzo di VPC Lattice con Amazon ECS, consulta. [Usa Amazon VPC Lattice per connettere, osservare e proteggere i tuoi servizi Amazon ECS](#)

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "The template used to create an ECS Service with VPC Lattice.",
  "Parameters": {
    "ECSClusterName": {
      "Type": "String",
      "Default": "vpc-lattice-cluster"
    },
    "ECSServiceName": {
      "Type": "String",
      "Default": "vpc-lattice-service"
    },
    "SecurityGroupIDs": {
      "Type": "List<AWS::EC2::SecurityGroup::Id>",
      "Description": "Security Group IDs for the ECS Service"
    }
  },
```

```

"SubnetIDs": {
  "Type": "List<AWS::EC2::Subnet::Id>",
  "Description": "Subnet IDs for the ECS Service"
},
"VpcID": {
  "Type": "AWS::EC2::VPC::Id",
  "Description": "VPC ID for the resources"
},
"ContainerImage": {
  "Type": "String",
  "Default": "public.ecr.aws/docker/library/httpd:2.4",
  "Description": "Container image to use for the task"
},
"TaskCpu": {
  "Type": "Number",
  "Default": 256,
  "AllowedValues": [256, 512, 1024, 2048, 4096],
  "Description": "CPU units for the task"
},
"TaskMemory": {
  "Type": "Number",
  "Default": 512,
  "AllowedValues": [512, 1024, 2048, 4096, 8192, 16384],
  "Description": "Memory (in MiB) for the task"
},
"LogGroupName": {
  "Type": "String",
  "Default": "/ecs/vpc-lattice-task",
  "Description": "CloudWatch Log Group name"
},
"EnableContainerInsights": {
  "Type": "String",
  "Default": "enabled",
  "AllowedValues": ["enabled", "disabled"],
  "Description": "Enable or disable CloudWatch Container Insights for the cluster"
},
"Resources": {
  "ECSCluster": {
    "Type": "AWS::ECS::Cluster",
    "Properties": {
      "ClusterName": {"Ref": "ECSClusterName"},
      "ClusterSettings": [
        {

```

```

    "Name": "containerInsights",
    "Value": {"Ref": "EnableContainerInsights"}
  },
],
"Tags": [
  {
    "Key": "Name",
    "Value": {"Ref": "ECSClusterName"}
  }
]
},
"ECSTaskExecutionRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "ManagedPolicyArns": [
      "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
    ]
  }
},
"TaskLogGroup": {
  "Type": "AWS::Logs::LogGroup",
  "DeletionPolicy": "Retain",
  "UpdateReplacePolicy": "Retain",
  "Properties": {
    "LogGroupName": {"Ref": "LogGroupName"},
    "RetentionInDays": 30
  }
},
"VpcLatticeTaskDefinition": {
  "Type": "AWS::ECS::TaskDefinition",
  "Properties": {

```

```

"ContainerDefinitions": [
  {
    "Command": [
      "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
<style>body {margin-top: 40px; background-color: #333;} </style> </head><body>
<div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
<h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html &&
httpd-foreground\""]
    ],
    "EntryPoint": [
      "sh",
      "-c"
    ],
    "Essential": true,
    "Image": {"Ref": "ContainerImage"},
    "LogConfiguration": {
      "LogDriver": "awslogs",
      "Options": {
        "mode": "non-blocking",
        "max-buffer-size": "25m",
        "awslogs-create-group": "true",
        "awslogs-group": {"Ref": "LogGroupName"},
        "awslogs-region": {"Ref": "AWS::Region"},
        "awslogs-stream-prefix": "ecs"
      }
    },
    "Name": "vpc-lattice-container",
    "PortMappings": [
      {
        "ContainerPort": 80,
        "HostPort": 80,
        "Protocol": "tcp",
        "Name": "vpc-lattice-port"
      }
    ]
  },
  {
    "Cpu": {"Ref": "TaskCpu"},
    "ExecutionRoleArn": {"Fn::GetAtt": ["ECSTaskExecutionRole", "Arn"]},
    "Family": "vpc-lattice-task-definition",
    "Memory": {"Ref": "TaskMemory"},
    "NetworkMode": "awsvpc",
    "RequiresCompatibilities": [

```

```

    "FARGATE"
  ],
  "RuntimePlatform": {
    "OperatingSystemFamily": "LINUX"
  }
},
"ECSService": {
  "Type": "AWS::ECS::Service",
  "Properties": {
    "Cluster": {"Ref": "ECSCluster"},
    "TaskDefinition": {"Ref": "VpcLatticeTaskDefinition"},
    "LaunchType": "FARGATE",
    "ServiceName": {"Ref": "ECSServiceName"},
    "SchedulingStrategy": "REPLICA",
    "DesiredCount": 2,
    "AvailabilityZoneRebalancing": "ENABLED",
    "NetworkConfiguration": {
      "AwsvpcConfiguration": {
        "AssignPublicIp": "ENABLED",
        "SecurityGroups": {
          "Ref": "SecurityGroupIDs"
        },
        "Subnets": {
          "Ref": "SubnetIDs"
        }
      }
    },
    "PlatformVersion": "LATEST",
    "VpcLatticeConfigurations": [
      {
        "RoleArn": "arn:aws:iam::111122223333:role/ecsInfrastructureRole",
        "PortName": "vpc-lattice-port",
        "TargetGroupArn": {
          "Ref": "TargetGroup1"
        }
      }
    ],
    "DeploymentConfiguration": {
      "DeploymentCircuitBreaker": {
        "Enable": true,
        "Rollback": true
      },
      "MaximumPercent": 200,

```

```
    "MinimumHealthyPercent": 100
  },
  "DeploymentController": {
    "Type": "ECS"
  },
  "ServiceConnectConfiguration": {
    "Enabled": false
  },
  "Tags": [],
  "EnableECSTags": true
}
},
"TargetGroup1": {
  "Type": "AWS::VpcLattice::TargetGroup",
  "Properties": {
    "Type": "IP",
    "Name": "first-target-group",
    "Config": {
      "Port": 80,
      "Protocol": "HTTP",
      "VpcIdentifier": {"Ref": "VpcID"},
      "HealthCheck": {
        "Enabled": true,
        "Path": "/"
      }
    }
  },
  "Tags": [
    {
      "Key": "ecs-application-networking/ServiceName",
      "Value": {"Ref": "ECSServiceName"}
    },
    {
      "Key": "ecs-application-networking/ClusterName",
      "Value": {"Ref": "ECSClusterName"}
    },
    {
      "Key": "ecs-application-networking/TaskDefinition",
      "Value": {"Ref": "VpcLatticeTaskDefinition"}
    },
    {
      "Key": "ecs-application-networking/VpcId",
      "Value": {"Ref": "VpcID"}
    }
  ]
}
```

```

    }
  },
  "Outputs": {
    "ClusterName": {
      "Description": "The cluster used to create the service.",
      "Value": {
        "Ref": "ECSCluster"
      }
    },
    "ClusterArn": {
      "Description": "The ARN of the ECS cluster",
      "Value": {
        "Fn::GetAtt": ["ECSCluster", "Arn"]
      }
    },
    "ECSService": {
      "Description": "The created service.",
      "Value": {
        "Ref": "ECSService"
      }
    },
    "TaskDefinitionArn": {
      "Description": "The ARN of the task definition",
      "Value": {
        "Ref": "VpcLatticeTaskDefinition"
      }
    }
  }
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Description: The template used to create an ECS Service with VPC Lattice.

Parameters:
  ECSClusterName:
    Type: String
    Default: vpc-lattice-cluster
  ECSServiceName:
    Type: String
    Default: vpc-lattice-service

```

```
SecurityGroupIDs:
  Type: List<AWS::EC2::SecurityGroup::Id>
  Description: Security Group IDs for the ECS Service
SubnetIDs:
  Type: List<AWS::EC2::Subnet::Id>
  Description: Subnet IDs for the ECS Service
VpcID:
  Type: AWS::EC2::VPC::Id
  Description: VPC ID for the resources
ContainerImage:
  Type: String
  Default: public.ecr.aws/docker/library/httpd:2.4
  Description: Container image to use for the task
TaskCpu:
  Type: Number
  Default: 256
  AllowedValues: [256, 512, 1024, 2048, 4096]
  Description: CPU units for the task
TaskMemory:
  Type: Number
  Default: 512
  AllowedValues: [512, 1024, 2048, 4096, 8192, 16384]
  Description: Memory (in MiB) for the task
LogGroupName:
  Type: String
  Default: /ecs/vpc-lattice-task
  Description: CloudWatch Log Group name
EnableContainerInsights:
  Type: String
  Default: 'enhanced'
  AllowedValues: ['enabled', 'disabled', 'enhanced']
  Description: Enable or disable CloudWatch Container Insights for the cluster

Resources:
  # ECS Cluster
  ECSCluster:
    Type: AWS::ECS::Cluster
    Properties:
      ClusterName: !Ref ECSClusterName
      ClusterSettings:
        - Name: containerInsights
          Value: !Ref EnableContainerInsights
      Tags:
        - Key: Name
```



```

    Value: !Ref ECSClusterName

# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy

# CloudWatch Logs
TaskLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    LogGroupName: !Ref LogGroupName
    RetentionInDays: 30

# Task Definition
VpcLatticeTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Command:
          - >-
            /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
            App</title> <style>body {margin-top: 40px; background-color:
            #333;} </style> </head><body> <div
            style=color:white;text-align:center> <h1>Amazon ECS Sample
            App</h1> <h2>Congratulations!</h2> <p>Your application is now
            running on a container in Amazon ECS.</p> </div></body></html>' >
            /usr/local/apache2/htdocs/index.html && httpd-foreground"
      EntryPoint:
        - sh
        - '-c'
    Essential: true
    Image: !Ref ContainerImage

```

```

LogConfiguration:
  LogDriver: awslogs
  Options:
    mode: non-blocking
    max-buffer-size: 25m
    awslogs-create-group: 'true'
    awslogs-group: !Ref LogGroupName
    awslogs-region: !Ref 'AWS::Region'
    awslogs-stream-prefix: ecs
  Name: vpc-lattice-container
PortMappings:
  - ContainerPort: 80
    HostPort: 80
    Protocol: tcp
    Name: vpc-lattice-port
Cpu: !Ref TaskCpu
ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
Family: vpc-lattice-task-definition
Memory: !Ref TaskMemory
NetworkMode: awsvpc
RequiresCompatibilities:
  - FARGATE
RuntimePlatform:
  OperatingSystemFamily: LINUX

```

ECSService:

```

Type: AWS::ECS::Service
Properties:
  Cluster: !Ref ECSCluster
  TaskDefinition: !Ref VpcLatticeTaskDefinition
  LaunchType: FARGATE
  ServiceName: !Ref ECSServiceName
  SchedulingStrategy: REPLICA
  DesiredCount: 2
  AvailabilityZoneRebalancing: ENABLED
  NetworkConfiguration:
    AwsvpcConfiguration:
      AssignPublicIp: ENABLED
      SecurityGroups: !Ref SecurityGroupIDs
      Subnets: !Ref SubnetIDs
  PlatformVersion: LATEST
  VpcLatticeConfigurations:
    - RoleArn: arn:aws:iam::111122223333:role/ecsInfrastructureRole
      PortName: vpc-lattice-port

```

```

    TargetGroupArn: !Ref TargetGroup1
  DeploymentConfiguration:
    DeploymentCircuitBreaker:
      Enable: true
      Rollback: true
      MaximumPercent: 200
      MinimumHealthyPercent: 100
    DeploymentController:
      Type: ECS
    ServiceConnectConfiguration:
      Enabled: false
    Tags: []
    EnableECSTags: true

```

```

TargetGroup1:
  Type: AWS::VpcLattice::TargetGroup
  Properties:
    Type: IP
    Name: first-target-group
    Config:
      Port: 80
      Protocol: HTTP
      VpcIdentifier: !Ref VpcID
      HealthCheck:
        Enabled: true
        Path: /
    Tags:
      - Key: ecs-application-networking/ServiceName
        Value: !Ref ECSServiceName
      - Key: ecs-application-networking/ClusterName
        Value: !Ref ECSClusterName
      - Key: ecs-application-networking/TaskDefinition
        Value: !Ref VpcLatticeTaskDefinition
      - Key: ecs-application-networking/VpcId
        Value: !Ref VpcID

```

```

Outputs:
  ClusterName:
    Description: The cluster used to create the service.
    Value: !Ref ECSCluster
  ClusterArn:
    Description: The ARN of the ECS cluster
    Value: !GetAtt ECSCluster.Arn
  ECSService:

```

```
Description: The created service.  
Value: !Ref ECSService  
TaskDefinitionArn:  
Description: The ARN of the task definition  
Value: !Ref VpcLatticeTaskDefinition
```

Implementa il servizio con una configurazione di volume

Il modello seguente include una configurazione del volume nella definizione del servizio. Amazon ECS supporta la configurazione dei seguenti volumi di dati utilizzando una configurazione di volume al momento del lancio: Volumi Amazon EBS. Per ulteriori informazioni sui volumi Amazon EBS, consulta [Usa i volumi Amazon EBS con Amazon ECS](#).

JSON

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "The template used to create an ECS Service that includes a  
volume configuration. The configuration is used to create Amazon EBS volumes for  
attachment to the tasks. One volume is attached per task.",  
  "Parameters": {  
    "ECSClusterName": {  
      "Type": "String",  
      "Default": "volume-config-cluster",  
      "Description": "Name of the ECS cluster"  
    },  
    "SecurityGroupIDs": {  
      "Type": "List<AWS::EC2::SecurityGroup::Id>",  
      "Description": "Security Group IDs for the ECS Service"  
    },  
    "SubnetIDs": {  
      "Type": "List<AWS::EC2::Subnet::Id>",  
      "Description": "Subnet IDs for the ECS Service"  
    },  
    "InfrastructureRoleArn": {  
      "Type": "String",  
      "Description": "ARN of the IAM role that ECS will use to manage EBS volumes"  
    },  
    "ContainerImage": {  
      "Type": "String",  
      "Default": "public.ecr.aws/nginx/nginx:latest",  
      "Description": "Container image to use for the task"  
    }  
  }
```

```
},
"TaskCpu": {
  "Type": "String",
  "Default": "2048",
  "Description": "CPU units for the task"
},
"TaskMemory": {
  "Type": "String",
  "Default": "4096",
  "Description": "Memory (in MiB) for the task"
},
"VolumeSize": {
  "Type": "String",
  "Default": "10",
  "Description": "Size of the EBS volume in GiB"
},
"VolumeType": {
  "Type": "String",
  "Default": "gp3",
  "AllowedValues": ["gp2", "gp3", "io1", "io2", "st1", "sc1", "standard"],
  "Description": "EBS volume type"
},
"VolumeIops": {
  "Type": "String",
  "Default": "3000",
  "Description": "IOPS for the EBS volume (required for io1, io2, and gp3)"
},
"VolumeThroughput": {
  "Type": "String",
  "Default": "125",
  "Description": "Throughput for the EBS volume (only for gp3)"
},
"FilesystemType": {
  "Type": "String",
  "Default": "xfs",
  "AllowedValues": ["xfs", "ext4"],
  "Description": "Filesystem type for the EBS volume"
},
"EnableContainerInsights": {
  "Type": "String",
  "Default": "enhanced",
  "AllowedValues": ["enabled", "disabled", "enhanced"],
  "Description": "Enable or disable CloudWatch Container Insights for the
cluster"
```

```

    }
  },
  "Resources": {
    "ECSCluster": {
      "Type": "AWS::ECS::Cluster",
      "Properties": {
        "ClusterName": {"Ref": "ECSClusterName"},
        "ClusterSettings": [
          {
            "Name": "containerInsights",
            "Value": {"Ref": "EnableContainerInsights"}
          }
        ],
        "Tags": [
          {
            "Key": "Name",
            "Value": {"Ref": "ECSClusterName"}
          }
        ]
      }
    },
    "ECSTaskExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Version": "2012-10-17",
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
              },
              "Action": "sts:AssumeRole"
            }
          ]
        },
        "ManagedPolicyArns": [
          "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
        ]
      }
    },
    "EBSTaskDefinition": {
      "Type": "AWS::ECS::TaskDefinition",
      "Properties": {

```

```

    "Family": "ebs-task-attach-task-def",
    "ExecutionRoleArn": {"Fn::GetAtt": ["ECSTaskExecutionRole", "Arn"]},
    "NetworkMode": "awsvpc",
    "RequiresCompatibilities": [
        "EC2",
        "FARGATE"
    ],
    "Cpu": {"Ref": "TaskCpu"},
    "Memory": {"Ref": "TaskMemory"},
    "ContainerDefinitions": [
        {
            "Name": "nginx",
            "Image": {"Ref": "ContainerImage"},
            "Essential": true,
            "PortMappings": [
                {
                    "Name": "nginx-80-tcp",
                    "ContainerPort": 80,
                    "HostPort": 80,
                    "Protocol": "tcp",
                    "AppProtocol": "http"
                }
            ],
            "MountPoints": [
                {
                    "SourceVolume": "ebs-vol",
                    "ContainerPath": "/foo-container-path",
                    "ReadOnly": false
                }
            ]
        }
    ],
    "Volumes": [
        {
            "Name": "ebs-vol",
            "ConfiguredAtLaunch": true
        }
    ]
},
"ECSService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
        "Cluster": {"Ref": "ECSCluster"},

```

```

"TaskDefinition": {"Ref": "EBSTaskDefinition"},
"LaunchType": "FARGATE",
"ServiceName": "ebs",
"SchedulingStrategy": "REPLICA",
"DesiredCount": 1,
"NetworkConfiguration": {
  "AwsVpcConfiguration": {
    "AssignPublicIp": "ENABLED",
    "SecurityGroups": {"Ref": "SecurityGroupIDs"},
    "Subnets": {"Ref": "SubnetIDs"}
  }
},
"PlatformVersion": "LATEST",
"DeploymentConfiguration": {
  "MaximumPercent": 200,
  "MinimumHealthyPercent": 100,
  "DeploymentCircuitBreaker": {
    "Enable": true,
    "Rollback": true
  }
},
"DeploymentController": {
  "Type": "ECS"
},
"Tags": [],
"EnableECSTags": true,
"VolumeConfigurations": [
  {
    "Name": "ebs-vol",
    "ManagedEBSVolume": {
      "RoleArn": {"Ref": "InfrastructureRoleArn"},
      "VolumeType": {"Ref": "VolumeType"},
      "Iops": {"Ref": "VolumeIops"},
      "Throughput": {"Ref": "VolumeThroughput"},
      "SizeInGiB": {"Ref": "VolumeSize"},
      "FilesystemType": {"Ref": "FilesystemType"},
      "TagSpecifications": [
        {
          "ResourceType": "volume",
          "PropagateTags": "TASK_DEFINITION"
        }
      ]
    }
  }
]
}

```



```

    ]
  }
}
},
"Outputs": {
  "ClusterName": {
    "Description": "The cluster used to create the service.",
    "Value": {"Ref": "ECSCluster"}
  },
  "ClusterArn": {
    "Description": "The ARN of the ECS cluster",
    "Value": {"Fn::GetAtt": ["ECSCluster", "Arn"]}
  },
  "ECSService": {
    "Description": "The created service.",
    "Value": {"Ref": "ECSService"}
  },
  "TaskDefinitionArn": {
    "Description": "The ARN of the task definition",
    "Value": {"Ref": "EBSTaskDefinition"}
  }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: 2010-09-09
Description: The template used to create an ECS Service that includes a volume
  configuration. The configuration is used to create Amazon EBS volumes for
  attachment to the tasks. One volume is attached per task.
Parameters:
  ECSClusterName:
    Type: String
    Default: volume-config-cluster
    Description: Name of the ECS cluster

  SecurityGroupIDs:
    Type: List<AWS::EC2::SecurityGroup::Id>
    Description: Security Group IDs for the ECS Service

  SubnetIDs:
    Type: List<AWS::EC2::Subnet::Id>
    Description: Subnet IDs for the ECS Service

```

InfrastructureRoleArn:

Type: String

Description: ARN of the IAM role that ECS will use to manage EBS volumes

ContainerImage:

Type: String

Default: public.ecr.aws/nginx/nginx:latest

Description: Container image to use for the task

TaskCpu:

Type: String

Default: "2048"

Description: CPU units for the task

TaskMemory:

Type: String

Default: "4096"

Description: Memory (in MiB) for the task

VolumeSize:

Type: String

Default: "10"

Description: Size of the EBS volume in GiB

VolumeType:

Type: String

Default: gp3

AllowedValues: [gp2, gp3, io1, io2, st1, sc1, standard]

Description: EBS volume type

VolumeIops:

Type: String

Default: "3000"

Description: IOPS for the EBS volume (required for io1, io2, and gp3)

VolumeThroughput:

Type: String

Default: "125"

Description: Throughput for the EBS volume (only for gp3)

FilesystemType:

Type: String

Default: xfs

```
AllowedValues: [xfs, ext4]
```

```
Description: Filesystem type for the EBS volume
```

```
EnableContainerInsights:
```

```
  Type: String
```

```
  Default: 'enhanced'
```

```
  AllowedValues: ['enabled', 'disabled', 'enhanced']
```

```
  Description: Enable or disable CloudWatch Container Insights for the cluster
```

```
Resources:
```

```
  # ECS Cluster
```

```
  ECSCluster:
```

```
    Type: AWS::ECS::Cluster
```

```
    Properties:
```

```
      ClusterName: !Ref ECSClusterName
```

```
      ClusterSettings:
```

- Name: containerInsights
- Value: !Ref EnableContainerInsights

```
      Tags:
```

- Key: Name
- Value: !Ref ECSClusterName

```
  # IAM Role for Task Execution
```

```
  ECSTaskExecutionRole:
```

```
    Type: AWS::IAM::Role
```

```
    Properties:
```

```
      AssumeRolePolicyDocument:
```

```
        Version: '2012-10-17'
```

```
        Statement:
```

- Effect: Allow
- Principal:
 - Service: ecs-tasks.amazonaws.com
- Action: sts:AssumeRole

```
      ManagedPolicyArns:
```

- arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy

```
  # Task Definition
```

```
  EBSTaskDefinition:
```

```
    Type: AWS::ECS::TaskDefinition
```

```
    Properties:
```

```
      Family: ebs-task-attach-task-def
```

```
      ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
```

```
      NetworkMode: awsvpc
```

```
      RequiresCompatibilities:
```

```
- EC2
- FARGATE
Cpu: !Ref TaskCpu
Memory: !Ref TaskMemory
ContainerDefinitions:
  - Name: nginx
    Image: !Ref ContainerImage
    Essential: true
    PortMappings:
      - Name: nginx-80-tcp
        ContainerPort: 80
        HostPort: 80
        Protocol: tcp
        AppProtocol: http
    MountPoints:
      - SourceVolume: ebs-vol
        ContainerPath: /foo-container-path
        ReadOnly: false
Volumes:
  - Name: ebs-vol
    ConfiguredAtLaunch: true
```

```
ECSService:
  Type: AWS::ECS::Service
  Properties:
    Cluster: !Ref ECSCluster
    TaskDefinition: !Ref EBSTaskDefinition
    LaunchType: FARGATE
    ServiceName: ebs
    SchedulingStrategy: REPLICA
    DesiredCount: 1
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups: !Ref SecurityGroupIDs
        Subnets: !Ref SubnetIDs
    PlatformVersion: LATEST
    DeploymentConfiguration:
      MaximumPercent: 200
      MinimumHealthyPercent: 100
      DeploymentCircuitBreaker:
        Enable: true
        Rollback: true
    DeploymentController:
```

```

    Type: ECS
    Tags: []
    EnableECSTags: true
    VolumeConfigurations:
      - Name: ebs-vol
        ManagedEBSVolume:
          RoleArn: !Ref InfrastructureRoleArn
          VolumeType: !Ref VolumeType
          Iops: !Ref VolumeIops
          Throughput: !Ref VolumeThroughput
          SizeInGiB: !Ref VolumeSize
          FilesystemType: !Ref FilesystemType
          TagSpecifications:
            - ResourceType: volume
              PropagateTags: TASK_DEFINITION

```

Outputs:

```

ClusterName:
  Description: The cluster used to create the service.
  Value: !Ref ECSCluster
ClusterArn:
  Description: The ARN of the ECS cluster
  Value: !GetAtt ECSCluster.Arn
ECSService:
  Description: The created service.
  Value: !Ref ECSService
TaskDefinitionArn:
  Description: The ARN of the task definition
  Value: !Ref EBSTaskDefinition

```

Implementa il servizio con i fornitori di capacità

Il modello seguente definisce un servizio che utilizza il provider di capacità per richiedere una capacità di AL2 023 su cui eseguire. I contenitori verranno avviati sulle AL2 023 istanze non appena saranno online.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run

```

on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.",

```
"Parameters": {
  "VpcId": {
    "Type": "String",
    "Description": "The VPC that the service is running inside of"
  },
  "PublicSubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of public subnet ID's to put the load balancer in"
  },
  "PrivateSubnetIds": {
    "Type": "List<AWS::EC2::Subnet::Id>",
    "Description": "List of private subnet ID's that the AWS VPC tasks are in"
  },
  "ClusterName": {
    "Type": "String",
    "Description": "The name of the ECS cluster into which to launch
capacity."
  },
  "ECSTaskExecutionRole": {
    "Type": "String",
    "Description": "The role used to start up an ECS task"
  },
  "CapacityProvider": {
    "Type": "String",
    "Description": "The cluster capacity provider that the service should use
to request capacity when it wants to start up a task"
  },
  "ServiceName": {
    "Type": "String",
    "Default": "web",
    "Description": "A name for the service"
  },
  "ImageUrl": {
    "Type": "String",
    "Default": "public.ecr.aws/docker/library/nginx:latest",
    "Description": "The url of a docker image that contains the application
process that will handle the traffic for this service"
  },
  "ContainerCpu": {
    "Type": "Number",
    "Default": 256,
    "Description": "How much CPU to give the container. 1024 is 1 CPU"
```

```

    },
    "ContainerMemory": {
      "Type": "Number",
      "Default": 512,
      "Description": "How much memory in megabytes to give the container"
    },
    "ContainerPort": {
      "Type": "Number",
      "Default": 80,
      "Description": "What port that the application expects traffic on"
    },
    "DesiredCount": {
      "Type": "Number",
      "Default": 2,
      "Description": "How many copies of the service task to run"
    }
  },
  "Resources": {
    "TaskDefinition": {
      "Type": "AWS::ECS::TaskDefinition",
      "Properties": {
        "Family": {
          "Ref": "ServiceName"
        },
        "Cpu": {
          "Ref": "ContainerCpu"
        },
        "Memory": {
          "Ref": "ContainerMemory"
        },
        "NetworkMode": "awsvpc",
        "RequiresCompatibilities": [
          "EC2"
        ],
        "ExecutionRoleArn": {
          "Ref": "ECSTaskExecutionRole"
        },
        "ContainerDefinitions": [
          {
            "Name": {
              "Ref": "ServiceName"
            },
            "Cpu": {
              "Ref": "ContainerCpu"
            }
          }
        ]
      }
    }
  }
}

```

```

    },
    "Memory": {
        "Ref": "ContainerMemory"
    },
    "Image": {
        "Ref": "ImageUrl"
    },
    "PortMappings": [
        {
            "ContainerPort": {
                "Ref": "ContainerPort"
            },
            "HostPort": {
                "Ref": "ContainerPort"
            }
        }
    ],
    "LogConfiguration": {
        "LogDriver": "awslogs",
        "Options": {
            "mode": "non-blocking",
            "max-buffer-size": "25m",
            "awslogs-group": {
                "Ref": "LogGroup"
            },
            "awslogs-region": {
                "Ref": "AWS::Region"
            },
            "awslogs-stream-prefix": {
                "Ref": "ServiceName"
            }
        }
    }
}

]
}

},
"Service": {
    "Type": "AWS::ECS::Service",
    "DependsOn": "PublicLoadBalancerListener",
    "Properties": {
        "ServiceName": {
            "Ref": "ServiceName"
        }
    },

```



```
"Cluster": {
  "Ref": "ClusterName"
},
"PlacementStrategies": [
  {
    "Field": "attribute:ecs.availability-zone",
    "Type": "spread"
  },
  {
    "Field": "cpu",
    "Type": "binpack"
  }
],
"CapacityProviderStrategy": [
  {
    "Base": 0,
    "CapacityProvider": {
      "Ref": "CapacityProvider"
    },
    "Weight": 1
  }
],
"NetworkConfiguration": {
  "AwsVpcConfiguration": {
    "SecurityGroups": [
      {
        "Ref": "ServiceSecurityGroup"
      }
    ],
    "Subnets": {
      "Ref": "PrivateSubnetIds"
    }
  }
},
"DeploymentConfiguration": {
  "MaximumPercent": 200,
  "MinimumHealthyPercent": 75
},
"DesiredCount": {
  "Ref": "DesiredCount"
},
"TaskDefinition": {
  "Ref": "TaskDefinition"
},
},
```

```

        "LoadBalancers": [
            {
                "ContainerName": {
                    "Ref": "ServiceName"
                },
                "ContainerPort": {
                    "Ref": "ContainerPort"
                },
                "TargetGroupArn": {
                    "Ref": "ServiceTargetGroup"
                }
            }
        ]
    },
    "ServiceSecurityGroup": {
        "Type": "AWS::EC2::SecurityGroup",
        "Properties": {
            "GroupDescription": "Security group for service",
            "VpcId": {
                "Ref": "VpcId"
            }
        }
    },
    "ServiceTargetGroup": {
        "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
        "Properties": {
            "HealthCheckIntervalSeconds": 6,
            "HealthCheckPath": "/",
            "HealthCheckProtocol": "HTTP",
            "HealthCheckTimeoutSeconds": 5,
            "HealthyThresholdCount": 2,
            "TargetType": "ip",
            "Port": {
                "Ref": "ContainerPort"
            },
            "Protocol": "HTTP",
            "UnhealthyThresholdCount": 10,
            "VpcId": {
                "Ref": "VpcId"
            },
            "TargetGroupAttributes": [
                {
                    "Key": "deregistration_delay.timeout_seconds",

```

```

        "Value": 0
      }
    ]
  },
  "PublicLoadBalancerSG": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Access to the public facing load balancer",
      "VpcId": {
        "Ref": "VpcId"
      },
      "SecurityGroupIngress": [
        {
          "CidrIp": "0.0.0.0/0",
          "IpProtocol": -1
        }
      ]
    }
  },
  "PublicLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {
      "Scheme": "internet-facing",
      "LoadBalancerAttributes": [
        {
          "Key": "idle_timeout.timeout_seconds",
          "Value": "30"
        }
      ],
      "Subnets": {
        "Ref": "PublicSubnetIds"
      },
      "SecurityGroups": [
        {
          "Ref": "PublicLoadBalancerSG"
        }
      ]
    }
  },
  "PublicLoadBalancerListener": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
      "DefaultActions": [

```

```

        {
            "Type": "forward",
            "ForwardConfig": {
                "TargetGroups": [
                    {
                        "TargetGroupArn": {
                            "Ref": "ServiceTargetGroup"
                        },
                        "Weight": 100
                    }
                ]
            }
        },
        "LoadBalancerArn": {
            "Ref": "PublicLoadBalancer"
        },
        "Port": 80,
        "Protocol": "HTTP"
    },
    "ServiceIngressfromLoadBalancer": {
        "Type": "AWS::EC2::SecurityGroupIngress",
        "Properties": {
            "Description": "Ingress from the public ALB",
            "GroupId": {
                "Ref": "ServiceSecurityGroup"
            },
            "IpProtocol": -1,
            "SourceSecurityGroupId": {
                "Ref": "PublicLoadBalancerSG"
            }
        }
    },
    "LogGroup": {
        "Type": "AWS::Logs::LogGroup"
    }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
```

Description: >-

An example service that deploys in AWS VPC networking mode on EC2 capacity. Service uses a capacity provider to request EC2 instances to run on. Service runs with networking in private subnets, but still accessible to the internet via a load balancer hosted in public subnets.

Parameters:**VpcId:**

Type: String

Description: The VPC that the service is running inside of

PublicSubnetIds:

Type: 'List<AWS::EC2::Subnet::Id>'

Description: List of public subnet ID's to put the load balancer in

PrivateSubnetIds:

Type: 'List<AWS::EC2::Subnet::Id>'

Description: List of private subnet ID's that the AWS VPC tasks are in

ClusterName:

Type: String

Description: The name of the ECS cluster into which to launch capacity.

ECSTaskExecutionRole:

Type: String

Description: The role used to start up an ECS task

CapacityProvider:

Type: String

Description: >-

The cluster capacity provider that the service should use to request capacity when it wants to start up a task

ServiceName:

Type: String

Default: web

Description: A name for the service

ImageUrl:

Type: String

Default: 'public.ecr.aws/docker/library/nginx:latest'

Description: >-

The url of a docker image that contains the application process that will handle the traffic for this service

ContainerCpu:

Type: Number

Default: 256

Description: How much CPU to give the container. 1024 is 1 CPU

ContainerMemory:

Type: Number

Default: 512

Description: How much memory in megabytes to give the container

```
ContainerPort:
  Type: Number
  Default: 80
  Description: What port that the application expects traffic on
DesiredCount:
  Type: Number
  Default: 2
  Description: How many copies of the service task to run
Resources:
  TaskDefinition:
    Type: 'AWS::ECS::TaskDefinition'
    Properties:
      Family: !Ref ServiceName
      Cpu: !Ref ContainerCpu
      Memory: !Ref ContainerMemory
      NetworkMode: awsvpc
      RequiresCompatibilities:
        - EC2
      ExecutionRoleArn: !Ref ECSTaskExecutionRole
      ContainerDefinitions:
        - Name: !Ref ServiceName
          Cpu: !Ref ContainerCpu
          Memory: !Ref ContainerMemory
          Image: !Ref ImageUrl
          PortMappings:
            - ContainerPort: !Ref ContainerPort
              HostPort: !Ref ContainerPort
          LogConfiguration:
            LogDriver: awslogs
            Options:
              mode: non-blocking
              max-buffer-size: 25m
              awslogs-group: !Ref LogGroup
              awslogs-region: !Ref AWS::Region
              awslogs-stream-prefix: !Ref ServiceName
  Service:
    Type: AWS::ECS::Service
    DependsOn: PublicLoadBalancerListener
    Properties:
      ServiceName: !Ref ServiceName
      Cluster: !Ref ClusterName
      PlacementStrategies:
        - Field: 'attribute:ecs.availability-zone'
          Type: spread
```

```

    - Field: cpu
      Type: binpack
CapacityProviderStrategy:
  - Base: 0
    CapacityProvider: !Ref CapacityProvider
    Weight: 1
NetworkConfiguration:
  AwsvpcConfiguration:
    SecurityGroups:
      - !Ref ServiceSecurityGroup
    Subnets: !Ref PrivateSubnetIds
DeploymentConfiguration:
  MaximumPercent: 200
  MinimumHealthyPercent: 75
DesiredCount: !Ref DesiredCount
TaskDefinition: !Ref TaskDefinition
LoadBalancers:
  - ContainerName: !Ref ServiceName
    ContainerPort: !Ref ContainerPort
    TargetGroupArn: !Ref ServiceTargetGroup
ServiceSecurityGroup:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:
    GroupDescription: Security group for service
    VpcId: !Ref VpcId
ServiceTargetGroup:
  Type: 'AWS::ElasticLoadBalancingV2::TargetGroup'
  Properties:
    HealthCheckIntervalSeconds: 6
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 2
    TargetType: ip
    Port: !Ref ContainerPort
    Protocol: HTTP
    UnhealthyThresholdCount: 10
    VpcId: !Ref VpcId
    TargetGroupAttributes:
      - Key: deregistration_delay.timeout_seconds
        Value: 0
PublicLoadBalancerSG:
  Type: 'AWS::EC2::SecurityGroup'
  Properties:

```

```

    GroupDescription: Access to the public facing load balancer
    VpcId: !Ref VpcId
    SecurityGroupIngress:
      - CidrIp: 0.0.0.0/0
        IpProtocol: -1
  PublicLoadBalancer:
    Type: 'AWS::ElasticLoadBalancingV2::LoadBalancer'
    Properties:
      Scheme: internet-facing
      LoadBalancerAttributes:
        - Key: idle_timeout.timeout_seconds
          Value: '30'
      Subnets: !Ref PublicSubnetIds
      SecurityGroups:
        - !Ref PublicLoadBalancerSG
  PublicLoadBalancerListener:
    Type: 'AWS::ElasticLoadBalancingV2::Listener'
    Properties:
      DefaultActions:
        - Type: forward
          ForwardConfig:
            TargetGroups:
              - TargetGroupArn: !Ref ServiceTargetGroup
                Weight: 100
      LoadBalancerArn: !Ref PublicLoadBalancer
      Port: 80
      Protocol: HTTP
  ServiceIngressfromLoadBalancer:
    Type: 'AWS::EC2::SecurityGroupIngress'
    Properties:
      Description: Ingress from the public ALB
      GroupId: !Ref ServiceSecurityGroup
      IpProtocol: -1
      SourceSecurityGroupId: !Ref PublicLoadBalancerSG
  LogGroup:
    Type: 'AWS::Logs::LogGroup'

```

Ruoli IAM per Amazon ECS

Puoi utilizzare AWS CloudFormation modelli per creare ruoli IAM da utilizzare con Amazon ECS. Per ulteriori informazioni sui ruoli IAM per Amazon ECS, consulta [Ruoli IAM per Amazon ECS](#).

Ruolo di esecuzione delle attività di Amazon ECS

Il ruolo di esecuzione delle attività concede al container Amazon ECS e agli agenti Fargate l'autorizzazione a effettuare chiamate AWS API per tuo conto. Il ruolo è richiesto in base ai requisiti dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Il modello seguente può essere utilizzato per creare un semplice ruolo di esecuzione delle attività che utilizza la politica AmazonECSTaskExecutionRolePolicy gestita.

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "CloudFormation template for ECS Task Execution Role",
  "Resources": {
    "ECSTaskExecutionRole": {
      "Type": "AWS::IAM::Role",
      "Properties": {
        "AssumeRolePolicyDocument": {
          "Statement": [
            {
              "Effect": "Allow",
              "Principal": {
                "Service": ["ecs-tasks.amazonaws.com"]
              },
              "Action": ["sts:AssumeRole"],
              "Condition": {
                "ArnLike": {
                  "aws:SourceArn": {
                    "Fn::Sub": "arn:aws:ecs:${AWS::Region}:${AWS::AccountId}:*"
                  }
                },
                "StringEquals": {
                  "aws:SourceAccount": {
                    "Ref": "AWS::AccountId"
                  }
                }
              }
            }
          ]
        }
      }
    }
  ],
  "Path": "/",
  "ManagedPolicyArns": [
```

```

        "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
    ]
}
},
"Outputs": {
    "ECSTaskExecutionRoleARN": {
        "Description": "ARN of the ECS Task Execution Role",
        "Value": {
            "Fn::GetAtt": ["ECSTaskExecutionRole", "Arn"]
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-ECSTaskExecutionRoleARN"
            }
        }
    },
    "ECSTaskExecutionRoleName": {
        "Description": "Name of the ECS Task Execution Role",
        "Value": {
            "Ref": "ECSTaskExecutionRole"
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-ECSTaskExecutionRoleName"
            }
        }
    }
}
}
}

```

YAML

```

AWSTemplateFormatVersion: '2010-09-09'
Description: 'CloudFormation template for ECS Task Execution Role'
Resources:
    ECSTaskExecutionRole:
        Type: AWS::IAM::Role
        Properties:
            AssumeRolePolicyDocument:
                Statement:
                    - Effect: Allow
                      Principal:

```

```

    Service: [ecs-tasks.amazonaws.com]
    Action: ['sts:AssumeRole']
    Condition:
      ArnLike:
        aws:SourceArn: !Sub arn:aws:ecs:${AWS::Region}:${AWS::AccountId}:*
      StringEquals:
        aws:SourceAccount: !Ref AWS::AccountId
    Path: /
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
Outputs:
  ECSTaskExecutionRoleARN:
    Description: ARN of the ECS Task Execution Role
    Value: !GetAtt ECSTaskExecutionRole.Arn
    Export:
      Name: !Sub "${AWS::StackName}-ECSTaskExecutionRoleARN"
  ECSTaskExecutionRoleName:
    Description: Name of the ECS Task Execution Role
    Value: !Ref ECSTaskExecutionRole
    Export:
      Name: !Sub "${AWS::StackName}-ECSTaskExecutionRoleName"

```

Creazione di risorse Amazon ECS utilizzando la console AWS CloudFormation

Un modo per utilizzare Amazon ECS con AWS CloudFormation è tramite AWS Management Console. Qui puoi creare i tuoi AWS CloudFormation stack per i componenti di Amazon ECS come definizioni di attività, cluster e servizi e distribuirli direttamente dalla console. Il seguente tutorial mostra come utilizzare la AWS CloudFormation console per creare un servizio Amazon ECS, una definizione di attività e un cluster.

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i seguenti prerequisiti.

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di [Amazon ECS FullAccess](#) policy IAM.

Fase 1: Creare un modello di stack

Utilizza i seguenti passaggi per creare un modello di AWS CloudFormation stack per un servizio Amazon ECS e altre risorse correlate.

1. Utilizzando un editor di testo a tua scelta, crea un file chiamato `ecs-tutorial-template.yaml`
2. Nel `ecs-tutorial-template.yaml` file, incolla il seguente modello e salva le modifiche.

```
AWSTemplateFormatVersion: 2010-09-09
Description: A template that deploys an application that is built on an Apache
  web server Docker image by creating an Amazon ECS cluster, task definition, and
  service. The template also creates networking and logging resources, and an Amazon
  ECS task execution role.
Parameters:
  ClusterName:
    Type: String
    Default: CFNCluster
    Description: Name of the ECS Cluster
  TaskFamily:
    Type: String
    Default: task-definition-cfn
    Description: Family name for the Task Definition
  ServiceName:
    Type: String
    Default: cfn-service
    Description: Name of the ECS Service
  ContainerImage:
    Type: String
    Default: public.ecr.aws/docker/library/httpd:2.4
    Description: Container image to use for the task
  TaskCpu:
    Type: Number
    Default: 256
    AllowedValues: [256, 512, 1024, 2048, 4096]
    Description: CPU units for the task
  TaskMemory:
    Type: Number
    Default: 512
    AllowedValues: [512, 1024, 2048, 4096, 8192, 16384]
    Description: Memory (in MiB) for the task
  DesiredCount:
```

```
    Type: Number
    Default: 1
    Description: Desired number of tasks to run
LogGroupName:
    Type: String
    Default: /ecs/fargate-task-definition
    Description: CloudWatch Log Group name
VpcCidr:
    Type: String
    Default: 10.0.0.0/16
    Description: CIDR block for the VPC
PublicSubnet1Cidr:
    Type: String
    Default: 10.0.0.0/24
    Description: CIDR block for public subnet 1
PublicSubnet2Cidr:
    Type: String
    Default: 10.0.1.0/24
    Description: CIDR block for public subnet 2
Resources:
  # VPC and Networking Resources
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCidr
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-VPC
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-IGW
  InternetGatewayAttachment:
    Type: AWS::EC2::VPCEGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
  PublicSubnet1:
    Type: AWS::EC2::Subnet
    Properties:
```

```

    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Ref PublicSubnet1Cidr
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicSubnet1
PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PublicSubnet2Cidr
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicSubnet2
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicRouteTable
DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1
PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2
# Security Group
ECSSecurityGroup:
  Type: AWS::EC2::SecurityGroup

```

```

Properties:
  GroupDescription: Security group for ECS tasks
  VpcId: !Ref VPC
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: 443
      ToPort: 443
      CidrIp: 0.0.0.0/0
# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
# CloudWatch Logs
TaskLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    LogGroupName: !Ref LogGroupName
    RetentionInDays: 30
# ECS Resources
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterName: !Ref ClusterName
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Command:
          - >-

```

```

    /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color:
#333;} </style> </head><body> <div
style=color:white;text-align:center> <h1>Amazon ECS Sample
App</h1> <h2>Congratulations!</h2> <p>Your application is now
running on a container in Amazon ECS.</p> </div></body></html>' >
    /usr/local/apache2/htdocs/index.html && httpd-foreground"s
EntryPoint:
  - sh
  - '-c'
Essential: true
Image: !Ref ContainerImage
LogConfiguration:
  LogDriver: awslogs
  Options:
    mode: non-blocking
    max-buffer-size: 25m
    awslogs-create-group: 'true'
    awslogs-group: !Ref LogGroupName
    awslogs-region: !Ref 'AWS::Region'
    awslogs-stream-prefix: ecs
Name: sample-fargate-app
PortMappings:
  - ContainerPort: 80
    HostPort: 80
    Protocol: tcp
Cpu: !Ref TaskCpu
ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
Family: !Ref TaskFamily
Memory: !Ref TaskMemory
NetworkMode: awsvpc
RequiresCompatibilities:
  - FARGATE
RuntimePlatform:
  OperatingSystemFamily: LINUX
ECSService:
  Type: AWS::ECS::Service
  DependsOn:
    - PublicSubnet1RouteTableAssociation
    - PublicSubnet2RouteTableAssociation
  Properties:
    ServiceName: !Ref ServiceName
    Cluster: !Ref ECSCluster
    DesiredCount: !Ref DesiredCount

```



```

    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups:
          - !Ref ECSSecurityGroup
        Subnets:
          - !Ref PublicSubnet1
          - !Ref PublicSubnet2
    TaskDefinition: !Ref ECSTaskDefinition
Outputs:
  ClusterName:
    Description: The name of the ECS cluster
    Value: !Ref ECSCluster
  TaskDefinitionArn:
    Description: The ARN of the task definition
    Value: !Ref ECSTaskDefinition
  ServiceName:
    Description: The name of the ECS service
    Value: !Ref ECSService
  VpcId:
    Description: The ID of the VPC
    Value: !Ref VPC
  PublicSubnet1:
    Description: The ID of public subnet 1
    Value: !Ref PublicSubnet1
  PublicSubnet2:
    Description: The ID of public subnet 2
    Value: !Ref PublicSubnet2
  SecurityGroup:
    Description: The ID of the security group
    Value: !Ref ECSSecurityGroup
  ExecutionRoleArn:
    Description: The ARN of the task execution role
    Value: !GetAtt ECSTaskExecutionRole.Arn

```

Fase 2: creare uno stack per le risorse Amazon ECS

Dopo aver creato un file per il modello, puoi seguire questi passaggi per creare uno stack con il modello utilizzando la console. AWS CloudFormation

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo <https://console.aws.amazon.com/cloudformazione>.
2. Nella pagina Stacks, nell'angolo in alto a destra, scegli Crea stack, quindi scegli con nuove risorse (standard).
3. Scegli un modello esistente.
4. Scegli Carica un file modello, quindi scegli Scegli file per scegliere il `ecs-tutorial-template` file.

Dopo aver caricato il file in un bucket Amazon S3, puoi scegliere View in Infrastructure Composer per visualizzare il modello in Infrastructure Composer. Per ulteriori informazioni sui AWS CloudFormation modelli e Infrastructure Composer, consulta [Creare modelli visivamente con Infrastructure](#) Composer nella Guida per l'utente.AWS CloudFormation

5. Scegli Next (Successivo).
6. Nella pagina Specificare i dettagli dello stack, in Nome dello stack, fornisci il seguente nome per lo stack: `ecs-tutorial-stack` Lasciate tutti i valori dei parametri in Parametri come predefiniti, quindi scegliete Avanti.
7. Nella pagina Configura le opzioni dello stack, in Capacità, seleziona la casella di controllo per confermare AWS CloudFormation la creazione di risorse IAM. Questo riconoscimento è necessario per creare il ruolo di esecuzione delle attività di Amazon ECS come definito nel modello. Lascia le altre impostazioni come predefinite e scegli Avanti.
8. Controlla i dettagli dello stack nella pagina Rivedi e crea, quindi scegli Invia per avviare la creazione dello stack.

Fase 3: Verifica

Utilizza i seguenti passaggi per verificare la creazione di risorse Amazon ECS utilizzando il modello fornito.

1. Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo <https://console.aws.amazon.com/cloudformazione>.
2. Nella pagina Stacks, scegli. `ecs-tutorial-stack`
3. Scegli la scheda Eventi. Se gli stati dell'evento lo dicono `CREATE_IN_PROGRESS`, attendi che la creazione sia completata e che gli stati cambino in. `CREATE_COMPLETE`
4. Dopo aver cambiato gli stati degli eventi **`CREATE_COMPLETE`**, scegli la scheda Risorse. Vedrai le risorse con l'ID logico `ECSCluster`, `ECSTaskDefinition`, rispettivamente `ECSService`.

5. Per verificare la creazione di un cluster Amazon ECS, scegli l'ID fisico associato `ECSCluster`
a. Verrai reindirizzato alla console Amazon ECS dove potrai vedere il cluster creato chiamato `CFNCluster`
6. Per verificare la creazione di un servizio Amazon ECS, scegli l'ID fisico associato `ECSService`
a. Verrai reindirizzato alla console Amazon ECS dove potrai vedere il servizio chiamato `cfn-service` creato nel cluster. `cfnCluster`
7. Per verificare la creazione di una definizione di attività Amazon ECS, scegli l'ID fisico associato `ECSTaskDefinition` a. Verrai reindirizzato alla console Amazon ECS dove potrai vedere la revisione della definizione dell'attività con il nome. `task-definition-cfn`

Fase 4: Eliminazione delle risorse

Per ripulire le risorse ed evitare di incorrere in ulteriori costi, segui questi passaggi.

1. [Accedi AWS Management Console e apri la AWS CloudFormation console all'indirizzo https://console.aws.amazon.com/cloudformazione.](https://console.aws.amazon.com/cloudformazione)
2. Nella pagina Stacks, scegli. `ecs-tutorial-stack`
3. Scegliere Delete (Elimina).
4. Quando viene richiesta la conferma, scegli nuovamente Elimina.
5. Scegli la scheda Eventi. Lo stato delle `ecs-tutorial-stack` modifiche a `DELETE_IN_PROGRESS` e quindi a `DELETE_COMPLETE` dopo l'eliminazione o la cancellazione delle risorse. L'eliminazione richiede un paio di minuti.
6. Scegliere la scheda Resources (Risorse). Verrà ora visualizzato un elenco di ID logici con lo stato aggiornato a `DELETE_COMPLETE`.

Creazione di risorse Amazon ECS utilizzando AWS CLI comandi per AWS CloudFormation

Il seguente tutorial mostra come utilizzare il AWS CLI per creare risorse Amazon ECS con un AWS CloudFormation modello.

Prerequisiti

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).

- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy [Amazon ECS_FullAccess](#) IAM.

Fase 1: Creare uno stack

Per creare uno stack utilizzando il AWS CLI, segui questi passaggi.

1. Usando un editor di testo a tua scelta, crea un file chiamato `ecs-tutorial-template.yaml`.
2. Nel `ecs-tutorial-template.yaml` file, incolla il seguente modello e salva le modifiche.

```
AWSTemplateFormatVersion: 2010-09-09
Description: A template that deploys an application that is built on an Apache
  web server Docker image by creating an Amazon ECS cluster, task definition, and
  service. The template also creates networking and logging resources, and an Amazon
  ECS task execution role.
Parameters:
  ClusterName:
    Type: String
    Default: CFNCluster
    Description: Name of the ECS Cluster
  TaskFamily:
    Type: String
    Default: task-definition-cfn
    Description: Family name for the Task Definition
  ServiceName:
    Type: String
    Default: cfn-service
    Description: Name of the ECS Service
  ContainerImage:
    Type: String
    Default: public.ecr.aws/docker/library/httpd:2.4
    Description: Container image to use for the task
  TaskCpu:
    Type: Number
    Default: 256
    AllowedValues: [256, 512, 1024, 2048, 4096]
    Description: CPU units for the task
  TaskMemory:
    Type: Number
    Default: 512
    AllowedValues: [512, 1024, 2048, 4096, 8192, 16384]
    Description: Memory (in MiB) for the task
```

```
DesiredCount:
  Type: Number
  Default: 1
  Description: Desired number of tasks to run
LogGroupName:
  Type: String
  Default: /ecs/fargate-task-definition
  Description: CloudWatch Log Group name
VpcCidr:
  Type: String
  Default: 10.0.0.0/16
  Description: CIDR block for the VPC
PublicSubnet1Cidr:
  Type: String
  Default: 10.0.0.0/24
  Description: CIDR block for public subnet 1
PublicSubnet2Cidr:
  Type: String
  Default: 10.0.1.0/24
  Description: CIDR block for public subnet 2
Resources:
  # VPC and Networking Resources
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCidr
      EnableDnsSupport: true
      EnableDnsHostnames: true
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-VPC
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub ${AWS::StackName}-IGW
  InternetGatewayAttachment:
    Type: AWS::EC2::VPCGatewayAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
      VpcId: !Ref VPC
  PublicSubnet1:
    Type: AWS::EC2::Subnet
```

```

Properties:
  VpcId: !Ref VPC
  AvailabilityZone: !Select [0, !GetAZs '']
  CidrBlock: !Ref PublicSubnet1Cidr
  MapPublicIpOnLaunch: true
  Tags:
    - Key: Name
      Value: !Sub ${AWS::StackName}-PublicSubnet1
PublicSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PublicSubnet2Cidr
    MapPublicIpOnLaunch: true
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicSubnet2
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub ${AWS::StackName}-PublicRouteTable
DefaultPublicRoute:
  Type: AWS::EC2::Route
  DependsOn: InternetGatewayAttachment
  Properties:
    RouteTableId: !Ref PublicRouteTable
    DestinationCidrBlock: 0.0.0.0/0
    GatewayId: !Ref InternetGateway
PublicSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet1
PublicSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PublicRouteTable
    SubnetId: !Ref PublicSubnet2
# Security Group
ECSSecurityGroup:

```

```
Type: AWS::EC2::SecurityGroup
Properties:
  GroupDescription: Security group for ECS tasks
  VpcId: !Ref VPC
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: 80
      ToPort: 80
      CidrIp: 0.0.0.0/0
    - IpProtocol: tcp
      FromPort: 443
      ToPort: 443
      CidrIp: 0.0.0.0/0
# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
# CloudWatch Logs
TaskLogGroup:
  Type: AWS::Logs::LogGroup
  DeletionPolicy: Retain
  UpdateReplacePolicy: Retain
  Properties:
    LogGroupName: !Ref LogGroupName
    RetentionInDays: 30
# ECS Resources
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterName: !Ref ClusterName
ECSTaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    ContainerDefinitions:
      - Command:
```

```

- >-
  /bin/sh -c "echo '<html> <head> <title>Amazon ECS Sample
  App</title> <style>body {margin-top: 40px; background-color:
  #333;} </style> </head><body> <div
  style=color:white;text-align:center> <h1>Amazon ECS Sample
  App</h1> <h2>Congratulations!</h2> <p>Your application is now
  running on a container in Amazon ECS.</p> </div></body></html>' >
  /usr/local/apache2/htdocs/index.html && httpd-foreground"s
EntryPoint:
- sh
- '-c'
Essential: true
Image: !Ref ContainerImage
LogConfiguration:
  LogDriver: awslogs
  Options:
    mode: non-blocking
    max-buffer-size: 25m
    awslogs-create-group: 'true'
    awslogs-group: !Ref LogGroupName
    awslogs-region: !Ref 'AWS::Region'
    awslogs-stream-prefix: ecs
Name: sample-fargate-app
PortMappings:
- ContainerPort: 80
  HostPort: 80
  Protocol: tcp
Cpu: !Ref TaskCpu
ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
Family: !Ref TaskFamily
Memory: !Ref TaskMemory
NetworkMode: awsvpc
RequiresCompatibilities:
- FARGATE
RuntimePlatform:
  OperatingSystemFamily: LINUX
ECSService:
  Type: AWS::ECS::Service
  DependsOn:
    - PublicSubnet1RouteTableAssociation
    - PublicSubnet2RouteTableAssociation
  Properties:
    ServiceName: !Ref ServiceName
    Cluster: !Ref ECSCluster

```



```

    DesiredCount: !Ref DesiredCount
    LaunchType: FARGATE
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: ENABLED
        SecurityGroups:
          - !Ref ECSSecurityGroup
        Subnets:
          - !Ref PublicSubnet1
          - !Ref PublicSubnet2
    TaskDefinition: !Ref ECSTaskDefinition
Outputs:
  ClusterName:
    Description: The name of the ECS cluster
    Value: !Ref ECSCluster
  TaskDefinitionArn:
    Description: The ARN of the task definition
    Value: !Ref ECSTaskDefinition
  ServiceName:
    Description: The name of the ECS service
    Value: !Ref ECSService
  VpcId:
    Description: The ID of the VPC
    Value: !Ref VPC
  PublicSubnet1:
    Description: The ID of public subnet 1
    Value: !Ref PublicSubnet1
  PublicSubnet2:
    Description: The ID of public subnet 2
    Value: !Ref PublicSubnet2
  SecurityGroup:
    Description: The ID of the security group
    Value: !Ref ECSSecurityGroup
  ExecutionRoleArn:
    Description: The ARN of the task execution role
    Value: !GetAtt ECSTaskExecutionRole.Arn

```

3. Dopo aver creato un file modello, utilizzate il seguente comando per creare uno stack. Il `--capabilities` flag è necessario per creare un ruolo di esecuzione delle attività di Amazon ECS come specificato nel modello. Puoi anche specificare il `--parameters` flag per personalizzare i parametri del modello.

```
aws cloudformation create-stack \
```

```
--stack-name ecs-stack \  
--template-body file://ecs-tutorial-template.yaml \  
--capabilities CAPABILITY_IAM
```

Fase 2: Verificare la creazione delle risorse

Per assicurarti che le risorse vengano create correttamente, segui questi passaggi. Puoi anche controllare la console Amazon ECS.

1. Esegui il comando seguente per elencare tutte le definizioni delle attività in un Regione AWS file.

```
aws ecs list-task-definitions
```

Il comando restituisce un elenco di definizioni di attività Amazon Resource Name (ARN). L'ARN della definizione dell'attività creata utilizzando il modello verrà visualizzato nel formato seguente.

```
{  
  "taskDefinitionArns": [  
    .....  
    "arn:aws:ecs:aws-region:111122223333:task-definition/task-definition-  
cfn:1",  
    .....  
  ]  
}
```

2. Esegui il comando seguente per elencare tutti i cluster in un file. Regione AWS

```
aws ecs list-clusters
```

Il comando restituisce un elenco di cluster ARNs. L'ARN del cluster creato utilizzando il modello verrà visualizzato nel seguente formato.

```
{  
  "clusterArns": [  
    .....  
    "arn:aws:ecs:aws-region:111122223333:cluster/CFNCluster",  
    .....  
  ]  
}
```

3. Esegui il comando seguente per elencare tutti i servizi del cluster `CFNCluster`.

```
aws ecs list-services \
    --cluster CFNCluster
```

Il comando restituisce un elenco di servizi ARNs. L'ARN del servizio creato utilizzando il modello verrà visualizzato nel seguente formato.

```
{
  "serviceArns": [
    "arn:aws:ecs:aws-region:111122223333:service/CFNCluster/cfn-service"
  ]
}
```

Fase 3: pulizia

Per ripulire le risorse che hai creato, esegui il comando seguente.

```
aws cloudformation delete-stack \
    --stack-name ecs-stack
```

Il `delete-stack` comando avvia l'eliminazione dello AWS CloudFormation stack creato in questo tutorial, eliminando tutte le risorse nello stack. Per verificare l'eliminazione, è possibile ripetere la procedura in [Fase 2: Verificare la creazione delle risorse](#). L'elenco degli ARNs output non includerà più una definizione di attività chiamata `task-definition-cfn` o un cluster chiamato `CFNCluster`. La `list-services` chiamata fallirà.

ECS Le migliori pratiche di Amazon

Puoi utilizzare una delle seguenti pagine per conoscere le migliori pratiche operative più importanti per il ECS networking Amazon.

Panoramica delle migliori pratiche	Ulteriori informazioni	
Connect le applicazioni a Internet	Connetti le applicazioni Amazon ECS a Internet	
Ricevi connessioni in entrata verso Amazon ECS da Internet.	Best practice per ricevere connessioni in entrata ad Amazon ECS da Internet	
Connect Amazon ECS ad altri AWS servizi all'interno di un VPC	Le migliori pratiche per connettere Amazon ECS ai AWS servizi dall'interno del tuo VPC	
Servizi di rete tra Account AWS e VPCs	Best practice per il collegamento in rete dei servizi Amazon ECS tra e Account AWS VPCs	
Risolvi i problemi di rete	AWS servizi per la risoluzione dei problemi di rete Amazon ECS	

Puoi utilizzare una qualsiasi delle pagine seguenti per apprendere le best practice operative più importanti per Fargate su Amazon. ECS

Panoramica delle migliori pratiche	Ulteriori informazioni	
Sicurezza Fargate	Le migliori pratiche di sicurezza di Fargate in Amazon ECS	
Considerazioni sulla sicurezza di Fargate	Considerazioni sulla sicurezza di Fargate per Amazon ECS	
Comportamento dell'immagine dei contenitori Linux su Fargate Container Pull	Comportamento dell'immagine dei contenitori Linux su Fargate Container Pull per Amazon ECS	
Comportamento dell'immagine dei contenitori Windows su Fargate Container Pull	Comportamento dell'immagine dei contenitori Windows su Fargate Container Pull per Amazon ECS	
Ritiro delle attività a Fargate	Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS	

È possibile utilizzare una qualsiasi delle pagine seguenti per apprendere le migliori pratiche operative più importanti per la definizione delle attività.

Panoramica delle migliori pratiche	Ulteriori informazioni	
Immagini di container	Best practice per le immagini dei container Amazon ECS	
Dimensioni processo	Le migliori pratiche per le dimensioni delle attività di Amazon ECS	

Panoramica delle migliori pratiche	Ulteriori informazioni	
Le migliori pratiche relative ai volumi	Opzioni di storage per le attività di Amazon ECS	

È possibile utilizzare una delle pagine seguenti per apprendere le migliori pratiche operative più importanti per i cluster e la capacità.

Panoramica delle migliori pratiche	Ulteriori informazioni	
Sicurezza Fargate	Le migliori pratiche di sicurezza di Fargate in Amazon ECS	
EC2considerazioni sulla sicurezza delle istanze dei container	Considerazioni sulla sicurezza delle istanze di EC2 container Amazon per Amazon ECS	
Scalabilità automatica del cluster	Ottimizza la scalabilità automatica dei cluster Amazon ECS	

È possibile utilizzare una qualsiasi delle pagine seguenti per apprendere le migliori pratiche operative più importanti per attività e servizi.

Panoramica delle migliori pratiche	Ulteriori informazioni	
Ottimizza il tempo di avvio delle attività	Ottimizza il tempo di avvio delle attività di Amazon ECS	
Parametri del servizio	Best practice per i parametri del servizio Amazon ECS	

Panoramica delle migliori pratiche	Ulteriori informazioni	
Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico	Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico per Amazon ECS	
Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico	Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS	
Ottimizzazione della scalabilità automatica dei servizi	Ottimizza la scalabilità automatica del servizio Amazon ECS	

È possibile utilizzare una qualsiasi delle pagine seguenti per apprendere le migliori pratiche operative per la sicurezza.

Panoramica delle migliori pratiche	Ulteriori informazioni	
Sicurezza di rete	Best practice di sicurezza di rete per Amazon ECS	
Sicurezza delle attività e dei container	Best practice per la sicurezza di attività e container di Amazon ECS	

AWS Fargate per Amazon ECS

AWS Fargate è una tecnologia che puoi usare con Amazon ECS per eseguire [container](#) senza dover gestire server o cluster di istanze Amazon EC2. Con AWS Fargate, non è più necessario effettuare il provisioning, configurare o scalare i cluster di macchine virtuali per eseguire i container. Viene anche eliminata la necessità di scegliere i tipi di server, di decidere quando dimensionare i cluster o ottimizzarne il packing.

Quando esegui i processi e i servizi con il tipo di avvio Fargate, crei un pacchetto dell'applicazione in container, specifichi i requisiti di CPU e di memoria, definisci le reti e le policy IAM e avvii l'applicazione. Ogni processo Fargate ha un proprio limite di isolamento e non condivide il kernel sottostante, le risorse CPU, le risorse di memoria o l'interfaccia di rete elastica con un altro processo. Configura le definizioni delle attività per Fargate impostando il parametro di definizione delle attività di `requiresCompatibilities` su FARGATE. Per ulteriori informazioni, consulta [Tipi di avvio](#).

Fargate offre versioni della piattaforma per Amazon Linux 2 (versione della piattaforma 1.3.0), il sistema operativo Bottlerocket (versione della piattaforma 1.4.0) e le edizioni Microsoft Windows 2019 Server Full e Core. Salvo diversa indicazione, le informazioni contenute in questa pagina si applicano a tutte le piattaforme Fargate.

In questo argomento vengono descritti i diversi componenti dei processi e dei servizi Fargate oltre a particolari considerazioni sull'utilizzo di Fargate con Amazon ECS.

Per informazioni sulle regioni che supportano container Linux su Fargate, consulta [the section called "Contentori Linux su AWS Fargate"](#).

Per informazioni sulle regioni che supportano container Windows su Fargate, consulta [the section called "Contentori Windows su AWS Fargate"](#).

Procedure guidate

Per informazioni su come iniziare a utilizzare la console, consulta:

- [Scopri come creare un'attività Amazon ECS Linux per il tipo di lancio Fargate](#)
- [Scopri come creare un'attività Amazon ECS Windows per il tipo di lancio Fargate](#)

Per informazioni su come iniziare a utilizzare il AWS CLI, consulta:

- [Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI](#)
- [Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI](#)

Provider di capacità

Sono disponibili i seguenti provider di capacità:

- Fargate
- Fargate Spot - Esegui attività Amazon ECS tolleranti alle interruzioni a una tariffa scontata rispetto al AWS Fargate prezzo. Fargate Spot esegue le attività nella capacità di elaborazione di riserva. Quando sarà AWS necessario ripristinare la capacità, le attività verranno interrotte con un avviso di due minuti. Per ulteriori informazioni, consulta [Cluster Amazon ECS per Fargate](#).

Definizioni di processi

I processi che utilizzano il tipo di avvio Fargate non supportano tutti i parametri di definizione dei processi di Amazon ECS disponibili. Alcuni parametri non sono supportati, mentre altri si comportano diversamente con i processi Fargate. Per ulteriori informazioni, consulta [CPU e memoria del processo](#).

Versioni della piattaforma

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Si tratta di una combinazione delle versioni del kernel e del runtime del container. La versione della piattaforma viene selezionata quando si esegue un'attività o quando si crea un servizio per mantenere una serie di attività identiche.

Nuove versioni della piattaforma vengono rilasciate con l'evolvere dell'ambiente di runtime, ad esempio se vengono introdotti aggiornamenti relativi al kernel o al sistema operativo, nuove funzionalità, correzioni di bug o aggiornamenti della sicurezza. La versione della piattaforma Fargate viene aggiornata attraverso una nuova revisione della versione della piattaforma. Ogni attività viene eseguita su una revisione della versione della piattaforma durante il suo ciclo di vita. Se desideri utilizzare l'ultima revisione della versione della piattaforma, devi avviare una nuova attività. Una nuova attività in esecuzione su Fargate viene eseguita sempre sulla versione della piattaforma più recente. Ciò garantisce che le attività vengano sempre avviate su un'infrastruttura sicura e con patch applicate.

Se viene rilevato un problema di sicurezza che riguarda una versione della piattaforma esistente, AWS crea una nuova revisione con patch della versione della piattaforma e annulla le attività in esecuzione sulla revisione vulnerabile. In alcuni casi viene inviata una notifica in merito alla programmazione del ritiro delle attività su Fargate. Per ulteriori informazioni, consulta [Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS](#).

Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Bilanciamento del carico nel servizio

Il tuo servizio Amazon ECS su AWS Fargate può essere configurato opzionalmente per utilizzare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività del servizio.

Servizi Amazon ECS su AWS Fargate supportano i tipi Application Load Balancer, Network Load Balancer e load balancer. Gli Application Load Balancer sono utilizzati per instradare il traffico HTTP/HTTPS (o livello 7). I Network Load Balancer sono utilizzati per instradare il traffico TCP o UDP (o livello 4). Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

Inoltre, quando crei gruppi di destinazione per questi servizi, devi scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'interfaccia di rete elastica, non a un' EC2 istanza Amazon. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

Utilizzo di un Network Load Balancer per indirizzare il traffico UDP verso Amazon ECS su AWS Fargate le attività sono supportate solo quando si utilizza la versione 1.4 o successiva della piattaforma.

Parametri di utilizzo

Puoi utilizzare le metriche di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse del tuo account. Utilizza queste metriche per visualizzare l'utilizzo corrente del servizio su CloudWatch grafici e dashboard.

AWS Fargate le metriche di utilizzo corrispondono alle quote di servizio. AWS È possibile configurare gli allarmi che avvertono quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sull'AWS Fargate quote di servizio, [endpoint Amazon ECS e quote](#) nel.. Riferimenti generali di Amazon Web Services

Per ulteriori informazioni sui AWS Fargate parametri di utilizzo, consulta [AWS Fargate metriche](#) di utilizzo.

Considerazioni sulla sicurezza di Amazon ECS su quando utilizzare il tipo di lancio Fargate

Consigliamo ai clienti che cercano un forte isolamento per le proprie attività di utilizzare Fargate. Fargate esegue ogni attività in un ambiente di virtualizzazione hardware. Ciò garantisce che questi carichi di lavoro containerizzati non condividano interfacce di rete, storage temporaneo Fargate, CPU o memoria con altre attività. [Per ulteriori informazioni, vedere Panoramica sulla sicurezza di AWS Fargate](#)

Le migliori pratiche di sicurezza di Fargate in Amazon ECS

Consigliamo di tenere in considerazione le best practice seguenti quando utilizzi AWS Fargate. Per ulteriori indicazioni, vedere [Panoramica sulla sicurezza di AWS Fargate](#)

Utilizzato AWS KMS per crittografare lo storage temporaneo per Fargate

È necessario che lo storage temporaneo sia crittografato con una delle chiavi gestite dal cliente o con le proprie chiavi gestite dal cliente. AWS KMS Per le attività ospitate su Fargate utilizzando una versione della piattaforma 1.4.0 o successiva, ogni attività riceve 20 GiB di spazio di archiviazione temporaneo. Per ulteriori informazioni, consulta [Customer Managed Key \(CMK\)](#). La quantità totale di spazio di archiviazione temporaneo può essere aumentata, fino ad un massimo di 200 GiB, specificando il parametro `ephemeralStorage` nella definizione di attività. Per tali attività lanciate il 28 maggio 2020 o successivamente, l'archiviazione temporanea è crittografata con un algoritmo di crittografia AES-256 utilizzando una chiave di crittografia gestita da Fargate.

Per ulteriori informazioni, consulta [Opzioni di storage per le attività di Amazon ECS](#).

Esempio: avvio di un'attività sulla piattaforma Fargate versione 1.4.0 con crittografia dello storage temporaneo

Il comando seguente avvierà un'operazione sulla piattaforma Fargate versione 1.4. Poiché questa attività viene avviata come parte del cluster, utilizza 20 GiB di storage temporaneo crittografato automaticamente.

```
aws ecs run-task --cluster clustername \
```

```
--task-definition taskdefinition:version \  
--count 1  
--launch-type "FARGATE" \  
--platform-version 1.4.0 \  
--network-configuration  
"awsvpcConfiguration={subnets=[subnetid],securityGroups=[securitygroupid]}" \  
--region region
```

Funzionalità SYS_PTRACE per il tracciamento delle syscall del kernel con Fargate

La configurazione predefinita delle funzionalità di Linux che vengono aggiunte o rimosse dal container è fornita da Docker.

Le attività avviate su Fargate supportano solo l'aggiunta della funzionalità del kernel SYS_PTRACE.

[Il video seguente mostra come utilizzare questa funzionalità attraverso il progetto Sysdig Falco.](#)

[# ContainersFromTheCouch - Risoluzione dei problemi relativi all'attività di Fargate utilizzando la funzionalità SYS_PTRACE](#)

[Il codice discusso nel video precedente può essere trovato qui. GitHub](#)

Usa Amazon GuardDuty con Fargate Runtime Monitoring

Amazon GuardDuty è un servizio di rilevamento delle minacce che aiuta a proteggere account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente. Utilizzando modelli di machine learning (ML) e funzionalità di rilevamento di anomalie e minacce, monitora GuardDuty continuamente diverse fonti di log e attività di runtime per identificare e dare priorità ai potenziali rischi per la sicurezza e alle attività dannose nel tuo ambiente.

Runtime Monitoring in GuardDuty protegge i carichi di lavoro in esecuzione su Fargate AWS monitorando continuamente i log e l'attività di rete per identificare comportamenti dannosi o non autorizzati. Runtime Monitoring utilizza un agente di GuardDuty sicurezza leggero e completamente gestito che analizza il comportamento sull'host, come l'accesso ai file, l'esecuzione dei processi e le connessioni di rete. Ciò riguarda questioni quali l'aumento dei privilegi, l'uso di credenziali esposte o la comunicazione con indirizzi IP e domini dannosi e la presenza di malware sulle istanze Amazon EC2 e sui carichi di lavoro dei container. [Per ulteriori informazioni, consulta GuardDuty Runtime Monitoring nella Guida per l'utente. GuardDuty](#)

Considerazioni sulla sicurezza di Fargate per Amazon ECS

Ogni attività dispone di una capacità di infrastruttura dedicata, perché Fargate esegue ciascun carico di lavoro su un ambiente virtuale isolato. I carichi di lavoro eseguiti su Fargate non condividono interfacce di rete, spazio di archiviazione temporaneo, CPU o memoria con altre attività. È possibile eseguire più container all'interno di un'attività, inclusi container di applicazioni e container sidecar (o semplicemente sidecar). Un sidecar è un container che viene eseguito parallelamente al container di un'applicazione in un'attività Amazon ECS. Sebbene il container dell'applicazione esegua il codice applicativo di base, i processi in esecuzione nei sidecar possono potenziare l'applicazione. I sidecar consentono di separare le funzioni dell'applicazione in container dedicati e di semplificare così l'aggiornamento di parti dell'applicazione.

I container che fanno parte della stessa attività condividono le risorse per il tipo di lancio Fargate, perché verranno sempre eseguiti sullo stesso host e condivideranno le risorse di elaborazione. Questi container condividono anche l'archiviazione temporanea fornita da Fargate. I container Linux in un'attività condividono gli spazi dei nomi di rete, inclusi l'indirizzo IP e le porte di rete. All'interno di un'attività, i container che appartengono a tale attività possono comunicare tra loro attraverso localhost.

L'ambiente di runtime di Fargate impedisce l'utilizzo di determinate funzionalità del controller supportate nelle istanze EC2. Quando progetti carichi di lavoro in esecuzione su Fargate, tieni presente gli aspetti seguenti:

- **Nessun container o accesso privilegiato:** funzionalità come i container o l'accesso privilegiato non sono attualmente disponibili su Fargate. Ciò influirà su casi d'uso come l'esecuzione di Docker in Docker.
- **Accesso limitato alle funzionalità di Linux:** l'ambiente in cui i container vengono eseguiti su Fargate è bloccato. Le funzionalità aggiuntive di Linux, come `CAP_SYS_ADMIN` e `CAP_NET_ADMIN`, sono limitate per impedire un'escalation dei privilegi. Fargate supporta l'aggiunta della funzionalità [CAP_SYS_PTRACE](#) di Linux alle attività per consentire agli strumenti di osservabilità e sicurezza implementati all'interno dell'attività di monitorare l'applicazione containerizzata.
- **Nessun accesso all'host sottostante:** né i clienti né AWS gli operatori possono connettersi a un host che esegue i carichi di lavoro dei clienti. Puoi utilizzare ECS exec per eseguire comandi in oppure ottenere uno shell (interprete di comandi) per un container in esecuzione su Fargate. Puoi utilizzare ECS exec per raccogliere informazioni diagnostiche per il debug. Fargate impedisce inoltre ai container di accedere alle risorse dell'host sottostante, come il file system, i dispositivi, le reti e il runtime di container.

- Rete: è possibile utilizzare i gruppi di sicurezza e la rete ACLs per controllare il traffico in entrata e in uscita. Le attività Fargate ricevono un indirizzo IP dalla sottorete configurata nel VPC.

Versioni della piattaforma Fargate per Amazon ECS

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Si tratta di una combinazione delle versioni del kernel e del runtime del container. La versione della piattaforma viene selezionata quando si esegue un'attività o quando si crea un servizio per mantenere una serie di attività identiche.

Nuove versioni della piattaforma vengono rilasciate con l'evolvere dell'ambiente di runtime, ad esempio se vengono introdotti aggiornamenti relativi al kernel o al sistema operativo, nuove funzionalità, correzioni di bug o aggiornamenti della sicurezza. La versione della piattaforma Fargate viene aggiornata attraverso una nuova revisione della versione della piattaforma. Ogni attività viene eseguita su una revisione della versione della piattaforma durante il suo ciclo di vita. Se desideri utilizzare l'ultima revisione della versione della piattaforma, devi avviare una nuova attività. Una nuova attività in esecuzione su Fargate viene eseguita sempre sulla versione della piattaforma più recente. Ciò garantisce che le attività vengano sempre avviate su un'infrastruttura sicura e con patch applicate.

Se viene rilevato un problema di sicurezza che riguarda una versione della piattaforma esistente, AWS crea una nuova revisione con patch della versione della piattaforma e annulla le attività in esecuzione sulla revisione vulnerabile. In alcuni casi viene inviata una notifica in merito alla programmazione del ritiro delle attività su Fargate. Per ulteriori informazioni, consulta [Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS](#).

La versione della piattaforma viene specificata quando si esegue un'attività o si distribuisce un servizio.

Tieni presente le considerazioni seguenti quando specifichi una versione della piattaforma:

- È possibile specificare un numero di versione specifico, ad esempio `1.4.0`, o `LATEST`.

L'ULTIMA versione della piattaforma Linux è `1.4.0`.

L'ULTIMA versione della piattaforma Windows è `1.0.0`.

- Se desideri aggiornare la versione della piattaforma per un servizio, crea un'implementazione. Ad esempio, supponiamo che tu disponga di un servizio che esegue attività nella versione della piattaforma Linux `1.3.0`. Per modificare il servizio in modo che esegua attività sulla versione della

piattaforma Linux 1.4.0, è necessario aggiornare il servizio e specificare una nuova versione della piattaforma. Le attività vengono implementate nuovamente con la versione e la revisione della versione della piattaforma più recenti. Per ulteriori informazioni sulle implementazioni, consulta [Servizi Amazon ECS](#).

- Se la capacità del servizio viene ridotta senza aggiornare la versione della piattaforma, le attività ricevono la versione della piattaforma specificata nell'implementazione corrente del servizio. Ad esempio, supponiamo che tu disponga di un servizio che esegue attività nella versione della piattaforma Linux 1.3.0. Se aumenti il conteggio desiderato del servizio, il pianificatore del servizio avvia le nuove attività utilizzando la revisione più recente della versione della piattaforma 1.3.0.
- Le nuove attività vengono sempre eseguite sulla revisione più recente di una versione della piattaforma. Ciò garantisce che le attività siano sempre eseguite su un'infrastruttura protetta e dotata di patch.
- I numeri di versione della piattaforma per i container Linux e Windows su Fargate sono indipendenti. Ad esempio, il comportamento, le funzionalità e il software utilizzati nella versione della piattaforma 1.0.0 per i container Windows su Fargate non sono paragonabili a quelli della versione della piattaforma 1.0.0 per i container Linux su Fargate.
- Quanto segue si applica alle versioni della piattaforma Fargate Windows.

Le immagini del container Microsoft Windows Server devono essere create da una versione specifica di Windows Server. Devi selezionare la stessa versione di Windows Server in `platformFamily` quando esegui un'attività o crei un servizio che corrisponde all'immagine del container di Windows Server. Inoltre, puoi fornire una `operatingSystemFamily` corrispondente nella definizione delle attività per evitare che vengano eseguite sulla versione di Windows errata. Per ulteriori informazioni, consulta [Corrispondenza della versione dell'host del container con le versioni delle immagini del container](#) sul sito Web di Microsoft Learn.

Migrazione alla versione 1.4.0 della piattaforma Linux su Amazon ECS

Quando si esegue la migrazione dei processi Amazon ECS su Fargate dalla versione della piattaforma 1.0.0, 1.1.0, 1.2.0 o 1.3.0 alla piattaforma 1.4.0 è necessario considerare quanto segue. È consigliabile verificare che l'attività funzioni correttamente sulla versione della piattaforma 1.4.0 prima di migrare le attività.

- Il comportamento del traffico di rete da e verso le attività è stato aggiornato. A partire dalla versione 1.4.0 della piattaforma, tutte le attività di Amazon ECS on Fargate ricevono un'unica interfaccia

di rete elastica (denominata task ENI) e tutto il traffico di rete fluisce attraverso tale ENI all'interno del tuo VPC. Il traffico è visibile tramite i log di flusso del VPC. Per ulteriori informazioni, consulta [Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#).

- Se utilizzi gli endpoint VPC dell'interfaccia, considera quanto segue.
 - Per le immagini dei container ospitate con Amazon ECR, sono necessari i seguenti endpoint. Per ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia Amazon ECR \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.
 - com.amazonaws. **region**.ecr.dkr Endpoint VPC Amazon ECR
 - com.amazonaws. **region**.ecr.api Endpoint VPC Amazon ECR
 - Endpoint gateway Amazon S3
 - Quando la definizione dell'attività fa riferimento ai segreti di Secrets Manager per recuperare dati sensibili per i contenitori, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta [Utilizzo di Secrets Manager con endpoint VPC](#) nella Guida per l'utente di AWS Secrets Manager .
 - Quando la definizione dell'attività fa riferimento ai parametri di Systems Manager Parameter Store per recuperare dati sensibili per i contenitori, è necessario creare gli endpoint VPC di interfaccia per Systems Manager. Per ulteriori informazioni, consulta [Migliorare la sicurezza delle EC2 istanze utilizzando gli endpoint VPC per Systems Manager](#) nella AWS Systems Manager Guida per l'utente.
- Il gruppo di sicurezza per l'Elastic Network Interface (ENI) associato all'attività necessita delle regole del gruppo di sicurezza per consentire il traffico tra l'attività e gli endpoint VPC.

Registro delle modifiche della versione della piattaforma Fargate Linux

Di seguito sono riportate le versioni della piattaforma Linux disponibili. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta [AWS Deprecazione della versione della piattaforma Fargate Linux](#).

1.4.0

Di seguito è riportato il changelog per la versione della piattaforma 1.4.0.

- A partire dal 5 novembre 2020, qualsiasi nuovo processo Amazon ECS avviato su Fargate tramite la versione della piattaforma 1.4.0 potrà utilizzare le seguenti funzioni:
 - Quando utilizzi Secrets Manager per archiviare dati sensibili, è possibile inserire una chiave JSON specifica o una versione specifica di un segreto come variabile di ambiente o in una

configurazione di log. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

- Specifica le variabili di ambiente in blocco utilizzando il parametro di definizione del container `environmentFiles`. Per ulteriori informazioni, consulta [Passa una singola variabile di ambiente a un contenitore Amazon ECS](#).
- Alle attività eseguite in un VPC e in una sottorete abilitata per IPv6 verranno assegnati sia un IPv4 indirizzo privato che un indirizzo IPv6. Per ulteriori informazioni, consulta le [opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#).
- L'endpoint dei metadati dei processi versione 4 fornisce metadati aggiuntivi relativi al processo e al container, inclusi il tipo di avvio del processo, l'Amazon Resource Name (ARN) del container e il driver di log e le relative opzioni utilizzate. Quando si esegue una query sull'endpoint `/stats` si ricevono anche le statistiche sulla velocità di rete per i container. Per ulteriori informazioni, consulta [Task metadata endpoint versione 4](#).
- A partire dal 30 luglio 2020, qualsiasi nuovo processo Amazon ECS avviato su Fargate tramite la versione della piattaforma 1.4.0 sarà in grado di instradare il traffico UDP utilizzando un Network Load Balancer ai suoi processi Amazon ECS su Fargate. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).
- A partire dal 28 maggio 2020, qualsiasi nuova attività Amazon ECS lanciata su Fargate utilizzando la 1.4.0 versione della piattaforma avrà lo storage temporaneo crittografato con un algoritmo di crittografia AES-256 utilizzando una chiave di crittografia proprietaria. AWS Per ulteriori informazioni, consultare [Archiviazione effimera delle attività Fargate per Amazon ECS](#) e [Opzioni di storage per le attività di Amazon ECS](#).
- Aggiunto il supporto per l'utilizzo dei volumi del file system Amazon EFS per l'archiviazione dei processi persistenti. Per ulteriori informazioni, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).
- Lo storage temporaneo delle attività è stato aumentato fino a un minimo di 20 GB per ogni attività. Per ulteriori informazioni, consulta [Archiviazione effimera delle attività Fargate per Amazon ECS](#).
- Il comportamento del traffico di rete da e verso le attività è stato aggiornato. A partire dalla versione 1.4.0 della piattaforma, tutti i processi Fargate ricevono un'unica interfaccia di rete elastica (denominata ENI di processo) e tutto il traffico di rete scorre attraverso tale ENI all'interno del VPC e sarà visibile all'utente attraverso i log di flusso del VPC. Per ulteriori informazioni sul networking per il tipo di EC2 avvio di Amazon, consulta le [opzioni di task networking di Amazon ECS per il tipo di EC2 avvio](#). Per ulteriori informazioni sulla rete per il tipo di lancio Fargate, vedere. [Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#)

- Task ENIs aggiunge il supporto per i jumbo frame. Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimensione del payload più grande che si adatta all'interno di un singolo frame. Più grande è l'MTU, più il payload dell'applicazione può essere adattato all'interno di un singolo fotogramma, riducendo il sovraccarico per fotogramma e aumentando l'efficienza. Il supporto dei frame jumbo riduce il sovraccarico quando il percorso di rete tra l'attività e la destinazione supporta frame jumbo, ad esempio tutto il traffico che rimane all'interno del VPC.
- CloudWatch Container Insights includerà metriche delle prestazioni di rete per le attività di Fargate. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#).
- Aggiunto il supporto per l'endpoint dei metadati dei processi versione 4 che fornisce informazioni aggiuntive per i processi Fargate, incluse le statistiche di rete per il processo e la zona di disponibilità in cui il processo è in esecuzione. Per ulteriori informazioni, consultare [Endpoint di metadati delle attività Amazon ECS versione 4](#) e [Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate](#).
- Aggiunto il supporto per il parametro Linux SYS_PTRACE nelle definizioni dei container. Per ulteriori informazioni, consulta [Parametri Linux](#).
- L'agente del container Fargate sostituisce l'uso dell'agente del container Amazon ECS per tutti i processi di Fargate. Spesso, questa modifica non dovrebbe avere effetto sull'esecuzione delle attività.
- Il runtime del container ora utilizza Containerd invece di Docker. Più frequentemente, questa modifica non dovrebbe avere effetto sull'esecuzione delle attività. Si noterà che alcuni messaggi di errore che hanno origine dal runtime del container passano dal menzionare Docker a errori più generici. Per ulteriori informazioni, consulta i [messaggi di errore dell'attività interrotta di Amazon ECS](#).
- Basato su Amazon Linux 2.

1.3.0

Di seguito è riportato il changelog per la versione della piattaforma 1.3.0.

- A partire dal 30 settembre 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log `awsfirelens`. Configura Amazon ECS FireLens per utilizzare i parametri di definizione delle attività per indirizzare i log verso un AWS servizio o una destinazione AWS Partner Network (APN) per l'archiviazione e l'analisi dei log. Per ulteriori informazioni, consulta [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

- Aggiunta la funzione di riavvio dei processi per i processi Fargate, che corrisponde al processo di aggiornamento dei processi che sono parte di un servizio Amazon ECS. Per ulteriori informazioni, consulta [Task retirement and maintenance for AWS Fargate su Amazon ECS](#).
- A partire dal 27 marzo 2019, ogni nuovo processo Fargate che viene avviato può utilizzare ulteriori parametri di definizione dei processi che consentono di definire una configurazione proxy, le dipendenze per l'avvio e l'arresto di container, nonché un valore di timeout di avvio e arresto per container. Per ulteriori informazioni, consulta [Configurazione del proxy](#), [Dipendenze per i container](#) e [Timeout del container](#).
- A partire dal 2 aprile 2019, qualsiasi nuova attività di Fargate che verrà lanciata supporta l'iniezione di dati sensibili nei contenitori archiviando i dati sensibili in AWS Secrets Manager segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).
- A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta il riferimento ai dati sensibili nella configurazione dei log di un container utilizzando il parametro di definizione del container `secretOptions`. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).
- A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log `sp1unk` in aggiunta ai driver di log `aws1ogs`. Per ulteriori informazioni, consulta [Archiviazione e registrazione](#).
- A partire dal 9 luglio 2019, tutte le nuove attività di Fargate lanciate supportano CloudWatch Container Insights. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#).
- A partire dal 3 dicembre 2019, è supportato il Capacity Provider Fargate Spot. Per ulteriori informazioni, consulta [Cluster Amazon ECS per Fargate](#).
- Basato su Amazon Linux 2.

AWS Deprecazione della versione della piattaforma Fargate Linux

Questa pagina elenca le versioni della piattaforma Linux che AWS Fargate ha reso obsolete o per le quali è stata pianificata la loro obsolescenza. Queste versioni della piattaforma rimarranno disponibili fino alla data di dichiarazione come obsolete pubblicata.

Una data di aggiornamento forzato viene fornita per ogni versione della piattaforma pianificata per l'interruzione dei processi. Alla data di aggiornamento forzato, qualsiasi servizio che utilizza la

versione della piattaforma LATEST che punta a una versione della piattaforma che è stata dichiarata come obsoleta verrà aggiornata utilizzando l'opzione Forza nuova implementazione. Quando il servizio viene aggiornato utilizzando l'opzione Forza nuova implementazione, tutti i processi in esecuzione su una versione della piattaforma pianificata per la dichiarazione come obsoleta vengono arrestati e vengono avviati nuovi processi che utilizzano la versione della piattaforma a cui punta il tag LATEST in quel momento. I processi o i servizi autonomi con una versione esplicita della piattaforma impostata non sono interessati dalla data di aggiornamento forzato.

Consigliamo di aggiornare i processi autonomi dei servizi in modo da utilizzare la versione più recente della piattaforma. Per ulteriori informazioni sulla migrazione alla versione più recente della piattaforma, consulta [Migrazione alla versione 1.4.0 della piattaforma Linux su Amazon ECS](#).

Una volta che la versione della piattaforma raggiunge la data di ritiro, la versione della piattaforma non sarà più disponibile per nuovi processi o servizi. Qualsiasi processo o servizio autonomo che utilizza esplicitamente una versione della piattaforma dichiarata obsoleta continuerà a utilizzare tale versione della piattaforma fino a quando i processi non saranno arrestati. Dopo la data di ritiro, una versione della piattaforma dichiarata obsoleta non riceverà più aggiornamenti di sicurezza o correzioni di bug.

Versione della piattaforma	Data di aggiornamento forzato	Data di ritiro
1.0.0	26 ottobre 2020	14 dicembre 2020
1.1.0	26 ottobre 2020	14 dicembre 2020
1.2.0	26 ottobre 2020	14 dicembre 2020

Per informazioni sulle versioni correnti della piattaforma, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Registro delle modifiche delle versioni obsolete di Fargate Linux

1.2.0

Di seguito è riportato il changelog per la versione della piattaforma 1.2.0.

 Note

La versione della piattaforma 1.2.0 non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta [AWS Deprecazione della versione della piattaforma Fargate Linux](#).

- È stato aggiunto il supporto per l'autenticazione del registro privato tramite AWS Secrets Manager. Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

1.1.0

Di seguito è riportato il changelog per la versione della piattaforma 1.1.0.

 Note

La versione della piattaforma 1.1.0 non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta [AWS Deprecazione della versione della piattaforma Fargate Linux](#).

- Aggiunto il supporto per gli endpoint dei metadati dei processi di Amazon ECS. Per ulteriori informazioni, consulta [Metadati delle attività di Amazon ECS disponibili per le attività su Fargate](#).
- Aggiunta del supporto per i controlli dello stato Docker nelle definizioni dei container. Per ulteriori informazioni, consulta [Controllo dello stato](#).
- Aggiunto il supporto per l'individuazione dei servizi di Amazon ECS. Per ulteriori informazioni, consulta [Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS](#).

1.0.0

Di seguito è riportato il changelog per la versione della piattaforma 1.0.0.

 Note

La versione della piattaforma 1.0.0 non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta [AWS Deprecazione della versione della piattaforma Fargate Linux](#).

- Basato su Amazon Linux 2017.09
- Versione iniziale.

Registro delle modifiche della versione della piattaforma Fargate Windows

Di seguito sono riportate le versioni della piattaforma disponibili per i container Windows.

1.0.0

Di seguito è riportato il changelog per la versione della piattaforma 1.0.0.

- Versione iniziale per il supporto sui seguenti sistemi operativi Microsoft Windows Server:
 - Windows Server 2019 Full
 - Windows Server 2019 Core
 - Windows Server 2022 Full
 - Windows Server 2022 Core

Contenitori Windows su Fargate: considerazioni per Amazon ECS

Di seguito sono riportate le differenze e le considerazioni da tenere presente quando si eseguono contenitori Windows su AWS Fargate.

Se è necessario eseguire attività su contenitori Linux e Windows, è necessario creare definizioni di attività separate per ciascun sistema operativo.

AWS gestisce la gestione delle licenze del sistema operativo, quindi non sono necessarie licenze Microsoft Windows Server aggiuntive.

I contenitori Windows su AWS Fargate supportano i seguenti sistemi operativi:

- Windows Server 2019 Full
- Windows Server 2019 Core
- Windows Server 2022 Full
- Windows Server 2022 Core

I contenitori Windows su AWS Fargate supportano il driver awslogs. Per ulteriori informazioni, consulta [the section called “Invia i log a CloudWatch”](#).

Le seguenti funzionalità non sono supportate nei container Windows su Fargate:

- Amazon FSx
- Trunking ENI
- g per Windows Containers MSAs
- Servizio App Mesh e integrazione proxy per i processi
- Integrazione del router di log Firelens per i processi
- Volumi EFS
- Volumi EBS
- I parametri di definizione delle attività riportati di seguito:
 - `maxSwap`
 - `swappiness`
 - `environmentFiles`
- Il provider di capacità Fargate Spot
- Volumi di immagine

L'opzione Dockerfile `volume` viene ignorata. Invece, usa montaggi vincolati nella tua definizione di attività. Per ulteriori informazioni, consulta [Usa bind mount con Amazon ECS](#).

- I parametri di CPU e memoria a livello di attività vengono ignorati per i contenitori Windows. Ti consigliamo di specificare risorse a livello di container per i container Windows.
- memoria per l'attività
- `MemoryReservation` sui contenitori
- Riavvia le politiche sui contenitori

Archiviazione effimera delle attività Fargate per Amazon ECS

Una volta eseguito il provisioning, ogni attività Amazon ECS ospitata su container Linux AWS Fargate riceve il seguente storage temporaneo per i bind mount. Può essere montato e condiviso tra container che utilizzano i parametri `volumes`, `mountPoints` e `volumesFrom` nella definizione di attività. Questa funzionalità non è supportata per i contenitori Windows su AWS Fargate.

Versioni della piattaforma container Linux Fargate

Versione 1.4.0 o successiva

Per impostazione predefinita, le attività Amazon ECS ospitate su Fargate che utilizzano la piattaforma versione 1.4.0 o successiva ricevono almeno 20 GiB di storage temporaneo. La quantità totale di storage temporaneo può essere aumentata, fino a un massimo di 200 GiB. Puoi eseguire questa operazione specificando il parametro `ephemeralStorage` nella definizione di attività.

L'immagine del container estratta, compressa e non compressa per il processo viene memorizzata nell'archiviazione temporanea. Per determinare la quantità totale di archiviazione temporanea che il processo deve utilizzare, è necessario sottrarre la quantità di archiviazione utilizzata dall'immagine del container dall'importo totale di archiviazione temporanea in cui è allocato il processo.

Per le attività che utilizzano la piattaforma versione 1.4.0 o successiva avviati il 28 maggio 2020 o successivamente, lo storage temporaneo viene crittografato con un algoritmo di crittografia AES-256. Questo algoritmo utilizza una chiave AWS di crittografia proprietaria oppure è possibile creare una chiave gestita dal cliente. Per ulteriori informazioni, consulta [Customer managed keys for AWS Fargate ephemeral storage](#).

Per le attività che utilizzano la piattaforma versione 1.4.0 o successiva, avviate a partire dal 18 novembre 2022, l'utilizzo di archiviazione temporanea viene segnalato tramite l'endpoint dei metadati dell'attività. Le applicazioni coinvolte nelle attività possono interrogare la versione 4 dell'endpoint dei metadati delle attività per ottenere la dimensione riservata dello spazio di archiviazione temporanea e la quantità utilizzata.

Inoltre, la dimensione riservata dello storage temporaneo e la quantità utilizzata vengono inviate a Amazon CloudWatch Container Insights se attivi Container Insights.

Note

Fargate riserva spazio su disco destinato unicamente a questo motore di calcolo. Non ti viene addebitato alcun costo. Sebbene non sia mostrato in queste metriche, puoi visualizzare questo spazio di archiviazione aggiuntivo in altri strumenti, come `df`.

Versione 1.3.0 o precedente

Per le attività Amazon ECS su Fargate che utilizzano la piattaforma versione 1.3.0 o precedente, ogni attività riceve il seguente storage temporaneo.

- 10 GB di storage di livello Docker

Note

Questo importo include gli artefatti dell'immagine sia del container compresso che di quello non compresso.

- 4 GB aggiuntivi per i montaggi dei volumi. Può essere montato e condiviso tra container che utilizzano i parametri `volumes`, `mountPoints` e `volumesFrom` nella definizione di attività.

Versioni della piattaforma container Windows Fargate

Versione 1.0.0 o successiva

Per impostazione predefinita, le attività Amazon ECS ospitate su Fargate che utilizzano la piattaforma versione 1.0.0 o successiva ricevono almeno 20 GiB di storage temporaneo. La quantità totale di storage temporaneo può essere aumentata, fino a un massimo di 200 GiB. Puoi eseguire questa operazione specificando il parametro `ephemeralStorage` nella definizione di attività.

L'immagine del container estratta, compressa e non compressa per il processo viene memorizzata nell'archiviazione temporanea. Per determinare la quantità totale di storage temporaneo che l'attività deve utilizzare, è necessario sottrarre la quantità di storage utilizzato dall'immagine del container dalla quantità totale di storage temporaneo in cui è allocata l'attività.

Per ulteriori informazioni, consulta [Usa bind mount con Amazon ECS](#).

Chiavi gestite dal cliente per lo AWS Fargate storage temporaneo per Amazon ECS

AWS Fargate supporta le chiavi gestite dai clienti per crittografare i dati per le attività di Amazon ECS archiviate in uno storage temporaneo per aiutare i clienti sensibili alle normative a soddisfare le proprie politiche di sicurezza interne. I clienti ottengono comunque i vantaggi della tecnologia serverless di Fargate, offrendo al contempo una maggiore visibilità sulla crittografia dello storage autogestita ai revisori di conformità. Sebbene Fargate disponga per impostazione predefinita della crittografia dello storage effimero gestita da Fargate, i clienti possono anche utilizzare le proprie chiavi autogestite per crittografare dati sensibili come informazioni finanziarie o sanitarie.

È possibile importare o creare chiavi personalizzate in AWS KMS. Queste chiavi autogestite vengono archiviate in AWS KMS ed eseguono azioni standard AWS KMS del ciclo di vita

come la rotazione, la disabilitazione e l'eliminazione. È possibile controllare l'accesso e l'utilizzo delle chiavi nei log. CloudTrail

Per impostazione predefinita, la chiave KMS supporta 50.000 concessioni per chiave. Fargate utilizza una singola AWS KMS sovvenzione per attività chiave gestita dal cliente, quindi supporta fino a 50.000 attività simultanee per ogni chiave. Se desideri aumentare questo numero, puoi richiedere un aumento del limite, che viene approvato su base volontaria. case-by-case

Fargate non addebita alcun costo aggiuntivo per l'utilizzo delle chiavi gestite dai clienti. Ti viene addebitato solo il prezzo standard per l'utilizzo AWS KMS delle chiavi per l'archiviazione e le richieste API.

Argomenti

- [Crea una chiave di crittografia per lo storage temporaneo Fargate per Amazon ECS](#)
- [Gestione delle AWS KMS chiavi per lo storage temporaneo Fargate per Amazon ECS](#)

Crea una chiave di crittografia per lo storage temporaneo Fargate per Amazon ECS

Crea una chiave gestita dal cliente per crittografare i dati archiviati nello storage temporaneo Fargate.

Note

La crittografia dello storage temporaneo Fargate con chiavi gestite dal cliente non è disponibile per i cluster di attività di Windows.

La crittografia dello storage temporaneo Fargate con chiavi gestite dal cliente non era disponibile prima di. `platformVersions 1.4.0`

Fargate riserva spazio su uno spazio di archiviazione temporaneo utilizzato solo da Fargate e non ti viene addebitato alcun costo per lo spazio. L'allocazione potrebbe differire dalle attività chiave non gestite dal cliente, ma lo spazio totale rimane lo stesso. Puoi visualizzare questa modifica in strumenti come. `df`

Le chiavi multiregione non sono supportate per l'archiviazione temporanea di Fargate.

Gli alias delle chiavi KMS non sono supportati per l'archiviazione temporanea di Fargate.

Per creare una chiave gestita dal cliente (CMK) per crittografare lo storage temporaneo per AWS KMS Fargate in, procedi nel seguente modo.

1. <https://console.aws.amazon.com/Passa> a /kms.

2. Segui le istruzioni per la [creazione di chiavi nella Guida](#) per gli [AWS Key Management Service sviluppatori](#).
3. Durante la creazione della AWS KMS chiave, assicuratevi di fornire al servizio Fargate le autorizzazioni AWS KMS operative pertinenti nelle politiche chiave. Le seguenti operazioni API devono essere consentite nella policy per utilizzare la chiave gestita dal cliente con le risorse del cluster Amazon ECS.
 - `kms:GenerateDataKeyWithoutPlainText`- Chiama `GenerateDataKeyWithoutPlainText` per generare una chiave dati crittografata dalla AWS KMS chiave fornita.
 - `kms:CreateGrant`- Aggiunge una concessione a una chiave gestita dal cliente. Concede l'accesso di controllo a una AWS KMS chiave specificata, che consente l'accesso alle operazioni di concessione richieste da Amazon ECS Fargate. [Per ulteriori informazioni sull'utilizzo di Grants, consulta la Guida per gli sviluppatori.AWS Key Management Service](#) Ciò consente ad Amazon ECS Fargate di effettuare le seguenti operazioni:
 - Chiama `Decrypt` per AWS KMS ottenere la chiave di crittografia per decrittografare i dati di archiviazione temporanei.
 - Configura un `RetireGrant` per consentire al servizio di farlo.
 - `kms:DescribeKey`- Fornisce i dettagli chiave gestiti dal cliente per consentire ad Amazon ECS di convalidare la chiave se è simmetrica e abilitata.

L'esempio seguente mostra una politica AWS KMS chiave da applicare alla chiave di destinazione per la crittografia. Per utilizzare le dichiarazioni politiche di esempio, *user input placeholders* sostituiscile con le tue informazioni. Come sempre, configura solo le autorizzazioni di cui hai bisogno, ma dovrai fornire AWS KMS le autorizzazioni ad almeno un utente per evitare errori.

```
{
  "Sid": "Allow generate data key access for Fargate tasks.",
  "Effect": "Allow",
  "Principal": { "Service": "fargate.amazonaws.com" },
  "Action": [
    "kms:GenerateDataKeyWithoutPlaintext"
  ],
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:aws:ecs:clusterAccount": [
```

```

        "customerAccountId"
    ],
    "kms:EncryptionContext:aws:ecs:clusterName": [
        "clusterName"
    ]
  },
  },
  "Resource": "*"
},
{
  "Sid": "Allow grant creation permission for Fargate tasks.",
  "Effect": "Allow",
  "Principal": { "Service": "fargate.amazonaws.com" },
  "Action": [
    "kms:CreateGrant"
  ],
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:aws:ecs:clusterAccount": [
        "customerAccountId"
      ],
      "kms:EncryptionContext:aws:ecs:clusterName": [
        "clusterName"
      ]
    },
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Decrypt"
      ]
    }
  },
  },
  "Resource": "*"
},
{
  "Sid": "Allow describe key permission for cluster operator - CreateCluster
and UpdateCluster.",
  "Effect": "Allow",
  "Principal": { "AWS": "arn:aws:iam::customerAccountId:role/customer-chosen-
role" },
  "Action": [
    "kms:DescribeKey"
  ],
  "Resource": "*"
}

```

```
}
```

Le attività Fargate utilizzano le `aws:ecs:clusterAccount` chiavi contestuali di `aws:ecs:clusterName` crittografia per le operazioni crittografiche con la chiave. I clienti devono aggiungere queste autorizzazioni per limitare l'accesso a un account e/o cluster specifico. Usa il nome del cluster e non l'ARN quando specifichi il cluster.

Per ulteriori informazioni, consultare [Contesto della crittografia](#) nella [Guida per gli sviluppatori di AWS KMS](#).

Quando si crea o si aggiorna un cluster, è possibile utilizzare la chiave `fargateEphemeralStorageKmsKeyId` di condizione. Questa chiave di condizione consente ai clienti di avere un controllo più granulare delle policy IAM. Gli aggiornamenti alla `fargateEphemeralStorageKmsKeyId` configurazione hanno effetto solo sulle nuove implementazioni di servizi.

Di seguito è riportato un esempio di come consentire ai clienti di concedere le autorizzazioni solo a un set specifico di chiavi approvate. AWS KMS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:UpdateCluster"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecs:fargate-ephemeral-storage-kms-key": "arn:aws:kms:us-
west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE1111"
        }
      }
    }
  ]
}
```

Segue un esempio per negare i tentativi di rimuovere AWS KMS chiavi già associate a un cluster.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "ecs:CreateCluster",
      "ecs:UpdateCluster"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "ecs:fargate-ephemeral-storage-kms-key": "true"
      }
    }
  }
}
```

I clienti possono verificare se le attività non gestite o le attività di servizio sono crittografate utilizzando la chiave utilizzando i AWS CLI `describe-tasks` `describe-cluster`, `describe-services`.

Per ulteriori informazioni, consulta [Condition keys AWS KMS](#) nella [AWS KMS Developer Guide](#).

AWS Management Console

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Scegli Cluster nella barra di navigazione a sinistra e Crea cluster in alto a destra oppure scegli un cluster esistente. Per un cluster esistente, scegli Aggiorna cluster in alto a destra.
3. Nella sezione Crittografia del flusso di lavoro, avrai la possibilità di selezionare la tua AWS KMS chiave in Archiviazione gestita e Archiviazione effimera Fargate. Puoi anche scegliere di creare una AWS KMS chiave da qui.
4. Scegli Crea una volta terminata la creazione del nuovo cluster o Aggiorna, se ne stavi aggiornando uno esistente.

AWS CLI

Di seguito è riportato un esempio di creazione di un cluster e configurazione dello storage temporaneo Fargate utilizzando (sostituisci AWS CLI *red* i valori con i tuoi):

```
aws ecs create-cluster --cluster clusterName \
--configuration '{"managedStorageConfiguration":
{"fargateEphemeralStorageKmsKeyId":"arn:aws:kms:us-
west-2:012345678901:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}}'
{
  "cluster": {
    "clusterArn": "arn:aws:ecs:us-west-2:012345678901:cluster/clusterName",
    "clusterName": "clusterName",
    "configuration": {
      "managedStorageConfiguration": {
        "fargateEphemeralStorageKmsKeyId": "arn:aws:kms:us-
west-2:012345678901:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      }
    },
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": []
  },
  "clusterCount": 5
}
```

AWS CloudFormation

Di seguito è riportato un modello di esempio per la creazione di un cluster e la configurazione dello storage temporaneo Fargate utilizzando (sostituisci AWS CloudFormation *red* i valori con i tuoi):

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  MyCluster:
    Type: AWS::ECS::Cluster
    Properties:
      ClusterName: "clusterName"
      Configuration:
        ManagedStorageConfiguration:
```

```
FargateEphemeralStorageKmsKeyId: "arn:aws:kms:us-  
west-2:012345678901:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
```

Gestione delle AWS KMS chiavi per lo storage temporaneo Fargate per Amazon ECS

Dopo aver creato o importato la AWS KMS chiave per crittografare l'archivio temporaneo Fargate, la gestisci nello stesso modo in cui faresti con qualsiasi altra chiave. AWS KMS

AWS KMS Rotazione automatica dei tasti

È possibile abilitare la rotazione automatica dei tasti o ruotarli manualmente. La rotazione automatica della chiave ruota automaticamente la chiave ogni anno generando nuovo materiale crittografico per la chiave. AWS KMS salva anche tutte le versioni precedenti del materiale crittografico, così sarete in grado di decrittografare tutti i dati che utilizzavano le versioni precedenti della chiave. Qualsiasi materiale ruotato non verrà eliminato AWS KMS finché non elimini la chiave.

La rotazione automatica dei tasti è opzionale e può essere attivata o disattivata in qualsiasi momento.

Disattivazione o AWS KMS revoca delle chiavi

Se disabiliti un accesso con chiave gestita dal cliente AWS KMS, ciò non ha alcun impatto sull'esecuzione delle attività, che continuano a funzionare per tutto il loro ciclo di vita. Se una nuova attività utilizza la chiave disabilitata o revocata, l'operazione ha esito negativo poiché non può accedere alla chiave. È necessario impostare un CloudWatch allarme o qualcosa di simile per assicurarsi che non sia mai necessaria una chiave disabilitata per decrittografare i dati già crittografati.

Eliminazione delle chiavi AWS KMS

L'eliminazione delle chiavi dovrebbe sempre essere l'ultima risorsa e dovrebbe essere eseguita solo se si è certi che la chiave eliminata non sia mai più necessaria. Le nuove attività che tentano di utilizzare la chiave eliminata falliranno perché non possono accedervi. AWS KMS consiglia di disabilitare una chiave anziché eliminarla. Se ritieni necessario eliminare una chiave, ti suggeriamo di disabilitarla prima e di impostare un CloudWatch allarme per assicurarti che non sia necessaria. Se elimini una chiave, hai AWS KMS a disposizione almeno sette giorni per cambiare idea.

Controllo dell'accesso alle AWS KMS chiavi

Puoi utilizzare CloudTrail i log per controllare l'accesso alla tua AWS KMS chiave. Puoi controllare le AWS KMS operazioni `CreateGrant` `GenerateDataKeyWithoutPlaintext` e `Decrypt`. Queste

operazioni mostrano anche il `aws:ecs:clusterAccount` e `aws:ecs:clusterName` come parte dell'`EncryptionContext` accesso. CloudTrail

Di seguito sono riportati CloudTrail gli eventi di esempio per

`GenerateDataKeyWithoutPlaintext` `GenerateDataKeyWithoutPlaintext`

(`DryRun`), `CreateGrant`, `CreateGrant (DryRun)`, e `RetireGrant` (sostituisci i *red* valori con i tuoi).

`GenerateDataKeyWithoutPlaintext`

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "ec2-frontend-api.amazonaws.com"
  },
  "eventTime": "2024-04-23T18:08:13Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "ec2-frontend-api.amazonaws.com",
  "userAgent": "ec2-frontend-api.amazonaws.com",
  "requestParameters": {
    "numberOfBytes": 64,
    "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "encryptionContext": {
      "aws:ecs:clusterAccount": "account-id",
      "aws:ebs:id": "vol-xxxxxxx",
      "aws:ecs:clusterName": "cluster-name"
    }
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "readOnly": true,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  ]
}
```

```

    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "account-id",
    "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
    "eventCategory": "Management"
}

```

GenerateDataKeyWithoutPlaintext (DryRun)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "fargate.amazonaws.com"
  },
  "eventTime": "2024-04-23T18:08:11Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKeyWithoutPlaintext",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "fargate.amazonaws.com",
  "userAgent": "fargate.amazonaws.com",
  "errorCode": "DryRunOperationException",
  "errorMessage": "The request would have succeeded, but the DryRun option is set.",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "dryRun": true,
    "numberOfBytes": 64,
    "encryptionContext": {
      "aws:ecs:clusterAccount": "account-id",
      "aws:ecs:clusterName": "cluster-name"
    }
  },
  "responseElements": null,
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "readOnly": true,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",

```

```

        "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "account-id",
    "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
    "eventCategory": "Management"
  }
}

```

CreateGrant

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "ec2-frontend-api.amazonaws.com"
  },
  "eventTime": "2024-04-23T18:08:13Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "ec2-frontend-api.amazonaws.com",
  "userAgent": "ec2-frontend-api.amazonaws.com",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "granteePrincipal": "fargate.us-west-2.amazonaws.com",
    "operations": [
      "Decrypt"
    ],
    "constraints": {
      "encryptionContextSubset": {
        "aws:ecs:clusterAccount": "account-id",
        "aws:ebs:id": "vol-xxxx",
        "aws:ecs:clusterName": "cluster-name"
      }
    },
    "retiringPrincipal": "ec2.us-west-2.amazonaws.com"
  },
  "responseElements": {

```

```

    "grantId":
      "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855",
    "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
  },
  "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
  "readOnly": false,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "account-id",
  "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
  "eventCategory": "Management"
}

```

CreateGrant (DryRun)

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "fargate.amazonaws.com"
  },
  "eventTime": "2024-04-23T18:08:11Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "fargate.amazonaws.com",
  "userAgent": "fargate.amazonaws.com",
  "errorCode": "DryRunOperationException",
  "errorMessage": "The request would have succeeded, but the DryRun option is
set.",
  "requestParameters": {
    "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111",

```

```

    "granteePrincipal": "fargate.us-west-2.amazonaws.com",
    "dryRun": true,
    "operations": [
      "Decrypt"
    ],
    "constraints": {
      "encryptionContextSubset": {
        "aws:ecs:clusterAccount": "account-id",
        "aws:ecs:clusterName": "cluster-name"
      }
    },
    "responseElements": null,
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "readOnly": false,
    "resources": [
      {
        "accountId": "AWS Internal",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "account-id",
    "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa",
    "eventCategory": "Management"
  }

```

RetireGrant

```

{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AWSService",
    "invokedBy": "AWS Internal"
  },
  "eventTime": "2024-04-20T18:37:38Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "RetireGrant",
  "awsRegion": "us-west-2",

```

```

    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
      "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    },
    "additionalEventData": {
      "grantId":
        "e3b0c44298fc1c149afb4c8996fb92427ae41e4649b934ca495991b7852b855"
    },
    "requestID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE33333",
    "readOnly": false,
    "resources": [
      {
        "accountId": "AWS Internal",
        "type": "AWS::KMS::Key",
        "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "account-id",
    "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEEaaaaa",
    "eventCategory": "Management"
  }
}

```

Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS

AWS è responsabile della manutenzione dell'infrastruttura sottostante per AWS Fargate. AWS determina quando una revisione della versione della piattaforma deve essere sostituita con una nuova revisione dell'infrastruttura. Questa operazione è nota come ritiro delle attività. AWS invia una notifica di ritiro dell'attività quando viene ritirata una revisione della versione della piattaforma. Aggiorniamo regolarmente le versioni della nostra piattaforma supportate per introdurre una nuova revisione contenente aggiornamenti al software runtime Fargate e alle dipendenze sottostanti come il sistema operativo e il runtime del contenitore. Dopo aver reso disponibile una revisione più recente, ritiriamo la versione precedente per garantire che tutti i carichi di lavoro dei clienti vengano eseguiti

sulla revisione più aggiornata della versione della piattaforma Fargate. Quando una revisione viene ritirata, tutte le attività in esecuzione su di essa vengono interrotte.

Le attività di Amazon ECS possono essere classificate come attività di servizio o come attività autonome. Le attività di servizio vengono distribuite come parte di un servizio e controllate dalla pianificazione di Amazon ECS. Per ulteriori informazioni, consulta [Servizi Amazon ECS](#). Le attività autonome sono attività avviate dall'RunTaskAPI Amazon ECS, direttamente o da uno strumento di pianificazione esterno, come le attività pianificate (che vengono avviate da Amazon EventBridge) oppure. AWS Batch AWS Step Functions Non è necessario intraprendere alcuna azione in risposta al ritiro delle attività di servizio perché lo scheduler di Amazon ECS sostituisce automaticamente le attività.

Per le attività autonome, potrebbe essere necessario eseguire una gestione aggiuntiva in risposta al ritiro delle attività. Per ulteriori informazioni, consulta [Amazon ECS può gestire automaticamente le attività autonome?](#).

Per le attività di servizio, non è necessario intraprendere alcuna azione per il ritiro delle attività, a meno che non si desideri sostituire tali attività prima dell'esecuzione. AWS Quando lo scheduler di Amazon ECS interrompe le attività, utilizza `maximumPercent` e avvia una nuova attività nel tentativo di mantenere il conteggio desiderato per il servizio. Per ridurre al minimo l'impatto del ritiro delle attività di AWS Fargate, è necessario seguire le best practice di Amazon ECS durante la distribuzione dei carichi di lavoro. Il `maximumPercent` valore predefinito per un servizio che utilizza l'utilità di pianificazione del servizio REPLICA è del 200%. Pertanto, quando AWS Fargate inizia a ritirare le attività, Amazon ECS pianifica prima una nuova attività e quindi attende che sia in esecuzione, prima di ritirare una vecchia attività. Quando imposti il `maximumPercent` valore al 100%, Amazon ECS interrompe prima l'attività, quindi la sostituisce.

Per il ritiro di un'attività autonoma, AWS interrompe l'attività alla o dopo la data di ritiro dell'attività. Amazon ECS non avvia un'attività sostitutiva quando un'attività viene interrotta. Se hai bisogno che queste attività continuino a essere eseguite, devi interrompere le attività in esecuzione e avviare un'attività sostitutiva prima del tempo indicato nella notifica. Pertanto, consigliamo ai clienti di monitorare lo stato delle attività autonome e, se necessario, di implementare una logica per sostituire quelle interrotte.

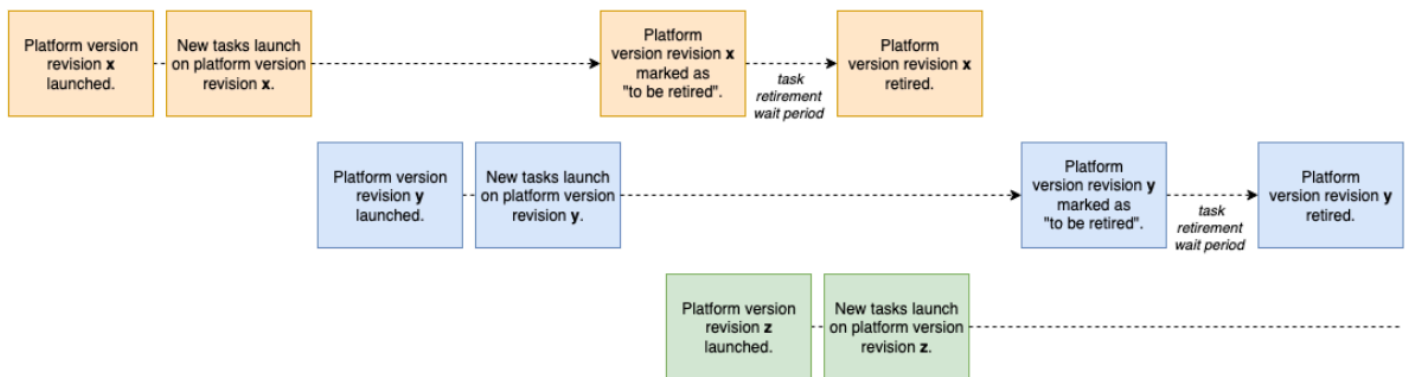
Quando un'attività viene interrotta in uno degli scenari seguenti, puoi eseguire il comando `describe-tasks`. Il `stoppedReason` nella risposta è `ECS is performing maintenance on the underlying infrastructure hosting the task`.

La manutenzione delle attività si applica quando è presente una nuova versione della piattaforma, la revisione deve essere sostituita con una nuova revisione. In caso di problemi con un host Fargate sottostante, Amazon ECS sostituisce l'host senza un avviso di ritiro dell'attività.

Panoramica dell'avviso di ritiro delle attività

Quando AWS contrassegna una revisione della versione della piattaforma come da ritirare, identifichiamo tutte le attività in esecuzione su quella revisione della versione della piattaforma in tutte le regioni. Inviando quindi una notifica per account per regione, evidenziando le attività o i servizi interessati e indicando la data in cui inizieranno i pensionamenti.

La figura seguente mostra il ciclo di vita di una revisione della versione della piattaforma Fargate, dal lancio di una nuova revisione al ritiro della versione della piattaforma.



Le seguenti informazioni forniscono dettagli.

- Dopo il lancio di una nuova revisione della versione della piattaforma, tutte le nuove attività vengono pianificate in base a questa revisione.
- Le attività esistenti che sono state pianificate e in esecuzione rimangono nella revisione in cui erano state originariamente inserite per tutta la durata dell'attività e non vengono migrate alla nuova revisione.
- Le nuove attività, ad esempio come parte di un aggiornamento di un servizio o del ritiro delle attività di Fargate, vengono inserite nell'ultima versione della piattaforma disponibile al momento del lancio.

Le notifiche di ritiro delle attività vengono inviate tramite AWS Health Dashboard e tramite e-mail all'indirizzo e-mail registrato e includono le seguenti informazioni:

- Data di ritiro dell'attività: l'attività viene interrotta a partire da questa data.

- Per le attività autonome, IDs le attività.
- Per le attività di servizio, l'ID del cluster in cui viene eseguito il servizio e il IDs nome del servizio.
- I passaggi successivi da eseguire.

In genere, inviamo una notifica ciascuna per le attività di servizio e le attività autonome. Regione AWS Tuttavia, in alcuni casi potresti ricevere più di un evento per ogni tipo di attività, ad esempio quando ci sono troppe attività per essere ritirate e superano i limiti previsti dai nostri meccanismi di notifica.

Puoi identificare le attività pianificate per il ritiro nei modi seguenti:

- Il AWS Health Dashboard

AWS Health le notifiche possono essere inviate tramite Amazon EventBridge a sistemi di archiviazione come Amazon Simple Storage Service, eseguire azioni automatiche come eseguire una AWS Lambda funzione o altri sistemi di notifica come Amazon Simple Notification Service. Per ulteriori informazioni, consulta [Monitoraggio AWS Health degli eventi con Amazon EventBridge](#). Per una configurazione di esempio per inviare notifiche ad Amazon Chime, Slack o Microsoft Teams, consulta l'archivio [AWS Health Aware](#) su. GitHub

Di seguito è riportato un esempio di evento. EventBridge

```
{
  "version": "0",
  "id": "3c268027-f43c-0171-7425-1d799EXAMPLE",
  "detail-type": "AWS Health Event",
  "source": "aws.health",
  "account": "123456789012",
  "time": "2023-08-16T23:18:51Z",
  "region": "us-east-1",
  "resources": [
    "cluster|service",
    "cluster|service"
  ],
  "detail": {
    "eventArn": "arn:aws:health:us-east-1::event/ECS/
AWS_ECS_TASK_PATCHING_RETIREMENT/AWS_ECS_TASK_PATCHING_RETIREMENT_test1",
    "service": "ECS",
    "eventScopeCode": "ACCOUNT_SPECIFIC",
```

```

    "communicationId":
      "7988399e2e6fb0b905ddc88e0e2de1fd17e4c9fa60349577446d95a18EXAMPLE",
    "lastUpdatedTime": "Wed, 16 Aug 2023 23:18:52 GMT",
    "eventRegion": "us-east-1",
    "eventTypeCode": "AWS_ECS_TASK_PATCHING_RETIREMENT",
    "eventTypeCategory": "scheduledChange",
    "startTime": "Wed, 16 Aug 2023 23:18:51 GMT",
    "endTime": "Fri, 18 Aug 2023 23:18:51 GMT",
    "eventDescription": [
      {
        "language": "en_US",
        "latestDescription": "\\n\\nA software update has been deployed to
Fargate which includes CVE patches or other critical patches. No action is required
on your part. All new tasks launched automatically uses the latest software
version. For existing tasks, your tasks need to be restarted in order for these
updates to apply. Your tasks running as part of the following ECS Services will
be automatically updated beginning Wed, 16 Aug 2023 23:18:51 GMT.\\n\\n\\nAfter Wed,
16 Aug 2023 23:18:51 GMT, the ECS scheduler will gradually replace these tasks,
respecting the deployment settings for your service. Typically, services should
see little to no interruption during the update and no action is required. When AWS
stops tasks, AWS uses the minimum healthy percent (1) and launches a new task in
an attempt to maintain the desired count for the service. By default, the minimum
healthy percent of a service is 100 percent, so a new task is started first before
a task is stopped. Service tasks are routinely replaced in the same way when
you scale the service or deploy configuration changes or deploy task definition
revisions. If you would like to control the timing of this restart you can update
the service before Wed, 16 Aug 2023 23:18:51 GMT, by running the update-service
command from the ECS command-line interface specifying force-new-deployment for
services using Rolling update deployment type. For example:\\n\\n\\n$ aws ecs update-
service -service service_name \\n\\n--cluster cluster_name -force-new-deployment\\
\\n\\nFor services using Blue/Green deployment type with AWS CodeDeploy:\\n\\nPlease
refer to create-deployment document (2) and create new deployment using same task
definition revision.\\n\\n\\nFor further details on ECS deployment types, please
refer to ECS Deployment Developer Guide (1).\\n\\nFor further details on Fargate's
update process, please refer to the AWS Fargate User Guide (3).\\n\\nIf you have
any questions or concerns, please contact AWS Support (4).\\n\\n\\n(1) https://
docs.aws.amazon.com/AmazonECS/latest/developerguide/deployment-types.html\\n(2)
https://docs.aws.amazon.com/cli/latest/reference/deploy/create-deployment.html\\n(3)
https://docs.aws.amazon.com/AmazonECS/latest/userguide/task-maintenance.html\\n(4)
https://aws.amazon.com/support\\n\\n\\nA list of your affected resources(s) can be
found in the 'Affected resources' tab in the 'Cluster/ Service' format in the AWS
Health Dashboard. \\n\\n\\n"
      }
    ],

```

```
    "affectedEntities": [
      {
        "entityValue": "cluster|service"
      },
      {
        "entityValue": "cluster|service"
      }
    ]
  }
}
```

- E-mail

Viene inviata un'e-mail all'indirizzo di posta elettronica registrato per l' Account AWS ID.

Per informazioni su come prepararsi al ritiro delle attività, vedere [Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS](#).

Posso rinunciare al ritiro delle attività?

No. Come parte del modello di responsabilità AWS condivisa, AWS è responsabile della gestione e della manutenzione dell'infrastruttura sottostante per AWS Fargate. Ciò include l'esecuzione di aggiornamenti periodici della piattaforma per garantire sicurezza e stabilità. Questi aggiornamenti vengono applicati automaticamente dai clienti AWS e non possono essere disattivati. Questo è uno dei vantaggi principali dell'utilizzo di AWS Fargate rispetto all'esecuzione dei carichi di lavoro su EC2 istanze, la responsabilità della manutenzione della piattaforma sottostante è gestita da AWS. Questo modello consente di concentrarsi sulle applicazioni anziché sulla manutenzione dell'infrastruttura. Applicando automaticamente questi aggiornamenti della piattaforma, AWS è in grado di mantenere l'ambiente Fargate sicuro up-to-date e protetto, senza che sia richiesta alcuna azione da parte dell'utente in qualità di cliente. Questo aiuta a fornire un ambiente containerizzato affidabile e sicuro per l'esecuzione dei carichi di lavoro su Fargate.

Posso ricevere notifiche di ritiro delle attività tramite altri servizi? AWS

AWS invia una notifica di ritiro dell'attività al AWS Health Dashboard e al contatto di posta elettronica principale su Account AWS. AWS Health Dashboard Fornisce una serie di integrazioni in altri AWS servizi, tra cui EventBridge. È possibile utilizzare EventBridge per automatizzare la visibilità degli avvisi (ad esempio inoltrando il messaggio a uno strumento). ChatOps Per ulteriori informazioni, consulta [Panoramica della soluzione](#): acquisizione delle notifiche di ritiro delle attività.

Posso modificare il ritiro di un'attività dopo averla pianificata?

No. La pianificazione si basa sul tempo di attesa per il ritiro dell'attività, che ha un valore predefinito di 7 giorni. Se hai bisogno di più tempo, puoi scegliere di configurare il periodo di attesa su 14 giorni. Per ulteriori informazioni, consulta [Fase 2: acquisisci le notifiche di cessazione delle attività per avvisare i team e intraprendere azioni](#). La modifica di questa configurazione si applica ai pensionamenti che verranno programmati in futuro. I pensionamenti attualmente programmati non sono influenzati. Se hai ulteriori dubbi, contatta. Supporto

In che modo Amazon ECS gestisce le attività che fanno parte di un servizio?

Per le attività di servizio, non è necessario intraprendere alcuna azione in risposta al ritiro dell'attività, a meno che non si desideri sostituire queste attività prima dell' AWS esecuzione. Quando lo scheduler di Amazon ECS interrompe le attività, utilizza la percentuale minima di integrità e avvia una nuova attività nel tentativo di mantenere il numero desiderato per il servizio. Per ridurre al minimo l'impatto del ritiro delle attività di Fargate, i carichi di lavoro devono essere distribuiti seguendo le best practice di Amazon ECS. Ad esempio, quando distribuiscono un'applicazione stateless come servizio Amazon ECS, come un server Web o API, i clienti devono distribuire più repliche di attività e impostarle al 100%. `minimumHealthyPercent` Per impostazione predefinita, la percentuale minima di integrità di un servizio è pari al 100%. Pertanto, quando Fargate inizia a ritirare le attività, Amazon ECS pianifica innanzitutto una nuova attività e attende che sia in esecuzione, prima di ritirare una vecchia attività. Le attività di servizio vengono regolarmente sostituite come parte del ritiro delle attività allo stesso modo quando si ridimensiona il servizio, si implementano modifiche alla configurazione o si distribuiscono revisioni delle definizioni delle attività. Per prepararsi al processo di ritiro delle attività, vedere. [Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS](#)

Amazon ECS può gestire automaticamente le attività autonome?

No. AWS non può creare un'attività sostitutiva per attività autonome avviate da `RunTask`, attività pianificate (ad esempio tramite `EventBridge Scheduler`) o. `AWS Batch` `AWS Step Functions` Amazon ECS gestisce solo le attività che fanno parte di un servizio.

Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS

Per prepararti al ritiro delle attività, esegui le seguenti operazioni:

1. Imposta il periodo di attesa per il ritiro dell'attività.

2. Registra le notifiche di ritiro delle attività per avvisare i membri del team.
3. Non puoi controllare l'ora esatta del ritiro di un'attività, tuttavia puoi controllare la sostituzione delle attività aggiornando il servizio con l'opzione di distribuzione forzata.

Passaggio 1: imposta il tempo di attesa dell'attività

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (0). Quando hai bisogno di un maggiore controllo, ad esempio, quando un'attività può essere interrotta solo durante una determinata finestra, configura l'opzione 7 (7) o 14 giorni (14).

Consigliamo di scegliere un periodo di attesa più breve per ricevere prima le revisioni delle versioni più recenti della piattaforma.

Configura il periodo di attesa eseguendo `put-account-setting-default` o `put-account-setting` come utente root o utente amministrativo. Utilizza l'opzione `fargateTaskRetirementWaitPeriod` per il name e l'opzione `value` impostata su uno dei seguenti valori:

- 0- AWS invia la notifica e inizia immediatamente a ritirare le attività interessate.
- 7- AWS invia la notifica e attende 7 giorni di calendario prima di iniziare a ritirare le attività interessate.
- 14- AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

L'impostazione di default è 7 giorni.

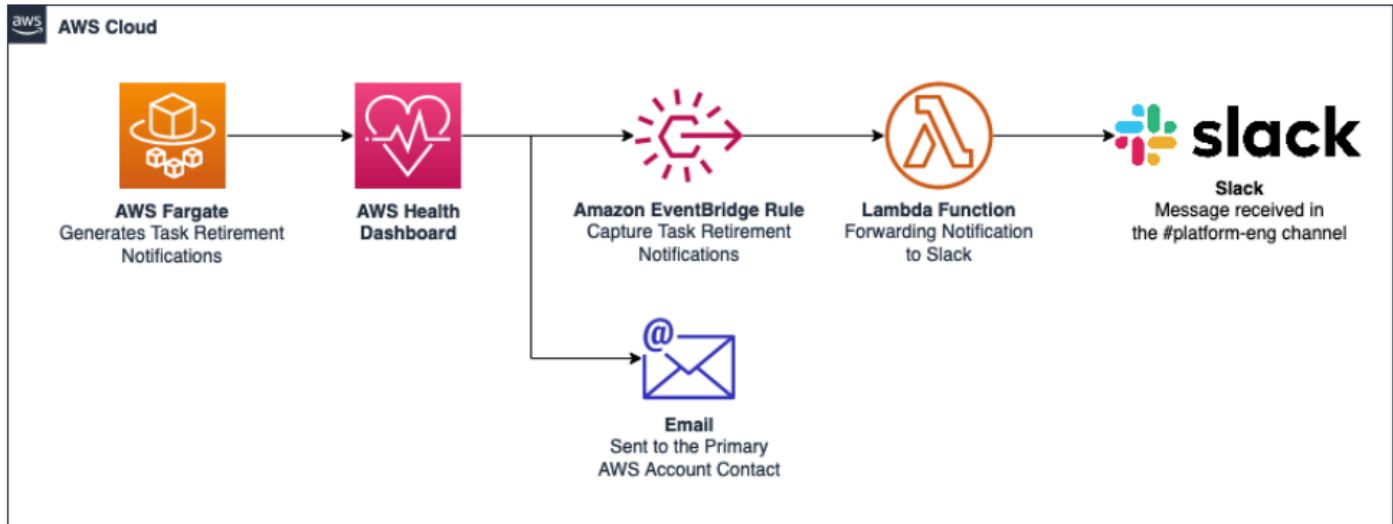
Per ulteriori informazioni, consulta [put-account-setting-default](#) e consulta il riferimento [put-account-setting](#) all'API di Amazon Elastic Container Service.

Fase 2: acquisisci le notifiche di cessazione delle attività per avvisare i team e intraprendere azioni

Quando è imminente il ritiro di un'attività, AWS invia una notifica di ritiro dell'attività alla AWS Health Dashboard e al contatto e-mail principale su Account AWS. La AWS Health dashboard offre una serie di integrazioni con altri AWS servizi, tra cui Amazon EventBridge. Puoi utilizzarla EventBridge per creare automazioni a partire da una notifica di ritiro di un'attività, ad esempio per aumentare la visibilità del pensionamento imminente inoltrando il messaggio a uno strumento.

ChatOps AWS Health Aware è una risorsa che mostra la potenza della AWS Health Dashboard e come le notifiche possono essere distribuite all'interno di un'organizzazione. Puoi inoltrare una notifica di ritiro dell'attività a un'applicazione di chat, come Slack.

La figura seguente mostra la panoramica della soluzione.



Le seguenti informazioni forniscono dettagli.

- Fargate invia la notifica di ritiro dell'attività alla Dashboard. AWS Health
- La AWS Health Dashboard invia posta elettronica al contatto e-mail principale sul e Account AWS notifica. EventBridge
- EventBridge ha una regola che registra la notifica di pensionamento.

La regola che cerca gli eventi con l'Event Detail Type: "AWS Health Event" and the Event Detail Type Code: "AWS_ECS_TASK_PATCHING_RETIREMENT"

- La regola attiva una funzione Lambda che inoltra le informazioni a Slack utilizzando uno Slack Incoming Webhook. [Per ulteriori informazioni, consulta Incoming Webhooks.](#)

Per un esempio di codice, vedi [Acquisizione delle notifiche di ritiro delle AWS Fargate attività](#) su Github.

Passaggio 3: controlla la sostituzione delle attività

Non è possibile controllare l'ora esatta del ritiro di un'attività, tuttavia è possibile definire un tempo di attesa. Se desideri controllare la sostituzione delle attività in base alla tua pianificazione, puoi

registrare l'avviso di ritiro dell'attività per comprendere innanzitutto la data di ritiro dell'attività. È quindi possibile ridistribuire il servizio per avviare attività sostitutive e, allo stesso modo, sostituire qualsiasi attività autonoma. Per i servizi che utilizzano la distribuzione progressiva, è possibile aggiornare il servizio utilizzando `update-service` l'opzione prima dell'inizio del pensionamento. `force-deployment`

L'esempio seguente utilizza l'opzione. `update-service force-deployment`

```
aws ecs update-service --service service_name \  
  --cluster cluster_name \  
  --force-new-deployment
```

Per i servizi che utilizzano la distribuzione blu/verde, è necessario creare una nuova distribuzione in. AWS CodeDeployPer informazioni su come creare la distribuzione, consulta [create-deployment](#) nella Guida di riferimento.AWS Command Line Interface

Regioni supportate per Amazon ECS su AWS Fargate

È possibile utilizzare le seguenti tabelle per verificare il supporto regionale per i contenitori Linux su AWS Fargate e i contenitori Windows su AWS Fargate.

Contenitori Linux su AWS Fargate

I contenitori Amazon ECS Linux su AWS Fargate sono supportati nei seguenti Regioni AWS casi. Le zone di disponibilità supportate IDs vengono indicate quando applicabile.

Nome della regione	Regione
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
Stati Uniti occidentali (California settentrionale)	us-west-1 (solo <code>usw1-az1</code> e <code>usw1-az3</code>)
US West (Oregon)	us-west-2
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1

Nome della regione	Regione
Messico (centrale)	mx-central-1
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacifico (Tokyo)	ap-northeast-1 (solo apne1-az1 , apne1-az2 e apne1-az4)
Asia Pacifico (Seul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Tailandia)	ap-southeast-7
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacifico (Malesia)	ap-southeast-5
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (solo cnn1-az1 e cnn1-az2)
Cina (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2

Nome della regione	Regione
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2
Europa (Stoccolma)	eu-north-1
South America (São Paulo)	sa-east-1
Israele (Tel Aviv)	il-central-1
Medio Oriente (Bahrein)	me-south-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1

Contenitori Windows su AWS Fargate

I contenitori Amazon ECS Windows attivi AWS Fargate sono supportati nei seguenti Regioni AWS casi. Le zone di disponibilità supportate IDs vengono indicate quando applicabile.

Nome della regione	Regione
US East (Ohio)	us-east-2
Stati Uniti orientali (Virginia settentrionale)	us-east-1 (solo use1-az1, use1-az2, use1-az4, use1-az5 e use1-az6)
Stati Uniti occidentali (California settentrionale)	us-west-1 (solo usw1-az1 e usw1-az3)

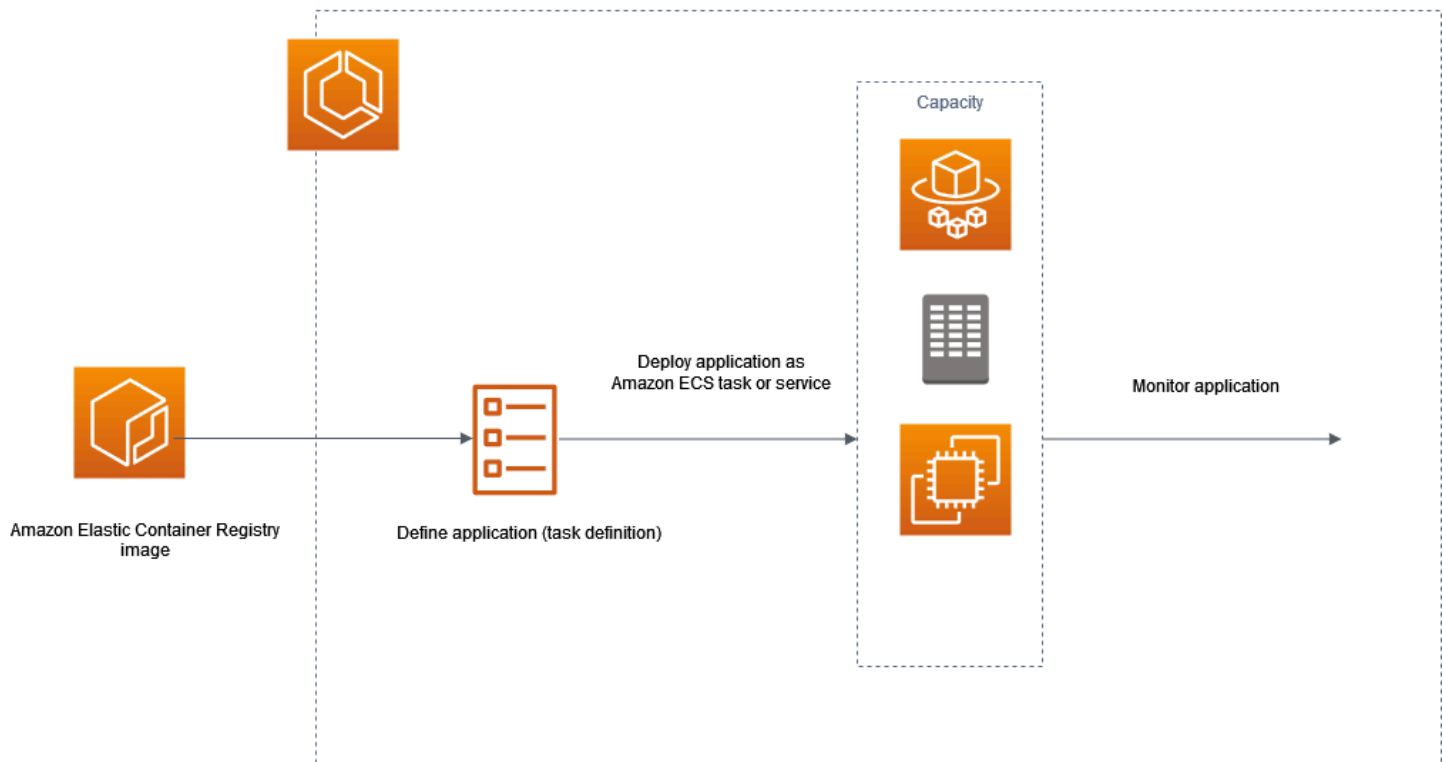
Nome della regione	Regione
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacifico (Malesia)	ap-southeast-5
Asia Pacifico (Tokyo)	ap-northeast-1 (solo apne1-az1 , apne1-az2 e apne1-az4)
Canada (Centrale)	ca-central-1 (solo cac1-az1 e cac1-az2)
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (solo cnn1-az1 e cnn1-az2)
Cina (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2

Nome della regione	Regione
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2
Europa (Stoccolma)	eu-north-1
South America (São Paulo)	sa-east-1
Israele (Tel Aviv)	il-central-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Medio Oriente (Bahrein)	me-south-1

Progetta la tua soluzione per Amazon ECS

Il diagramma seguente mostra il ciclo di vita dell'applicazione e il funzionamento con i componenti di Amazon ECS.

Amazon ECS Application Lifecycle



È necessario progettare le applicazioni in modo che possano essere eseguite su contenitori. Un container è un'unità di sviluppo software standardizzata che contiene tutto ciò che è necessario per l'esecuzione dell'applicazione software. Ciò include codice, runtime, strumenti di sistema e librerie di sistema pertinenti. I container vengono creati da un modello di sola lettura denominato immagine. Le immagini sono generalmente create da un Dockerfile. Un Dockerfile è un file di testo semplice che contiene le istruzioni per la creazione di un contenitore. In seguito alla creazione, tali immagini vengono archiviate in un registro, ad esempio Amazon ECR, da cui possono essere scaricate.

Dopo aver creato e archiviato l'immagine, puoi creare una definizione dell'attività di Amazon ECS. Una definizione di attività è uno schema dell'applicazione. Si tratta di un file di testo in formato JSON che descrive i parametri e uno o più container che formano l'applicazione. Ad esempio, puoi utilizzarla per specificare l'immagine e i parametri del sistema operativo, i container da utilizzare, le

porte da aprire per l'applicazione e i volumi di dati da utilizzare con i container nell'attività. I parametri specifici disponibili per la definizione di attività dipendono dalle esigenze dell'applicazione specifica.

Dopo aver stabilito la definizione di attività, implementala come servizio o attività nel cluster. Un cluster è un raggruppamento logico di attività o servizi in esecuzione sull'infrastruttura di capacità registrata in un cluster.

Si definisce attività la creazione dell'istanza relativa a una definizione di attività all'interno di un cluster. Puoi eseguire un processo autonomo oppure eseguire un processo come parte di un servizio. Puoi utilizzare un servizio Amazon ECS per eseguire e mantenere simultaneamente il numero desiderato di processi in un cluster Amazon ECS. Il funzionamento è che, se uno dei processi non riesce o si interrompe per qualsiasi motivo, il pianificatore del servizio Amazon ECS lancia un'altra istanza in base alla definizione di attività. Lo fa per sostituirlo e quindi mantenere il numero desiderato di processi nel servizio.

L'agente del container funziona su ogni istanza di container in un cluster Amazon ECS. L'agente invia ad Amazon ECS informazioni sui processi correntemente in esecuzione e sull'utilizzo delle risorse dei tuoi container. Avvia e interrompe i processi ogni volta che riceve una richiesta da Amazon ECS.

Dopo aver implementato l'attività o il servizio, puoi utilizzare uno degli strumenti seguenti per monitorare l'implementazione e l'applicazione:

- CloudWatch
- Monitoraggio del runtime

Capacità

La capacità è l'infrastruttura in cui vengono eseguiti i contenitori. Le opzioni sono le seguenti:

- EC2 Istanze Amazon
- Senza server (AWS Fargate)
- Macchine virtuali (VM) o server on-premise

L'infrastruttura viene specificata quando si crea un cluster. È inoltre possibile specificare il tipo di infrastruttura quando si registra una definizione di attività. La definizione dell'attività si riferisce all'infrastruttura come al «tipo di avvio». Il tipo di avvio viene utilizzato anche quando si esegue un'attività autonoma o si distribuisce un servizio. Per informazioni sulle opzioni del tipo di avvio, consulta [Tipi di avvio di Amazon ECS](#)

Endpoint di servizio

L'endpoint del servizio è l'URL del punto di ingresso per Amazon ECS che usi per connetterti al servizio in modo programmatico utilizzando Internet Protocol versione 4 (IPv4) o Internet Protocol versione 6 (IPv6). Per impostazione predefinita, le richieste effettuate per connettersi ad Amazon ECS utilizzano programmaticamente endpoint di servizio che supportano solo le richieste IPv4. Per connetterti a livello di codice con Amazon ECS utilizzando una delle due IPv4 IPv6 richieste, puoi utilizzare un endpoint dual-stack. Per informazioni sull'utilizzo di un endpoint dual-stack, consulta.

[Utilizzo degli endpoint dual-stack Amazon ECS](#)

Rete

AWS le risorse vengono create in sottoreti. Quando usi EC2 le istanze, Amazon ECS avvia le istanze nella sottorete specificata al momento della creazione di un cluster. Le tue attività vengono eseguite nella sottorete dell'istanza. Per le macchine virtuali Fargate o locali, si specifica la sottorete quando si esegue un'attività o si crea un servizio.

A seconda dell'applicazione, la sottorete può essere una sottorete privata o pubblica e può trovarsi in una delle seguenti risorse: AWS

- Zone di disponibilità
- Zone locali
- Zone Wavelength
- Regioni AWS
- AWS Outposts

Per ulteriori informazioni, consulta [Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones](#) o [Amazon Elastic Container Service su AWS Outposts](#).

È possibile connettere l'applicazione a Internet utilizzando uno dei seguenti metodi:

- Una sottorete pubblica con un gateway Internet

Utilizza le sottoreti pubbliche quando hai applicazioni pubbliche che richiedono grandi quantità di larghezza di banda o una latenza minima. Gli scenari applicabili includono lo streaming video e i servizi di gioco.

- Una sottorete privata con un gateway NAT

Utilizza sottoreti private quando desideri proteggere i contenitori dall'accesso esterno diretto. Gli scenari applicabili includono sistemi di elaborazione dei pagamenti o contenitori che archiviano dati e password degli utenti.

- AWS PrivateLink

AWS PrivateLink Utilizzatelo per avere una connettività privata tra VPCs AWS i servizi e le reti locali senza esporre il traffico alla rete Internet pubblica.

Accesso alle funzionalità

Puoi utilizzare le impostazioni del tuo account Amazon ECS per accedere alle seguenti funzionalità:

- Container Insights

CloudWatch Container Insights raccoglie, aggrega e riepiloga i parametri e i log delle applicazioni e dei microservizi containerizzati. I parametri includono l'utilizzo di risorse come CPU, memoria, dischi e rete.

- awsvpctrunking

Per alcuni tipi di EC2 istanze, è possibile avere interfacce di rete aggiuntive (ENIs) disponibili sulle istanze di container appena lanciate.

- Autorizzazione all'assegnazione di tag

Gli utenti devono disporre delle autorizzazioni per le azioni che creano una risorsa, ad esempio. `ecsCreateCluster` Se i tag sono specificati nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva sull'azione per verificare se gli utenti o i ruoli dispongono delle `ecs:TagResource` autorizzazioni per creare tag.

- Conformità FIPS-140 di Fargate

Fargate supporta il Federal Information Processing Standard (FIPS-140) che specifica i requisiti di sicurezza previsti per i moduli crittografici che proteggono le informazioni sensibili. È l'attuale standard governativo degli Stati Uniti e del Canada ed è applicabile ai sistemi che devono essere conformi al Federal Information Security Management Act (FISMA) o al Federal Risk and Authorization Management Program (FedRAMP).

- Modifiche all'orario di pensionamento delle attività di Fargate

È possibile configurare il periodo di attesa prima che le attività di Fargate vengano ritirate per l'applicazione delle patch.

- VPC a doppio stack

Consenti alle attività di comunicare tra di IPv4 loro o con IPv6 entrambe.

- Formato Amazon Resource Name (ARN)

Alcune funzionalità, come l'autorizzazione all'etichettatura, richiedono un nuovo formato Amazon Resource Name (ARN).

Per ulteriori informazioni, consulta [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#).

Ruoli IAM

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

Alcune funzionalità di Amazon ECS richiedono ruoli. Per ulteriori informazioni, consulta [Ruoli IAM per Amazon ECS](#).

Registrazione

La registrazione e il monitoraggio sono aspetti importanti per mantenere l'affidabilità, la disponibilità e le prestazioni dei carichi di lavoro Amazon ECS. Sono disponibili le seguenti opzioni:

- Amazon CloudWatch log: indirizza i log verso Amazon CloudWatch
- FireLens per Amazon ECS: indirizza i log verso un AWS servizio o una AWS Partner Network destinazione per l'archiviazione e l'analisi dei log. AWS Partner Network È una comunità globale di partner che sfrutta programmi, competenze e risorse per creare, commercializzare e vendere offerte ai clienti.

Tipi di avvio di Amazon ECS

Il tipo di avvio della definizione dell'attività definisce, ad esempio, la capacità con cui l'attività può essere eseguita AWS Fargate.

Dopo aver scelto il tipo di avvio, Amazon ECS verifica che i parametri di definizione delle attività configurati funzionino con il tipo di avvio.

La tabella seguente descrive i tipi di avvio disponibili.

Tipo di avvio		Ulteriori informazioni
Fargate	Fargate è un motore di pay-as-you-go elaborazione senza server che ti consente di concentrarti sulla creazione di applicazioni senza gestire i server.	Tipo di lancio Fargate per Amazon ECS
EC2	Scegli il tipo di istanza e il numero di istanze per gestire la capacità.	EC2 tipo di avvio per Amazon ECS
Macchine virtuali (VM) o server on-premise	Amazon ECS Anywhere fornisce supporto per la registrazione di un'istanza esterna, ad esempio un server locale o una macchina virtuale (VM), nel tuo cluster.	Tipo di lancio esterno (Amazon ECS Anywhere) per Amazon ECS

Tipo di lancio Fargate per Amazon ECS

Fargate è un motore di pay-as-you-go elaborazione senza server che ti consente di concentrarti sulla creazione di applicazioni senza gestire i server. Scegliendo Fargate, non è necessario gestire un'EC2 infrastruttura. Tutto ciò che devi fare è creare l'immagine del contenitore e definire su quale cluster desideri eseguire le tue applicazioni. Fargate ha un'integrazione nativa con AWS servizi tra cui:

- Amazon VPC
- Auto Scaling
- Sistema di bilanciamento del carico elastico
- IAM
- Secrets Manager

Con Fargate avete più controllo che EC2 selezionando esattamente la CPU e la memoria di cui la vostra applicazione ha bisogno. Fargate gestisce la scalabilità della tua capacità, quindi non devi preoccuparti dei picchi di traffico. Ciò significa che l'impegno operativo con Fargate è minore.

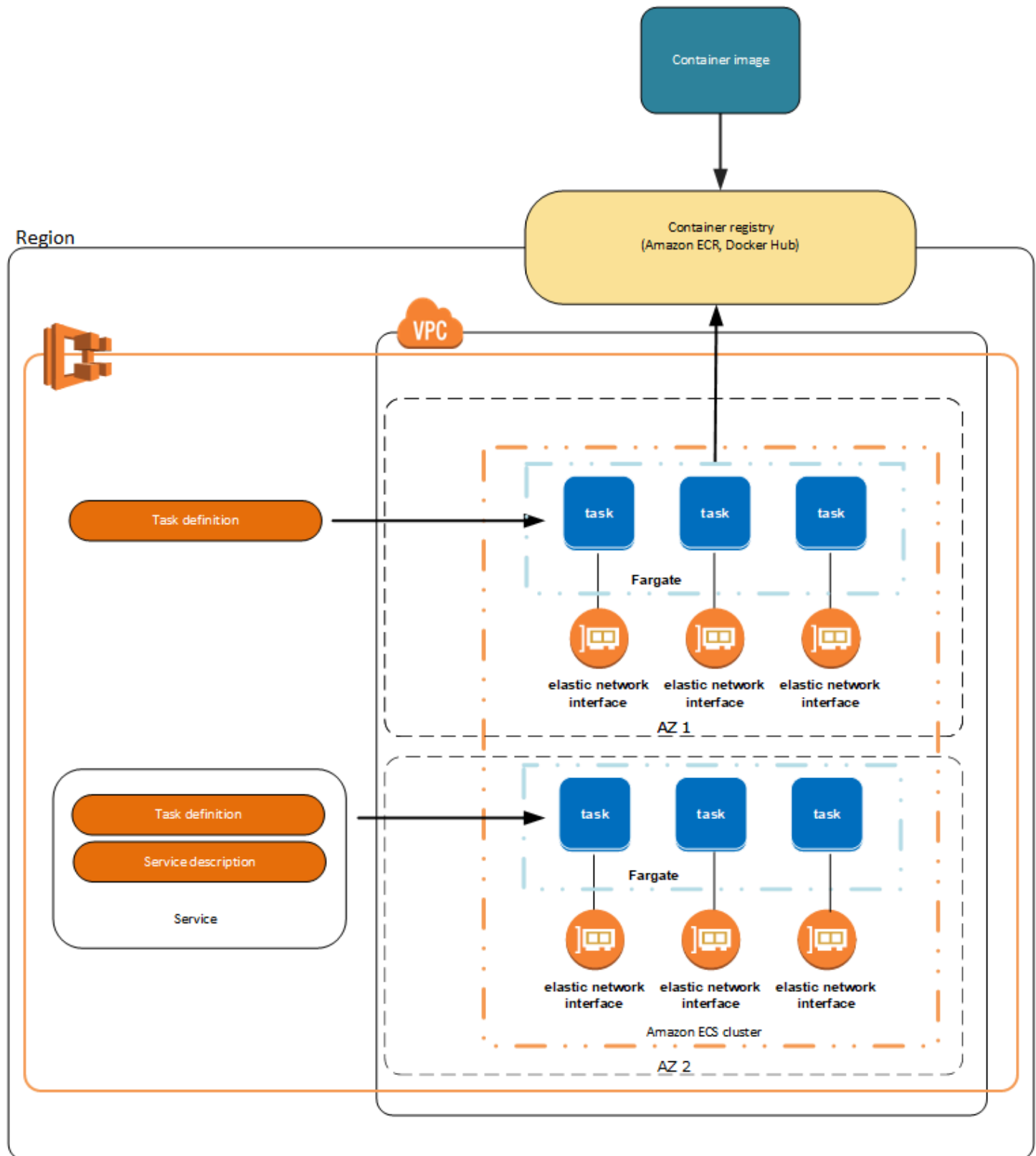
Fargate soddisfa gli standard per i programmi di conformità tra cui PCI, FIPS 140-3, FedRAMP e HIPAA. [Per ulteriori informazioni, vedere Services in Scope by Compliance Program.AWS](#)

Fargate è adatto per i seguenti carichi di lavoro:

- Carichi di lavoro di grandi dimensioni che richiedono un sovraccarico operativo ridotto
- Piccoli carichi di lavoro con sequenza occasionale
- Carichi di lavoro ridotti
- Carichi di lavoro in batch

Per informazioni sulle regioni che supportano Fargate, consulta [the section called “AWS Regioni di Fargate”](#).

Il diagramma seguente illustra l'architettura generale.



Per ulteriori informazioni su Amazon ECS con Fargate, consulta [AWS Fargate per Amazon ECS](#).

Comportamento dell'immagine dei contenitori Linux su Fargate Container Pull per Amazon ECS

Ogni attività Fargate viene eseguita su una propria istanza monouso con tenant singolo. Quando si eseguono contenitori Linux su Fargate, le immagini dei contenitori o i livelli di immagini dei contenitori non vengono memorizzati nella cache dell'istanza. Pertanto, per ogni immagine del contenitore definita nell'operazione, l'intera immagine del contenitore deve essere estratta dal registro delle immagini del contenitore per ogni attività Fargate. Il tempo necessario per estrarre le immagini è direttamente correlato al tempo impiegato per avviare un'attività Fargate.

Tenete conto di quanto segue per ottimizzare il tempo di caricamento dell'immagine.

Prossimità dell'immagine del contenitore

Per ridurre il tempo necessario per scaricare le immagini dei container, posiziona i dati il più vicino possibile al calcolo. L'estrazione dell'immagine di un contenitore da Internet o dall'altra parte Regioni AWS potrebbe influire sui tempi di download. Ti consigliamo di archiviare l'immagine del contenitore nella stessa regione in cui verrà eseguita l'attività. Se memorizzi l'immagine del contenitore in Amazon ECR, utilizza un endpoint di interfaccia VPC per ridurre ulteriormente il tempo di recupero dell'immagine. Per ulteriori informazioni, consulta gli [endpoint VPC dell'interfaccia Amazon ECR AWS PrivateLink\(\)](#) nella Amazon ECR User Guide.

Riduzione delle dimensioni dell'immagine del contenitore

Le dimensioni di un'immagine del contenitore influiscono direttamente sul tempo di download. La riduzione delle dimensioni dell'immagine del contenitore o del numero di livelli dell'immagine del contenitore può ridurre il tempo necessario per il download di un'immagine. Le immagini di base leggere (come l'immagine minima del contenitore Amazon Linux 2023) possono essere significativamente più piccole di quelle basate sulle immagini di base del sistema operativo tradizionale. Per ulteriori informazioni sull'immagine minima, consulta [AL2023 Minimal container image](#) nella Amazon Linux 2023 User Guide.

Algoritmi di compressione alternativi

I livelli di immagini dei contenitori vengono spesso compressi quando vengono inseriti in un registro di immagini dei contenitori. La compressione del livello di immagine del contenitore riduce la quantità di dati che devono essere trasferiti attraverso la rete e archiviati nel registro delle immagini del contenitore. Dopo che un livello di immagine del contenitore è stato scaricato su un'istanza dal runtime del contenitore, tale livello viene decompresso. L'algoritmo di compressione utilizzato e la quantità di v CPUs disponibile per il runtime influiscono sul tempo necessario per

decomprimere l'immagine del contenitore. Su Fargate, puoi aumentare le dimensioni dell'attività o sfruttare l'algoritmo di compressione zstd più performante per ridurre il tempo impiegato per la decompressione. [Per ulteriori informazioni, vedere zstd on. GitHub](#) Per informazioni su come implementare le immagini per Fargate, vedere [Riduzione dei tempi di AWS Fargate avvio con le immagini dei container compressi zstd.](#)

Lazy Loading delle immagini dei contenitori

Per immagini di contenitori di grandi dimensioni (> 250 MB), potrebbe essere ottimale caricare lentamente un'immagine del contenitore anziché scaricarla tutta. Su Fargate, puoi utilizzare Seekable OCI (SOCI) per caricare lentamente un'immagine del contenitore da un registro delle immagini del contenitore. Per ulteriori informazioni, consulta [soci-snapshotter](#) on GitHub e [Lazy loading](#) container images using Seekable OCI (SOCI).

Comportamento dell'immagine dei contenitori Windows su Fargate Container Pull per Amazon ECS

Fargate Windows memorizza nella cache l'immagine di base servercore del mese più recente e quella del mese precedente fornita da Microsoft. Queste immagini corrispondono alle patch con numero KB/Build aggiornate ogni Patch Tuesday. Ad esempio, il 9 aprile 2024, Microsoft ha rilasciato KB5 036896 (17763.5696) per Windows Server 2019. Il numero di KB del mese precedente, il 12 marzo 2024, era 035849 (17763.5576). KB5 Quindi, per le piattaforme WINDOWS_SERVER_2019_CORE e i seguenti contenitori, le immagini sono state memorizzate WINDOWS_SERVER_2019_FULL nella cache:

- `mcr.microsoft.com/windows/servercore:ltsc2019`
- `mcr.microsoft.com/windows/servercore:10.0.17763.5696`
- `mcr.microsoft.com/windows/servercore:10.0.17763.5576`

Inoltre, il 9 aprile 2024, Microsoft ha rilasciato KB5 036909 (20348.2402) per Windows Server 2022. Il mese precedente KB del 12 marzo 2024 era 035857 (20348.2340). KB5 Quindi, per le piattaforme WINDOWS_SERVER_2022_CORE e i seguenti contenitori, le immagini sono state memorizzate nella cache: WINDOWS_SERVER_2022_FULL

- `mcr.microsoft.com/windows/servercore:ltsc2022`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2402`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2340`

EC2 tipo di avvio per Amazon ECS

Il tipo di EC2 lancio è adatto per carichi di lavoro di grandi dimensioni che devono essere ottimizzati in termini di prezzo.

Quando si considera come modellare le definizioni delle attività e i servizi utilizzando il tipo di EC2 avvio, si consiglia di considerare quali processi devono essere eseguiti insieme e come è possibile ridimensionare ciascun componente.

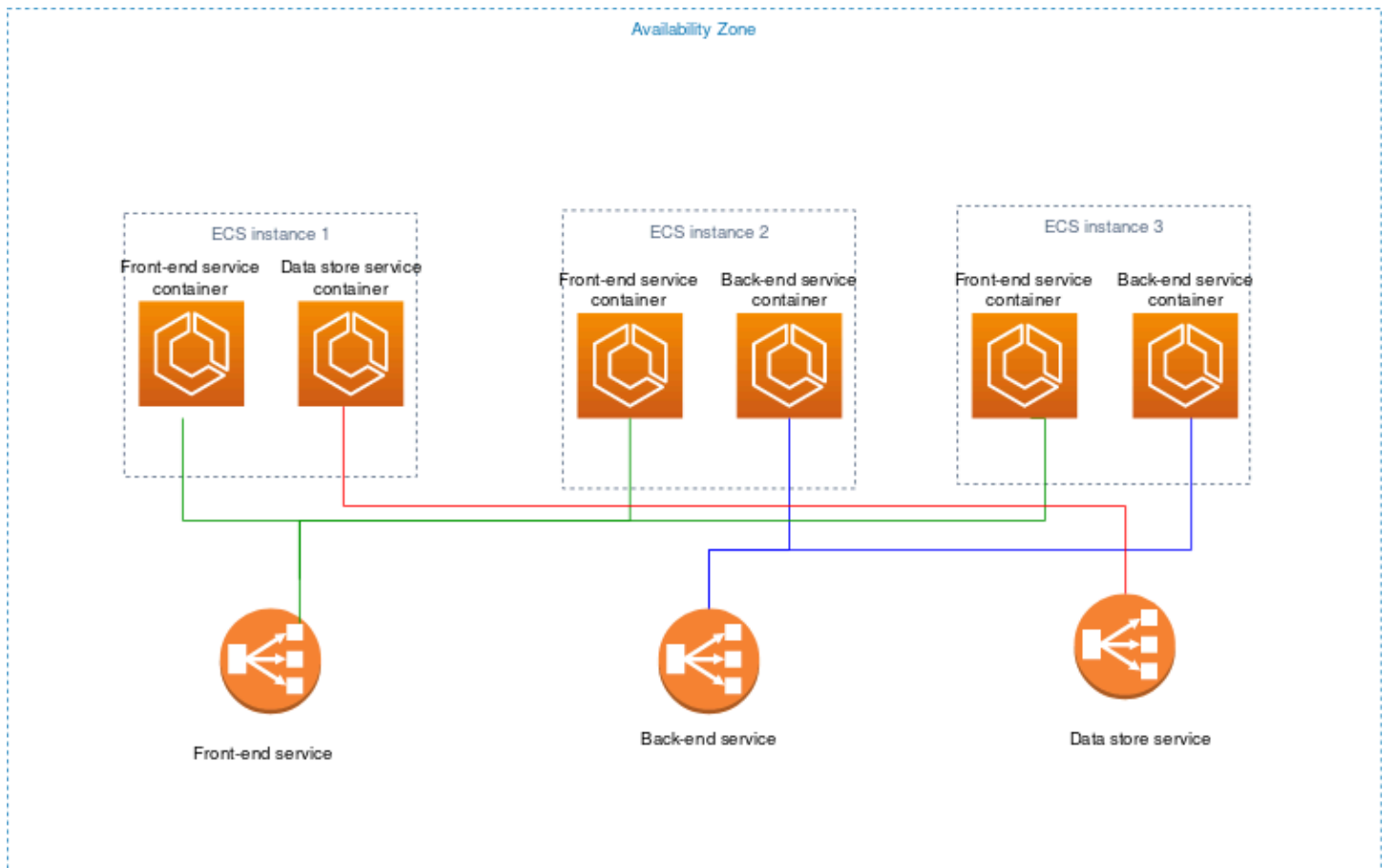
Ad esempio, immagina un'applicazione che comprende i seguenti componenti:

- Un servizio front-end che visualizza le informazioni su una pagina Web
- Un servizio di backend che fornisce il servizio di APIs frontend
- Un datastore

Per questo esempio, crea definizioni di attività che raggruppino i container utilizzati per scopi comuni. Dividi i diversi componenti in definizioni di attività multiple e separate. Il seguente esempio di cluster presenta tre istanze di container su cui vengono eseguiti tre container per i servizi front-end, due container per i servizi back-end e un container per i servizi di datastore.

Puoi raggruppare i container correlati in una definizione di attività, ad esempio i container collegati che devono essere eseguiti contemporaneamente. Ad esempio, aggiungi un container per lo streaming dei log al servizio front-end, includendolo nella stessa definizione di attività.

Una volta configurate, dalle definizioni di attività puoi creare dei servizi per mantenere la disponibilità dei tuoi processi desiderati. Per ulteriori informazioni, consulta [Creazione di un servizio Amazon ECS utilizzando la console](#). Nei servizi, puoi associare i container mediante i load balancer di Elastic Load Balancing. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#). Quando le tue esigenze relative alle applicazioni cambiano, puoi aggiornare i tuoi servizi per incrementare o ridurre il numero di attività desiderate. In alternativa, puoi aggiornare i servizi per implementare versioni più recenti dei container nelle attività. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).



Comportamento dell'immagine del contenitore per EC2 i tipi di avvio esterni per Amazon ECS

Il tempo impiegato da un contenitore per l'avvio varia in base all'immagine del contenitore sottostante. Ad esempio, un'immagine più grande (versioni complete di Debian, Ubuntu e Amazon1/2) potrebbe richiedere più tempo per l'avvio perché ci sono più servizi che vengono eseguiti nei contenitori rispetto alle rispettive versioni slim (Debian-slim, Ubuntu-slim e Amazon-slim) o immagini di base più piccole (Alpine).

Quando l'agente Amazon ECS avvia un'attività, estrae l'immagine Docker dal registro remoto e quindi memorizza nella cache una copia locale. Quando utilizzi un nuovo tag di immagine per ogni versione dell'applicazione, questo comportamento non è necessario.

Il parametro `ECS_IMAGE_PULL_BEHAVIOR` agent determina il comportamento di estrazione dell'immagine. Sono disponibili le seguenti opzioni:

- `ECS_IMAGE_PULL_BEHAVIOR: default`

L'immagine verrà estratta da remoto. Se l'estrazione fallisce, viene utilizzata l'immagine memorizzata nella cache dell'istanza.

- `ECS_IMAGE_PULL_BEHAVIOR: always`

L'immagine verrà estratta da remoto. Se l'estrazione fallisce, l'operazione fallisce.

Per velocizzare la distribuzione, imposta il parametro dell'agente Amazon ECS su uno dei seguenti valori:

- `ECS_IMAGE_PULL_BEHAVIOR: once`

L'immagine viene recuperata in remoto solo se non è stata recuperata da un'attività precedente sulla stessa istanza del contenitore o se l'immagine memorizzata nella cache è stata rimossa dal processo automatico di pulizia delle immagini. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. In questo modo si ha la garanzia che non vengano eseguite estrazioni di immagini non necessarie.

- `ECS_IMAGE_PULL_BEHAVIOR: prefer-cached`

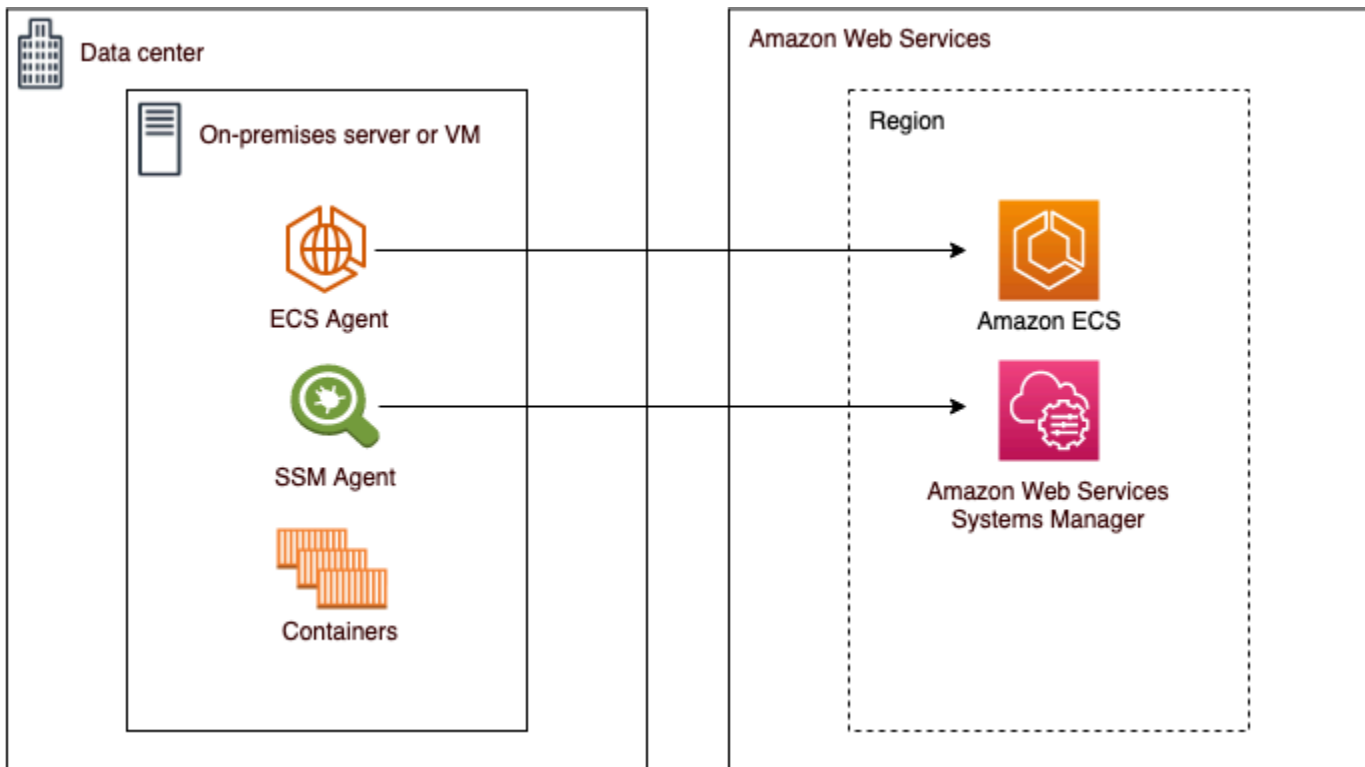
L'immagine viene recuperata in remoto se non è presente alcuna immagine memorizzata nella cache. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. La pulizia automatica delle immagini è disattivata per il contenitore per garantire che l'immagine memorizzata nella cache non venga rimossa.

L'impostazione del `ECS_IMAGE_PULL_BEHAVIOR` parametro su uno dei valori precedenti può far risparmiare tempo perché l'agente Amazon ECS utilizza l'immagine scaricata esistente. Per immagini Docker più grandi, il trasferimento dalla rete potrebbe richiedere 10-20 secondi per il download.

Tipo di lancio esterno (Amazon ECS Anywhere) per Amazon ECS

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premise o una macchina virtuale (VM) nel cluster Amazon ECS. Le istanze esterne sono ottimizzate per l'esecuzione di applicazioni che generano traffico in uscita o dati di processo. Se l'applicazione richiede traffico in entrata, la mancanza di supporto Elastic Load Balancing rende l'esecuzione di questi carichi di lavoro meno efficiente. Amazon ECS ha aggiunto un nuovo tipo di avvio `EXTERNAL` che è possibile utilizzare per creare servizi o eseguire processi sulle istanze esterne.

Di seguito viene fornita una panoramica dell'architettura di sistema di alto livello di Amazon ECS Anywhere. Sul tuo server locale sono installati sia l'agente Amazon ECS che l'agente SSM.



Per ulteriori informazioni, consulta [Cluster Amazon ECS per il tipo di lancio esterno](#).

Utilizzo degli endpoint dual-stack Amazon ECS

Gli endpoint dual-stack di Amazon ECS supportano le richieste ad Amazon ECS tramite Internet Protocol versione 4 (IPv4) e Internet Protocol versione 6 (). IPv6 Per un elenco di tutti gli endpoint Amazon ECS, consulta [Endpoint e quote Amazon ECS](#) nel. Riferimenti generali di AWS

Quando utilizzi l'API REST, accedi direttamente a un endpoint Amazon ECS utilizzando il nome dell'endpoint (URI). Amazon ECS supporta solo nomi di endpoint dual-stack regionali, il che significa che è necessario specificare la regione come parte del nome.

Utilizza la seguente convenzione di denominazione per i nomi degli endpoint dual-stack:
`ecs.region.api.aws`

Quando si utilizza il AWS Command Line Interface (AWS CLI) and AWS SDKs, è possibile utilizzare un parametro o un flag per passare a un endpoint dual-stack. Puoi anche specificare l'endpoint dual-stack direttamente come override dell'endpoint Amazon ECS nel file di configurazione.

Le sezioni seguenti descrivono come utilizzare gli endpoint dual-stack provenienti dall'API, dalla e dall'API REST. AWS CLI AWS SDKs

Argomenti

- [Utilizzo degli endpoint dual-stack dall'AWS CLI](#)
- [Utilizzo di endpoint dual-stack provenienti da AWS SDKs](#)
- [Utilizzo degli endpoint dual-stack dall'API REST](#)

Utilizzo degli endpoint dual-stack dall'AWS CLI

Questa sezione fornisce esempi di AWS CLI comandi utilizzati per effettuare richieste a un endpoint dual-stack. Per ulteriori informazioni sull'installazione AWS CLI o sull'aggiornamento alla versione più recente, vedere [Installazione o aggiornamento all'ultima versione di AWS CLI nella Guida per l'AWS Command Line Interface utente della versione 2](#).

Per utilizzare un endpoint dual-stack, puoi impostare il valore di configurazione `true` nel `config` file per `use_dualstack_endpoint` AWS CLI indirizzare tutte le richieste Amazon ECS effettuate dal `ecs` AWS CLI comando all'endpoint dual-stack per la regione specificata. È possibile specificare la regione nel `config` file o in un comando utilizzando l'opzione. `--region` Per ulteriori informazioni sui file di configurazione per la AWS CLI, vedere [Impostazioni dei file di configurazione e credenziali AWS CLI nella Guida per l'AWS Command Line Interface utente della versione 2](#).

È inoltre possibile utilizzare un comando per aggiornare la configurazione. Il comando seguente viene aggiornato `use_dualstack_endpoint true` al profilo predefinito.

```
$ aws configure set default.ecs.use_dualstack_endpoint true
```

Se desideri utilizzare un endpoint dual-stack per AWS CLI comandi specifici, puoi utilizzare uno dei seguenti metodi:

- È possibile utilizzare l'endpoint dual-stack per comando impostando il parametro `su o` per qualsiasi comando. `--endpoint-url https://ecs.aws-region.api.aws http://ecs.aws-region.api.aws` `ecs`

Il comando di esempio seguente elenca tutti i cluster disponibili e utilizza l'endpoint dual-stack per la richiesta.

```
$ aws ecs list-clusters --endpoint-url https://ecs.aws-region.api.aws
```

- È possibile impostare profili separati nel file. AWS Config Ad esempio, si può creare un profilo che imposta `use_dualstack_endpoint` su `true` e un profilo che non imposta `use_dualstack_endpoint`. Quando si esegue un comando, specificare il profilo da usare, a seconda se si desidera o meno utilizzare l'endpoint dual-stack.

Utilizzo di endpoint dual-stack provenienti da AWS SDKs

Questa sezione fornisce esempi di come accedere a un endpoint dual-stack utilizzando. AWS SDKs

AWS SDK for Java 2.x

L'esempio seguente mostra come specificare un endpoint dual-stack per la regione utilizzando. `us-east-1` AWS SDK for Java 2.x

```
Region region = Region.US_EAST_1
EcsClient client =
    EcsClient.builder().region(region).dualstackEnabled(true).build();
```

AWS SDK per Go

L'esempio seguente mostra come specificare un endpoint dual-stack per la Regione utilizzando. `us-east-1` AWS SDK per Go

```
sess := session.Must(session.NewSession())
svc := ecs.New(sess, &aws.Config{
    Region: aws.String(endpoints.UsEast1RegionID),
    Endpoint: aws.String("https://ecs.us-east-1.api.aws")
})
```

Per ulteriori informazioni, consulta gli endpoint [Dual-stack e FIPS nella Guida di riferimento agli strumenti e](#) agli strumenti. AWS SDKs

Utilizzo degli endpoint dual-stack dall'API REST

Quando utilizzi l'API REST, puoi accedere direttamente a un endpoint dual-stack specificandolo nella richiesta. L'esempio seguente utilizza l'endpoint dual-stack per elencare tutti i cluster Amazon ECS nella regione. `us-east-1`

```
POST / HTTP/1.1
```

```
Host: ecs.us-east-1.api.aws
Accept-Encoding: identity
Content-Length: 2
X-Amz-Target: AmazonEC2ContainerServiceV20141113.ListClusters
X-Amz-Date: 20150429T170621Z
Content-Type: application/x-amz-json-1.1
Authorization: AUTHPARAMS

{}
```

Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones

Amazon ECS supporta carichi di lavoro che utilizzano Local Zones, Wavelength Zones e quando è richiesta una bassa latenza o AWS Outposts l'elaborazione locale dei dati.

- Puoi utilizzare Local Zones come estensione di una Regione AWS per collocare le risorse in più posizioni più vicine agli utenti finali.
- Le zone Wavelength possono essere utilizzate per creare applicazioni che offrono latenze molto basse a dispositivi 5G e agli utenti finali. Wavelength implementa servizi di elaborazione e archiviazione AWS standard ai margini delle reti 5G dei gestori di telecomunicazioni.
- AWS Outposts offre modelli nativi Servizi AWS, infrastrutturali e operativi praticamente a qualsiasi data center, spazio di co-ubicazione o struttura locale.

Important

Amazon ECS sui AWS Fargate carichi di lavoro non è supportato in Local Zones, Wavelength Zones o attivo al momento. AWS Outposts

Per informazioni sulle differenze tra Local Zones, Wavelength Zones AWS Outposts e, [vedi Come devo pensare a quando usare AWS Wavelength, AWS Local Zones o per applicazioni che richiedono bassa latenza o elaborazione locale](#) dei dati in. AWS Outposts AWS Wavelength FAQs

Sottoreti condivise

È possibile utilizzare la Condivisione VPC per condividere sottoreti con altri account AWS all'interno della stessa AWS Organizations.

È possibile utilizzare shared VPCs per il tipo di EC2 avvio con le seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarla per le risorse Amazon ECS.
- Non puoi utilizzare il gruppo di sicurezza predefinito VPC per le istanze del contenitore perché appartiene al proprietario. Inoltre, i partecipanti non possono avviare istanze utilizzando gruppi di sicurezza di proprietà di altri partecipanti o del proprietario.
- In una sottorete condivisa, il partecipante e il proprietario controllano separatamente i gruppi di sicurezza all'interno di ciascun account. Il proprietario della sottorete può visualizzare i gruppi di sicurezza che sono stati creati dai partecipanti, ma non può eseguire operazioni sugli stessi. Se il proprietario della sottorete desidera rimuovere o modificare questi gruppi di sicurezza, è il partecipante che ha creato il gruppo di sicurezza che deve eseguire l'operazione.
- Il proprietario del VPC condiviso non può visualizzare, aggiornare o eliminare un cluster creato da un partecipante nella sottorete condivisa. Ciò si aggiunge alle risorse VPC alle quali ogni account ha un accesso diverso. Per ulteriori informazioni, consulta la sezione [Responsabilità e autorizzazioni per proprietari e partecipanti](#) nella Guida per l'utente di Amazon VPC.

Puoi usare shared VPCs per il tipo di lancio Fargate con le seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarla per le risorse Amazon ECS.
- Non puoi creare un servizio o eseguire un'attività utilizzando il gruppo di sicurezza predefinito per il VPC perché appartiene al proprietario. Inoltre, i partecipanti non possono creare un servizio o eseguire un'attività utilizzando gruppi di sicurezza di proprietà di altri partecipanti o del proprietario.
- In una sottorete condivisa, il partecipante e il proprietario controllano separatamente i gruppi di sicurezza all'interno di ciascun account. Il proprietario della sottorete può visualizzare i gruppi di sicurezza che sono stati creati dai partecipanti, ma non può eseguire operazioni sugli stessi. Se il proprietario della sottorete desidera rimuovere o modificare questi gruppi di sicurezza, è il partecipante che ha creato il gruppo di sicurezza che deve eseguire l'operazione.
- Il proprietario del VPC condiviso non può visualizzare, aggiornare o eliminare un cluster creato da un partecipante nella sottorete condivisa. Ciò si aggiunge alle risorse VPC alle quali ogni account ha un accesso diverso. Per ulteriori informazioni, consulta la sezione [Responsabilità e autorizzazioni per proprietari e partecipanti](#) nella Guida per l'utente di Amazon VPC.

Per ulteriori informazioni sulla condivisione delle sottoreti VPC, consulta la pagina [Condivisione del VPC con altri account](#) nella Guida per l'utente di Amazon VPC.

Zone locali

Una zona locale è un'estensione di una Regione AWS zona geografica molto vicina agli utenti. Le zone locali hanno le loro connessioni a Internet e supportano AWS Direct Connect. Le risorse create in una zona locale possono servire gli utenti locali con comunicazioni a latenza molto bassa. Per ulteriori informazioni, consulta [AWS Zone locali](#).

Una zona locale è rappresentata da un codice della regione seguito da un identificatore che indica la posizione, ad esempio `us-west-2-lax-1a`.

Per utilizzare una zona locale, è necessario prima attivare la zona. Una volta eseguita l'attivazione, dovrai creare un Amazon VPC e una sottorete nella zona locale.

Puoi avviare EC2 istanze Amazon, FSx file server Amazon e Application Load Balancer da utilizzare per i cluster e le attività Amazon ECS.

Per ulteriori informazioni, consulta [What is AWS Local Zones?](#) nella AWS Local Zones User Guide.

Zone Wavelength

È possibile utilizzare AWS Wavelength per creare applicazioni che offrono latenze molto basse a dispositivi mobili e utenti finali. Wavelength implementa servizi di elaborazione e archiviazione AWS standard ai margini delle reti 5G dei gestori di telecomunicazioni. È possibile estendere un Amazon Virtual Private Cloud (VPC) a una o più zone Wavelength. Quindi, puoi utilizzare AWS risorse come le EC2 istanze Amazon per eseguire applicazioni che richiedono una latenza estremamente bassa e una connessione Servizi AWS nella regione.

Una zona Wavelength è una zona isolata nella posizione carrier in cui viene implementata l'infrastruttura Wavelength. Le Wavelength Zones sono legate a un. Regione AWS Una zona Wavelength è un'estensione logica di una regione ed è gestita dal piano di controllo nella regione.

Una zona Wavelength è rappresentata da un codice della regione seguito da un identificatore che indica la zona Wavelength, ad esempio `us-east-1-wl1-bos-wlz-1`.

Per utilizzare una zona Wavelength, devi prima attivare la zona. Una volta eseguita l'attivazione, dovrai creare un Amazon VPC e una sottorete nella zona Wavelength. Quindi, puoi avviare le EC2 istanze Amazon nella Zona da utilizzare per i cluster e le attività Amazon ECS.

Per ulteriori informazioni, consulta [Nozioni di base su AWS Wavelength](#) nella Guida per gli sviluppatori di AWS Wavelength .

Le Wavelength Zone non sono disponibili in tutte. Regioni AWS Per informazioni sulle regioni che supportano le zone Wavelength, consulta [Zone Wavelength disponibili](#) nella Guida per gli sviluppatori di AWS Wavelength .

Amazon Elastic Container Service su AWS Outposts

AWS Outposts abilita AWS servizi, infrastrutture e modelli operativi nativi nelle strutture locali. Negli AWS Outposts ambienti, è possibile utilizzare gli stessi AWS APIs strumenti e la stessa infrastruttura utilizzati in. Cloud AWS

Amazon ECS on AWS Outposts è ideale per carichi di lavoro a bassa latenza che devono essere eseguiti in prossimità di dati e applicazioni locali.

[Per ulteriori informazioni su AWS Outposts, consulta la Guida per l'utente.AWS Outposts](#)

Considerazioni

Di seguito sono riportate alcune considerazioni sull'utilizzo di Amazon ECS su: AWS Outposts

- Amazon Elastic Container Registry e Network Load Balancer vengono eseguiti nella AWS regione, non su. AWS Identity and Access Management AWS Outposts In tal modo si aumenta la latenza tra questi servizi e i container.
- AWS Fargate non è disponibile su. AWS Outposts

Di seguito sono riportate le considerazioni sulla connettività di rete per AWS Outposts:

- Se la connettività di rete tra l'utente AWS Outposts e la relativa AWS regione viene interrotta, i cluster continueranno a funzionare. Tuttavia, non potrai creare nuovi cluster o intraprendere nuove azioni nei cluster esistenti fino a quando la connettività non viene ripristinata. In caso di errori di istanza, l'istanza non verrà sostituita automaticamente. L'agente CloudWatch Logs non sarà in grado di aggiornare i registri e i dati degli eventi.
- Ti consigliamo di fornire una connettività affidabile, ad alta disponibilità e a bassa latenza tra la tua regione AWS Outposts e quella corrispondente. AWS

Prerequisiti

Di seguito sono riportati i prerequisiti per l'utilizzo di Amazon ECS su: AWS Outposts

- Devi aver installato e configurato un Outpost nel data center locale.
- É necessaria una connessione di rete affidabile tra l'Outpost e la relativa Regione AWS .

Panoramica della creazione di cluster su AWS Outposts

Di seguito è riportata una panoramica della configurazione:

1. Crea un ruolo e una politica con diritti attivi AWS Outposts.
2. Crea un profilo dell'istanza IAM con diritti per AWS Outposts.
3. Crea un VPC o usane uno esistente che si trova nella stessa regione del tuo. AWS Outposts
4. Crea una sottorete o usane una esistente associata a. AWS Outposts

Questa è la sottorete in cui vengono eseguite le istanze del contenitore.

5. Crea un gruppo di sicurezza per le istanze del contenitore nel tuo cluster.
6. Crea un cluster Amazon ECS.
7. Definisci le variabili di ambiente dell'agente container Amazon ECS per avviare l'istanza nel cluster.
8. Esegui un contenitore.

Per informazioni dettagliate su come integrare Amazon ECS con AWS Outposts, consulta [Estendere Amazon ECS su due AWS Outposts rack](#).

Nell'esempio seguente viene creato un cluster Amazon ECS in un AWS Outposts.

1. Crea un ruolo e una policy con diritti attivi. AWS Outposts

Il file `role-policy.json` è il documento di policy che contiene l'effetto e le operazioni per le risorse. Per informazioni sul formato del file, consulta [PutRolePolicy](#) la sezione IAM API Reference

```
aws iam create-role --role-name ecsRole \  
    --assume-role-policy-document file://ecs-policy.json  
aws iam put-role-policy --role-name ecsRole --policy-name ecsRolePolicy \  
    --policy-document file://ecs-policy.json
```



```
--policy-document file://role-policy.json
```

2. Crea un profilo dell'istanza IAM con diritti per AWS Outposts.

```
aws iam create-instance-profile --instance-profile-name outpost
aws iam add-role-to-instance-profile --instance-profile-name outpost \
  --role-name ecsRole
```

3. Crea un VPC.

```
aws ec2 create-vpc --cidr-block 10.0.0.0/16
```

4. Crea una sottorete associata alla tua AWS Outposts.

```
aws ec2 create-subnet \
  --cidr-block 10.0.3.0/24 \
  --vpc-id vpc-xxxxxxxx \
  --outpost-arn arn:aws:outposts:us-west-2:123456789012:outpost/op-xxxxxxxxxxxxxxxxxxxx \
  --availability-zone-id usw2-az1
```

5. Crea un gruppo di sicurezza per le istanze del container, specificando l'intervallo CIDR corretto per l' AWS Outposts. (Questo passaggio è diverso per AWS Outposts.)

```
aws ec2 create-security-group --group-name MyOutpostSG
aws ec2 authorize-security-group-ingress --group-name MyOutpostSG --protocol tcp \
  --port 22 --cidr 10.0.3.0/24
aws ec2 authorize-security-group-ingress --group-name MyOutpostSG --protocol tcp \
  --port 80 --cidr 10.0.3.0/24
```

6. Creare il cluster.
7. Definisci le variabili di ambiente dell'agente del container di Amazon ECS per avviare l'istanza nel cluster creato nella fase precedente e definisci i tag che desideri aggiungere per aiutare a identificare il cluster (ad esempio, Outpost per indicare che il cluster è per un Outpost).

```
#!/bin/bash
cat << 'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_IMAGE_PULL_BEHAVIOR=prefer-cached
ECS_CONTAINER_INSTANCE_TAGS={"environment": "Outpost"}
EOF
```

Note

Per evitare ritardi causati dal pull delle immagini del container da Amazon ECR nella regione, utilizza le cache delle immagini. A tale scopo, ogni volta che viene eseguita un'attività, configura l'agente Amazon ECS sui valori di default per utilizzare l'immagine memorizzata nella cache nell'istanza stessa impostando `ECS_IMAGE_PULL_BEHAVIOR` su `prefer-cached`.

8. Crea l'istanza di container, specificando il VPC e la sottorete per l' AWS Outposts in cui deve essere eseguita questa istanza e un tipo di istanza disponibile nell' AWS Outposts. (Questo passaggio è diverso per AWS Outposts.)

Il file `userdata.txt` contiene i dati utente che l'istanza può utilizzare per eseguire i comuni processi di configurazione automatizzati e anche per eseguire script all'avvio dell'istanza.

Per informazioni sul file per le chiamate API, consulta [Esegui comandi sulla tua istanza Linux all'avvio](#) nella Amazon EC2 User Guide.

```
aws ec2 run-instances --count 1 --image-id ami-xxxxxxx --instance-type c5.large \  
  --key-name aws-outpost-key --subnet-id subnet-xxxxxxxxxxxxxxxx \  
  --iam-instance-profile Name outpost --security-group-id sg-xxxxxx \  
  --associate-public-ip-address --user-data file:///userdata.txt
```

Note

Questo comando viene utilizzato anche quando si aggiungono istanze al cluster. Tutti i container distribuiti nel cluster vengono posizionati nell' AWS Outposts specifico.

Ottimizza la capacità e la disponibilità di Amazon ECS

La disponibilità delle applicazioni è fondamentale per fornire un'esperienza priva di errori e per ridurre al minimo la latenza delle applicazioni. La disponibilità dipende dalla disponibilità di risorse con una capacità sufficiente per soddisfare la domanda. AWS fornisce diversi meccanismi per gestire la disponibilità. Per le applicazioni ospitate su Amazon ECS, queste includono la scalabilità automatica e le zone di disponibilità (AZs). La scalabilità automatica gestisce il numero di attività o istanze in base a parametri definiti dall'utente, mentre le zone di disponibilità consentono di ospitare l'applicazione in posizioni isolate ma geograficamente vicine.

Come per quanto riguarda le dimensioni delle attività, la capacità e la disponibilità presentano alcuni compromessi da considerare. Idealmente, la capacità sarebbe perfettamente allineata alla domanda. Ci sarebbe sempre la capacità sufficiente per soddisfare le richieste ed elaborare i lavori per soddisfare gli obiettivi del livello di servizio (SLOs), tra cui una bassa latenza e un tasso di errore. La capacità non sarebbe mai troppo elevata, con conseguenti costi eccessivi, né troppo bassa, con conseguenti alti tassi di latenza e di errore.

La scalabilità automatica è un processo latente. Innanzitutto, è necessario fornire metriche in tempo reale a CloudWatch. Quindi, devono essere aggregate per l'analisi, che può richiedere fino a diversi minuti a seconda della granularità della metrica. CloudWatch confronta le metriche con le soglie di allarme per identificare una carenza o un eccesso di risorse. Per evitare l'instabilità, configura gli allarmi in modo da richiedere che la soglia impostata venga superata per alcuni minuti prima che l'allarme si spenga. Inoltre, è necessario del tempo per fornire nuove attività e terminare quelle che non sono più necessarie.

A causa di questi potenziali ritardi nel sistema descritto, è importante mantenere un certo margine di manovra mediante un approvvigionamento eccessivo. In questo modo è possibile far fronte a picchi di domanda a breve termine. Ciò consente inoltre all'applicazione di soddisfare richieste aggiuntive senza raggiungere la saturazione. È buona norma impostare l'obiettivo di scalabilità tra il 60 e l'80% dell'utilizzo. Questo aiuta l'applicazione a gestire meglio i picchi di domanda extra mentre la capacità aggiuntiva è ancora in fase di provisioning.

Un altro motivo per cui consigliamo di effettuare un eccesso di provisioning è quello di poter rispondere rapidamente ai guasti della zona di disponibilità. Consigliamo di gestire i carichi di lavoro di produzione da più zone di disponibilità. Questo perché, se si verifica un errore nella zona di disponibilità, le attività in esecuzione nelle zone di disponibilità rimanenti possono comunque soddisfare la domanda. Se l'applicazione viene eseguita in due zone di disponibilità, è necessario raddoppiare il normale numero di attività. In questo modo è possibile fornire una capacità immediata in caso di potenziale guasto. Se l'applicazione viene eseguita in tre zone di disponibilità, si consiglia di eseguire 1,5 volte il numero di attività normale. Cioè, esegui tre attività ogni due necessarie per il servizio ordinario.

Massimizzazione della velocità di scalabilità

La scalabilità automatica è un processo reattivo che richiede tempo per avere effetto. Tuttavia, esistono alcuni modi per ridurre al minimo il tempo necessario per la scalabilità orizzontale.

Ridurre al minimo le dimensioni dell'immagine. Le immagini più grandi richiedono più tempo per essere scaricate da un archivio di immagini e decomprimate. Pertanto, mantenere le dimensioni delle

immagini più piccole riduce il tempo necessario per l'avvio di un contenitore. Per ridurre le dimensioni dell'immagine, puoi seguire questi consigli specifici:

- Se puoi creare un file binario statico o usare Golang, crea la tua immagine FROM scratch e includi solo la tua applicazione binaria nell'immagine risultante.
- Usa immagini di base ridotte al minimo da fornitori di distribuzioni upstream, come Amazon Linux o Ubuntu.
- Non includete artefatti di costruzione nell'immagine finale. L'utilizzo di build in più fasi può aiutarti in questo senso.
- RUNPalchi compatti laddove possibile. Ogni RUN fase crea un nuovo livello di immagine, che comporta un ulteriore percorso di andata e ritorno per scaricare il layer. Una singola RUN fase con più comandi uniti && ha meno livelli di una con più RUN fasi.
- Se desideri includere dati, come i dati di inferenza ML, nell'immagine finale, includi solo i dati necessari per avviare e iniziare a generare traffico. Se recuperi dati su richiesta da Amazon S3 o altro storage senza influire sul servizio, archivia invece i dati in quei luoghi.

Mantieni le tue immagini vicine. Maggiore è la latenza di rete, maggiore è il tempo necessario per scaricare l'immagine. Ospita le tue immagini in un repository nella stessa regione in cui si trova il tuo carico di lavoro. Amazon ECR è un repository di immagini ad alte prestazioni disponibile in tutte le regioni in cui è disponibile Amazon ECS. Evita di utilizzare Internet o un collegamento VPN per scaricare le immagini dei contenitori. L'hosting delle immagini nella stessa regione migliora l'affidabilità complessiva. Riduce il rischio di problemi di connettività di rete e problemi di disponibilità in una regione diversa. In alternativa, puoi anche implementare la replica interregionale di Amazon ECR per aiutarti in questo.

Riduci le soglie di controllo dello stato del sistema di bilanciamento del carico. I sistemi di bilanciamento del carico eseguono controlli di integrità prima di inviare traffico all'applicazione. La configurazione predefinita del controllo dello stato di salute per un gruppo target può richiedere 90 secondi o più. Durante questo periodo, il load balancer controlla lo stato di salute e riceve le richieste. La riduzione dell'intervallo dei controlli sanitari e del numero di soglie può far sì che l'applicazione accetti il traffico più rapidamente e riduca il carico su altre attività.

Considerate le prestazioni con avvio a freddo. Alcune applicazioni utilizzano runtime come la compilazione Java perform Just-In-Time (JIT). Il processo di compilazione, almeno all'avvio, può rallentare le prestazioni dell'applicazione. Una soluzione alternativa consiste nel riscrivere le parti critiche per la latenza del carico di lavoro in linguaggi che non impongano un calo delle prestazioni a freddo.

Utilizza la scalabilità graduale, non le politiche di scalabilità basate sul tracciamento degli obiettivi. Esistono diverse opzioni di Application Auto Scaling per le attività. Il monitoraggio degli obiettivi è la modalità più semplice da utilizzare, in quanto è necessario soltanto impostare un valore target per un parametro, ad esempio l'utilizzo medio della CPU. L'autoscaler gestisce automaticamente il numero di attività necessarie per raggiungere tale valore. Il dimensionamento per fasi consente di reagire più rapidamente alle variazioni della domanda, poiché si definiscono le soglie specifiche per i parametri di dimensionamento e il numero di attività da aggiungere o rimuovere quando le soglie vengono superate. In particolare, consente di reagire molto rapidamente alle variazioni della domanda riducendo al minimo il tempo di superamento di una soglia di allarme. Per ulteriori informazioni, vedere [Service Auto Scaling](#).

Se utilizzi EC2 istanze Amazon per fornire capacità di cluster, prendi in considerazione i seguenti consigli:

Usa EC2 istanze Amazon più grandi e volumi Amazon EBS più veloci. Puoi migliorare la velocità di download e preparazione delle immagini utilizzando un' EC2 istanza Amazon più grande e un volume Amazon EBS più veloce. All'interno di una determinata famiglia di istanze, il throughput massimo della rete e di Amazon EBS aumenta all'aumentare delle dimensioni dell'istanza (ad esempio, da m5.xlarge a m5.2xlarge). Inoltre, puoi personalizzare i volumi Amazon EBS per aumentarne il throughput e gli IOPS. Ad esempio, se utilizzi volumi, utilizza gp2 volumi più grandi che offrono una maggiore velocità di base. Se utilizzi gp3 volumi, specifica la velocità effettiva e gli IOPS quando crei il volume.

Usa la modalità di rete bridge per le attività in esecuzione su EC2 istanze Amazon. Le attività che utilizzano la modalità di bridge rete su Amazon vengono EC2 avviate più rapidamente delle attività che utilizzano la modalità aws-vpc di rete. Quando viene utilizzata la modalità di aws-vpc rete, Amazon ECS collega un'elastic network interface (ENI) all'istanza prima di avviare l'attività. Ciò introduce una latenza aggiuntiva. Tuttavia, ci sono diversi compromessi per l'utilizzo del bridge networking. Queste attività non dispongono di un proprio gruppo di sicurezza e vi sono alcune implicazioni per il bilanciamento del carico. Per ulteriori informazioni, consulta [Load Balancer target group](#) nella Elastic Load Balancing User Guide.

Gestione degli shock legati alla domanda

Alcune applicazioni subiscono improvvisi forti shock della domanda. Ciò accade per una serie di motivi: un evento giornalistico, una grande vendita, un evento mediatico o qualche altro evento che diventa virale e causa un aumento rapido e significativo del traffico in un lasso di tempo molto breve. Se non pianificato, ciò può far sì che la domanda superi rapidamente le risorse disponibili.

Il modo migliore per gestire gli shock legati alla domanda è prevederli e pianificare di conseguenza. Poiché la scalabilità automatica può richiedere del tempo, consigliamo di ridimensionare l'applicazione prima che cominci lo shock della domanda. Per ottenere risultati ottimali, consigliamo di adottare un piano aziendale che preveda una stretta collaborazione tra i team che utilizzano un calendario condiviso. Il team che pianifica l'evento dovrebbe lavorare in anticipo a stretto contatto con il team responsabile della candidatura. Questo dà al team abbastanza tempo per avere un piano di pianificazione chiaro. Possono pianificare la scalabilità orizzontale prima dell'evento e ampliarla dopo l'evento. Per ulteriori informazioni, consulta [Dimensionamento pianificato](#) nella Guida per l'utente di Dimensionamento automatico delle applicazioni.

Se disponi di un piano Enterprise Support, assicurati di collaborare anche con il tuo Technical Account Manager (TAM). Il TAM può verificare le quote di servizio e garantire che le quote necessarie vengano aumentate prima dell'inizio dell'evento. In questo modo, non si raggiungono accidentalmente le quote di servizio. Inoltre, possono aiutarvi con servizi di preriscaldamento, come i sistemi di bilanciamento del carico, per garantire che l'evento si svolga senza intoppi.

La gestione degli shock imprevisti della domanda è un problema più difficile. Gli shock non programmati, se di ampiezza sufficientemente elevata, possono far sì che la domanda superi rapidamente la capacità. Può anche superare la capacità di reazione della scalabilità automatica. Il modo migliore per prepararsi agli shock non programmati è quello di fornire risorse in eccesso. È necessario disporre di risorse sufficienti per gestire la massima richiesta di traffico prevista in qualsiasi momento.

Mantenere la capacità massima in previsione di shock non programmati della domanda può essere costoso. Per mitigare l'impatto sui costi, individuate un indicatore, una metrica o un evento anticipatore che preveda l'imminenza di un forte shock della domanda. Se la metrica o l'evento forniscono in modo affidabile un preavviso significativo, avvia il processo di scalabilità orizzontale immediatamente quando si verifica l'evento o quando la metrica supera la soglia specifica che hai impostato.

Se la tua applicazione è soggetta a improvvisi shock di domanda non programmati, prendi in considerazione l'aggiunta di una modalità ad alte prestazioni all'applicazione che sacrifichi le funzionalità non critiche ma mantenga funzionalità cruciali per il cliente. Ad esempio, supponiamo che l'applicazione possa passare dalla generazione di costose risposte personalizzate alla pubblicazione di una pagina di risposta statica. In questo scenario, è possibile aumentare in modo significativo la velocità effettiva senza scalare affatto l'applicazione.

Potete prendere in considerazione la possibilità di smontare i servizi monolitici per far fronte meglio agli shock della domanda. Se la tua applicazione è un servizio monolitico, costoso da eseguire e

lento da scalare, potresti riuscire a estrarre o riscrivere parti essenziali per le prestazioni ed eseguirle come servizi separati. Questi nuovi servizi possono quindi essere scalati indipendentemente dai componenti meno critici. La flessibilità necessaria per scalare le funzionalità essenziali in termini di prestazioni separatamente dalle altre parti dell'applicazione può ridurre il tempo necessario per aggiungere capacità e contribuire a ridurre i costi.

Best practice di rete di Amazon ECS

Le applicazioni moderne sono in genere costituite da più componenti distribuiti che comunicano tra loro. Ad esempio, un'applicazione mobile o web potrebbe comunicare con un endpoint API e l'API potrebbe essere alimentata da più microservizi che comunicano su Internet.

Per informazioni sulle migliori pratiche per le applicazioni di connessione a Internet, consulta.

[Connetti le applicazioni Amazon ECS a Internet](#)

Per informazioni sulle best practice per ricevere connessioni in entrata ad Amazon ECS da Internet, consulta. [Best practice per ricevere connessioni in entrata ad Amazon ECS da Internet](#)

Per informazioni sulle best practice per connettere Amazon ECS ad altri AWS servizi, consulta [Le migliori pratiche per connettere Amazon ECS ai AWS servizi dall'interno del tuo VPC](#).

Per informazioni sulle best practice per connettere i servizi all'interno di un VPC, consulta. [Best practice per connettere i servizi Amazon ECS in un VPC](#)

Per informazioni sulle migliori pratiche per i servizi di rete in Account AWS e VPCs, vedi [Best practice per il collegamento in rete dei servizi Amazon ECS tra e Account AWS VPCs](#).

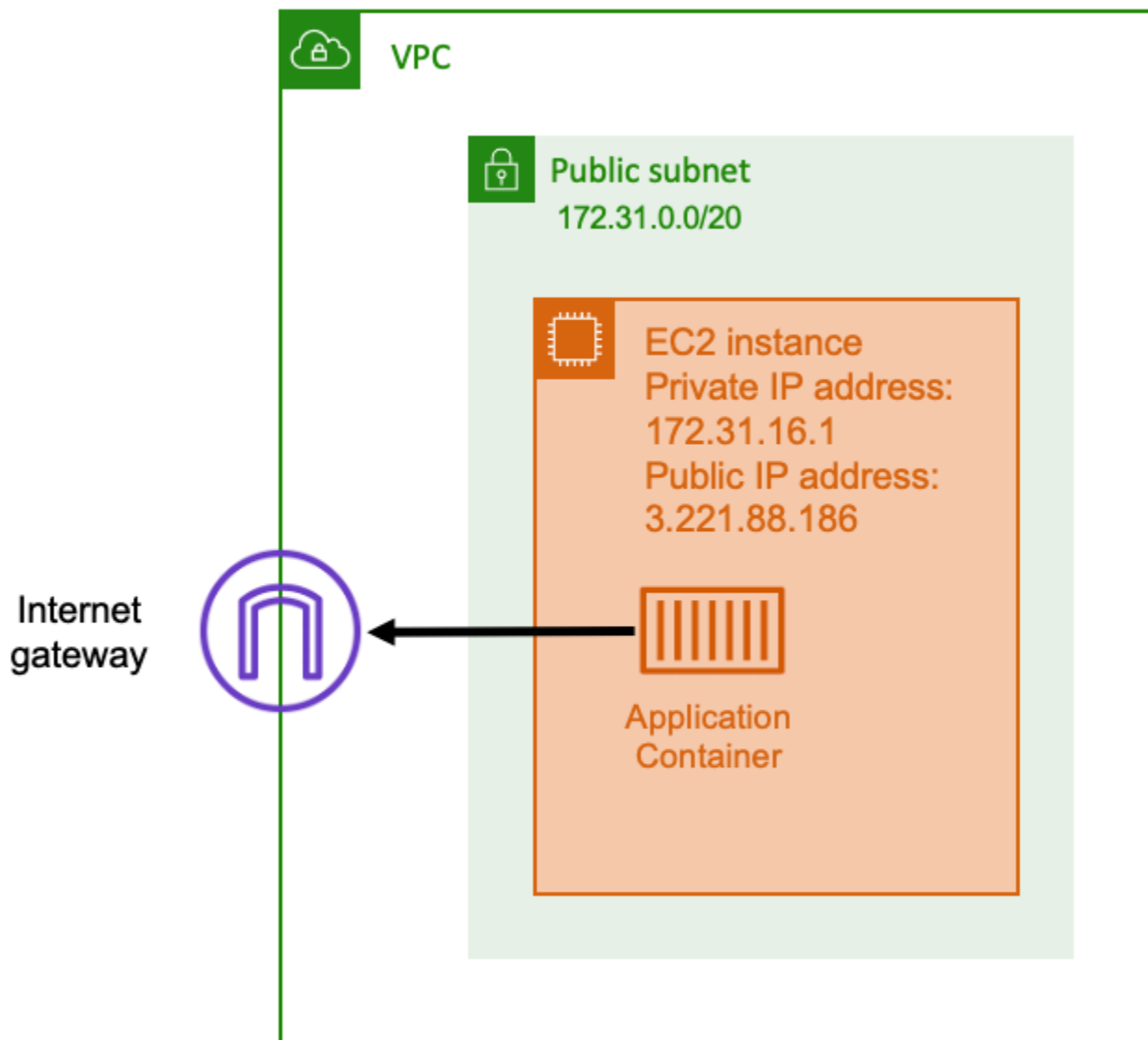
Per informazioni sulle migliori pratiche relative ai servizi per la risoluzione dei problemi di rete, vedere. [AWS servizi per la risoluzione dei problemi di rete Amazon ECS](#)

Connetti le applicazioni Amazon ECS a Internet

La maggior parte delle applicazioni containerizzate ha almeno alcuni componenti che richiedono l'accesso in uscita a Internet. Ad esempio, il backend di un'app mobile richiede l'accesso in uscita alle notifiche push.

Amazon Virtual Private Cloud offre due metodi principali per facilitare la comunicazione tra il tuo VPC e Internet.

Sottorete pubblica e gateway Internet



Quando si utilizza una sottorete pubblica con un percorso verso un gateway Internet, l'applicazione containerizzata può essere eseguita su un host all'interno di un VPC su una sottorete pubblica. All'host che gestisce il contenitore viene assegnato un indirizzo IP pubblico. Questo indirizzo IP pubblico è instradabile da Internet. Per ulteriori informazioni, consulta la sezione [Gateway Internet](#) nella Guida per l'utente di Amazon VPC.

Questa architettura di rete facilita la comunicazione diretta tra l'host che esegue l'applicazione e altri host su Internet. La comunicazione è bidirezionale. Ciò significa che non solo puoi stabilire una connessione in uscita con qualsiasi altro host su Internet, ma anche altri host su Internet potrebbero tentare di connettersi al tuo host. Pertanto, è necessario prestare molta attenzione al gruppo di

sicurezza e alle regole del firewall. Ciò garantisce che altri host su Internet non possano aprire connessioni che non desideri vengano aperte.

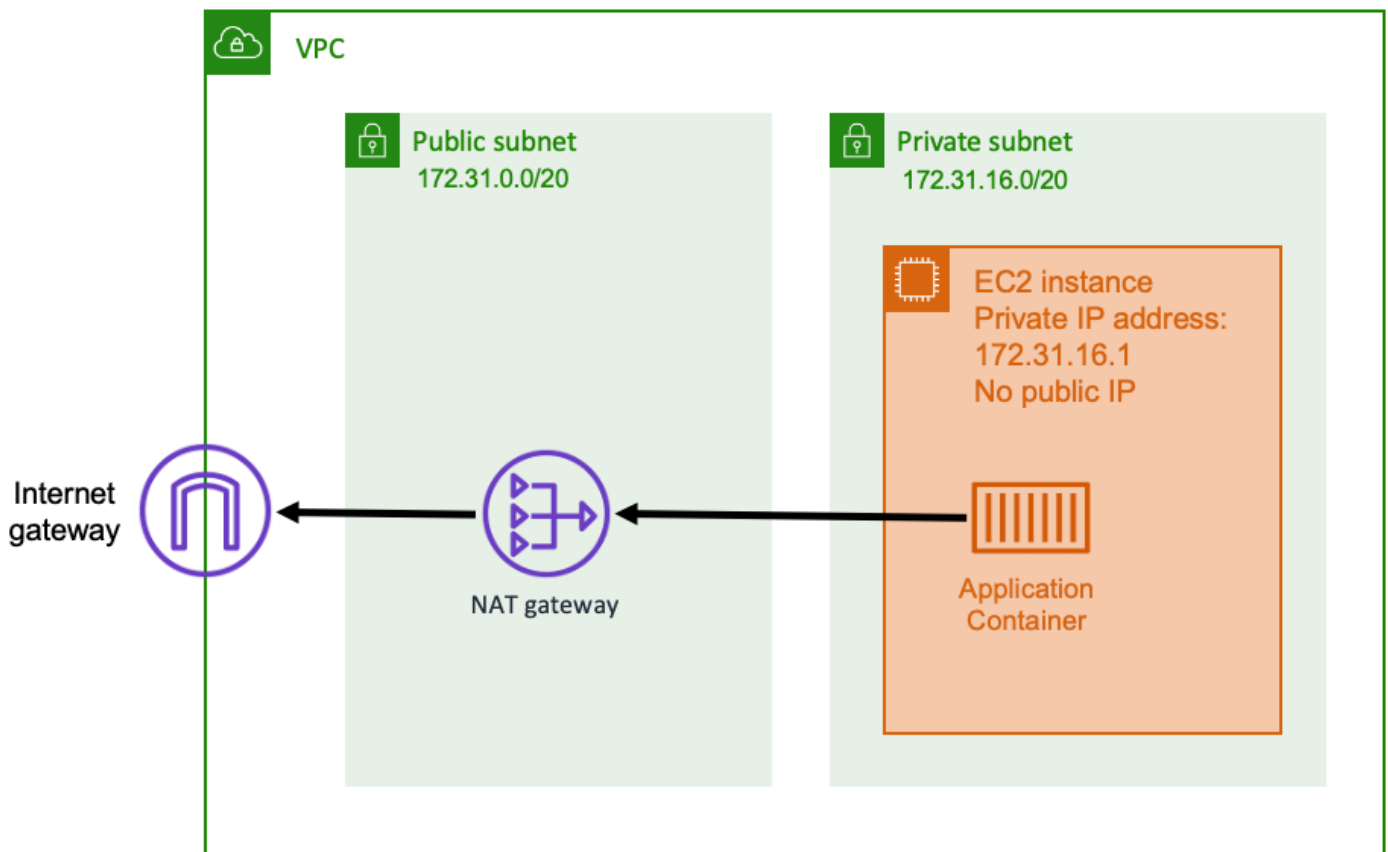
Ad esempio, se la tua applicazione viene eseguita su Amazon EC2, assicurati che la porta 22 per l'accesso SSH non sia aperta. Altrimenti, la tua istanza potrebbe ricevere continui tentativi di connessione SSH da bot malintenzionati su Internet. Questi bot esplorano gli indirizzi IP pubblici. Dopo aver trovato una porta SSH aperta, tentano di utilizzare password con forza bruta per tentare di accedere alla tua istanza. Per questo motivo, molte organizzazioni limitano l'uso delle sottoreti pubbliche e preferiscono avere la maggior parte, se non tutte, le risorse all'interno di sottoreti private.

L'uso di sottoreti pubbliche per la rete è adatto per applicazioni pubbliche che richiedono grandi quantità di larghezza di banda o una latenza minima. I casi d'uso applicabili includono lo streaming video e i servizi di gioco.

Questo approccio di rete è supportato sia quando usi Amazon ECS su Amazon EC2 sia quando lo usi su AWS Fargate.

- Amazon EC2 : puoi avviare EC2 istanze su una sottorete pubblica. Amazon ECS utilizza queste EC2 istanze come capacità del cluster e qualsiasi contenitore in esecuzione sulle istanze può utilizzare l'indirizzo IP pubblico sottostante dell'host per le reti in uscita. Questo vale sia per le modalità di rete che per quelle di retehost. bridge Tuttavia, la modalità di awsvpc rete non fornisce attività ENIs con indirizzi IP pubblici. Pertanto, non possono utilizzare direttamente un gateway Internet.
- Fargate: quando crei il tuo servizio Amazon ECS, specifica le sottoreti pubbliche per la configurazione di rete del servizio e utilizza l'opzione Assegna indirizzo IP pubblico. Ogni attività Fargate è collegata in rete nella sottorete pubblica e dispone di un proprio indirizzo IP pubblico per la comunicazione diretta con Internet.

Sottorete privata e gateway NAT



Quando si utilizza una sottorete privata e un gateway NAT, è possibile eseguire l'applicazione containerizzata su un host che si trova in una sottorete privata. Pertanto, questo host dispone di un indirizzo IP privato che è instradabile all'interno del tuo VPC, ma non è instradabile da Internet. Ciò significa che altri host all'interno del VPC possono connettersi all'host utilizzando il suo indirizzo IP privato, ma gli altri host su Internet non possono effettuare comunicazioni in entrata verso l'host.

Con una sottorete privata, è possibile utilizzare un gateway NAT (Network Address Translation) per consentire a un host all'interno di una sottorete privata di connettersi a Internet. Gli host su Internet ricevono una connessione in entrata che sembra provenire dall'indirizzo IP pubblico del gateway NAT che si trova all'interno di una sottorete pubblica. Il gateway NAT funge da ponte tra Internet e la sottorete privata. Questa configurazione è spesso preferita per motivi di sicurezza perché significa che il tuo VPC è protetto dall'accesso diretto degli aggressori su Internet. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC.

Questo approccio di rete privata è adatto per scenari in cui si desidera proteggere i contenitori dall'accesso esterno diretto. Gli scenari applicabili includono sistemi di elaborazione dei pagamenti o

contenitori che archiviano dati e password degli utenti. Ti vengono addebitati solo i costi di creazione e di utilizzo di un gateway NAT nell'account. Si applicano anche le tariffe orarie di utilizzo e di elaborazione dei dati del gateway NAT. Ai fini della ridondanza, è necessario disporre di un gateway NAT in ogni zona di disponibilità. In questo modo, la perdita di disponibilità di una singola zona di disponibilità non compromette la connettività in uscita. Per questo motivo, se hai un carico di lavoro ridotto, potrebbe essere più conveniente utilizzare sottoreti private e gateway NAT.

Questo approccio di rete è supportato sia quando si utilizza Amazon ECS su Amazon EC2 sia quando lo si utilizza su AWS Fargate.

- Amazon EC2 : puoi avviare EC2 istanze su una sottorete privata. I contenitori che vengono eseguiti su questi EC2 host utilizzano la rete degli host sottostanti e le richieste in uscita passano attraverso il gateway NAT.
- Fargate: quando crei il servizio Amazon ECS, specificate le sottoreti private per la configurazione di rete del servizio e non utilizzate l'opzione Assegna indirizzo IP pubblico. Ogni attività Fargate è ospitata in una sottorete privata. Il suo traffico in uscita viene instradato attraverso qualsiasi gateway NAT associato a quella sottorete privata.

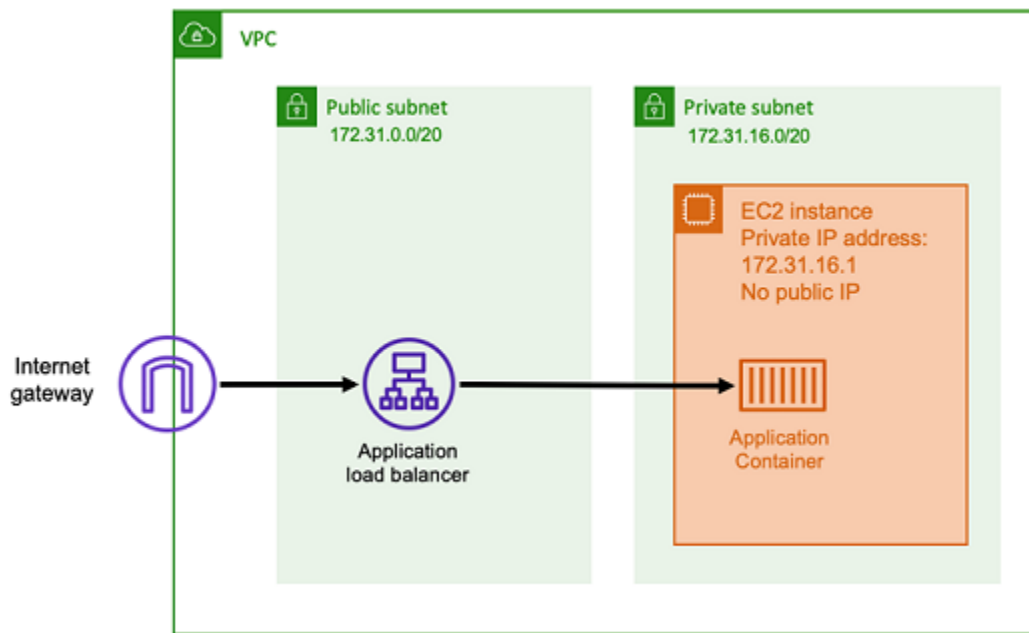
Best practice per ricevere connessioni in entrata ad Amazon ECS da Internet

Se gestisci un servizio pubblico, devi accettare il traffico in entrata da Internet. Ad esempio, il sito Web pubblico deve accettare richieste HTTP in entrata dai browser. In tal caso, anche altri host su Internet devono avviare una connessione in entrata all'host dell'applicazione.

Un approccio a questo problema consiste nell'avviare i contenitori su host che si trovano in una sottorete pubblica con un indirizzo IP pubblico. Tuttavia, non lo consigliamo per applicazioni su larga scala. Per queste, un approccio migliore consiste nell'avere un livello di input scalabile che si colloca tra Internet e l'applicazione. Per questo approccio, è possibile utilizzare uno qualsiasi dei AWS servizi elencati in questa sezione come input.

Application Load Balancer

Un Application Load Balancer funziona a livello di applicazione. È il settimo livello del modello Open Systems Interconnection (OSI). Ciò rende un Application Load Balancer adatto ai servizi HTTP pubblici. Se disponi di un sito Web o di un'API REST HTTP, un Application Load Balancer è un sistema di bilanciamento del carico adatto per questo carico di lavoro. Per ulteriori informazioni, consulta [Cos'è un Application Load Balancer?](#) nella Guida per l'utente di Application Load Balancers.



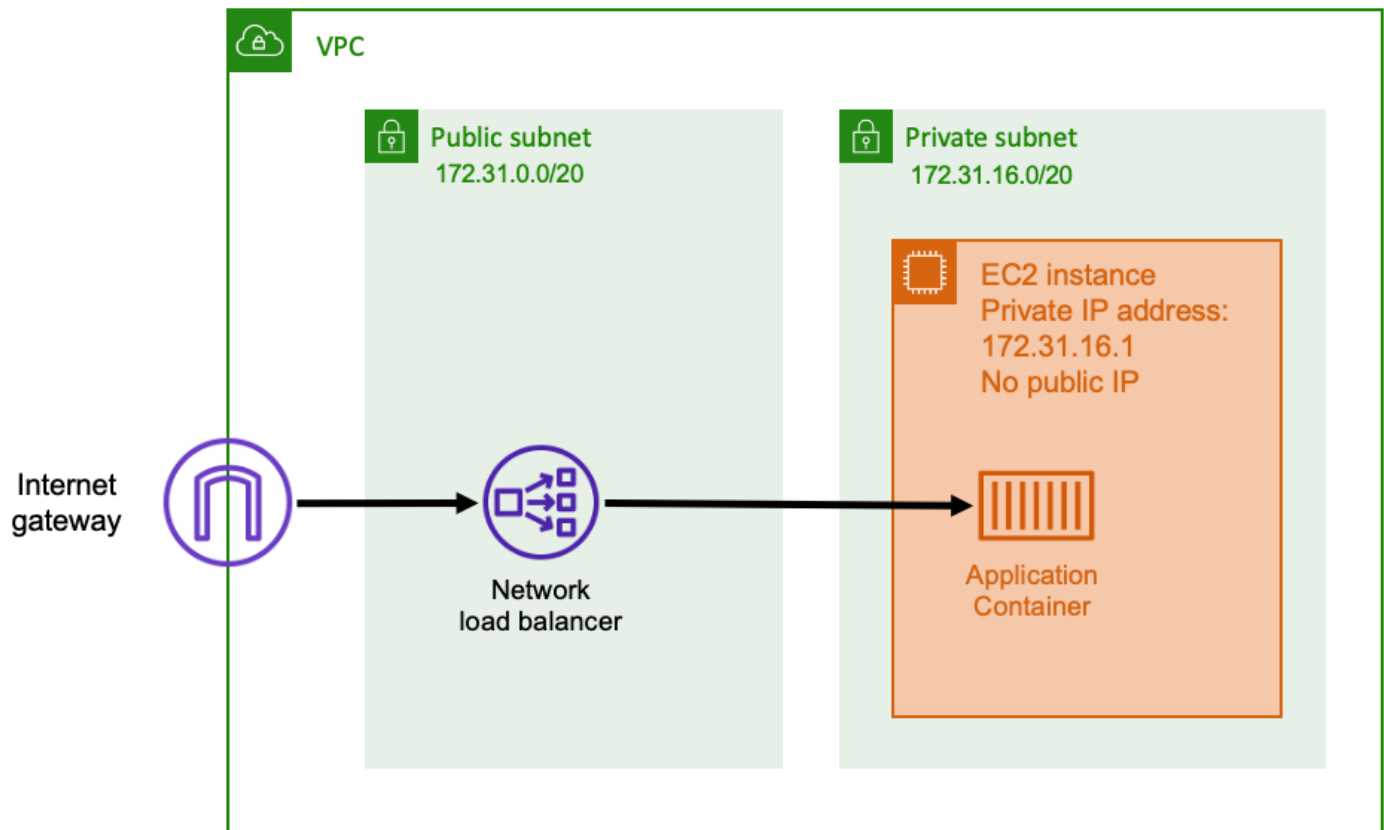
Con questa architettura, si crea un Application Load Balancer in una sottorete pubblica in modo che abbia un indirizzo IP pubblico e possa ricevere connessioni in entrata da Internet. Quando l'Application Load Balancer riceve una connessione in entrata, o più specificamente una richiesta HTTP, apre una connessione all'applicazione utilizzando il suo indirizzo IP privato. Quindi, inoltra la richiesta tramite la connessione interna.

Un Application Load Balancer presenta i seguenti vantaggi.

- **Terminazione SSL/TLS:** un Application Load Balancer può supportare comunicazioni e certificati HTTPS sicuri per le comunicazioni con i client. Facoltativamente, può interrompere la connessione SSL a livello di load balancer in modo da non dover gestire i certificati nella propria applicazione.
- **Routing avanzato:** un Application Load Balancer può avere più nomi host DNS. Dispone inoltre di funzionalità di routing avanzate per inviare richieste HTTP in entrata a diverse destinazioni in base a metriche come il nome host o il percorso della richiesta. Ciò significa che è possibile utilizzare un singolo Application Load Balancer come input per molti servizi interni diversi o anche microservizi su percorsi diversi di un'API REST.
- **Supporto gRPC e websocket:** un Application Load Balancer può gestire più di un semplice HTTP. Può anche bilanciare il carico di servizi basati su gRPC e websocket, con supporto HTTP/2.
- **Sicurezza:** un Application Load Balancer aiuta a proteggere l'applicazione dal traffico dannoso. Include funzionalità come le mitigazioni della sincronizzazione HTTP ed è integrato con AWS Web Application Firewall (AWS WAF). AWS WAF può filtrare ulteriormente il traffico dannoso che potrebbe contenere schemi di attacco, come SQL injection o cross-site scripting.

Network Load Balancer

Un sistema Network Load Balancer funziona al quarto livello del modello Open Systems Interconnection (OSI). È adatto per protocolli o scenari non HTTP in cui è necessaria end-to-end la crittografia, ma non ha le stesse funzionalità specifiche HTTP di un Application Load Balancer. Pertanto, un Network Load Balancer è più adatto per le applicazioni che non utilizzano HTTP. Per ulteriori informazioni, consulta [Cos'è un Network Load Balancer?](#) nella Guida per l'utente di Network Load Balancers.



Quando un Network Load Balancer viene utilizzato come input, funziona in modo simile a un Application Load Balancer. Questo perché è creato in una sottorete pubblica e dispone di un indirizzo IP pubblico a cui è possibile accedere su Internet. Il Network Load Balancer apre quindi una connessione all'indirizzo IP privato dell'host che esegue il container e invia i pacchetti dal lato pubblico a quello privato.

Funzionalità di Network Load Balancer

Poiché il Network Load Balancer opera a un livello inferiore dello stack di rete, non ha lo stesso set di funzionalità di Application Load Balancer. Tuttavia, presenta le seguenti caratteristiche importanti.

- End-to-end crittografia: poiché un Network Load Balancer opera al quarto livello del modello OSI, non legge il contenuto dei pacchetti. Ciò lo rende adatto per il bilanciamento del carico di comunicazioni che richiedono la crittografia. end-to-end
- Crittografia TLS: oltre alla end-to-end crittografia, Network Load Balancer può anche interrompere le connessioni TLS. In questo modo, le tue applicazioni di backend non devono implementare il proprio TLS.
- Supporto UDP: poiché un Network Load Balancer opera al quarto livello del modello OSI, è adatto per carichi di lavoro e protocolli non HTTP diversi dal TCP.

Chiusura delle connessioni

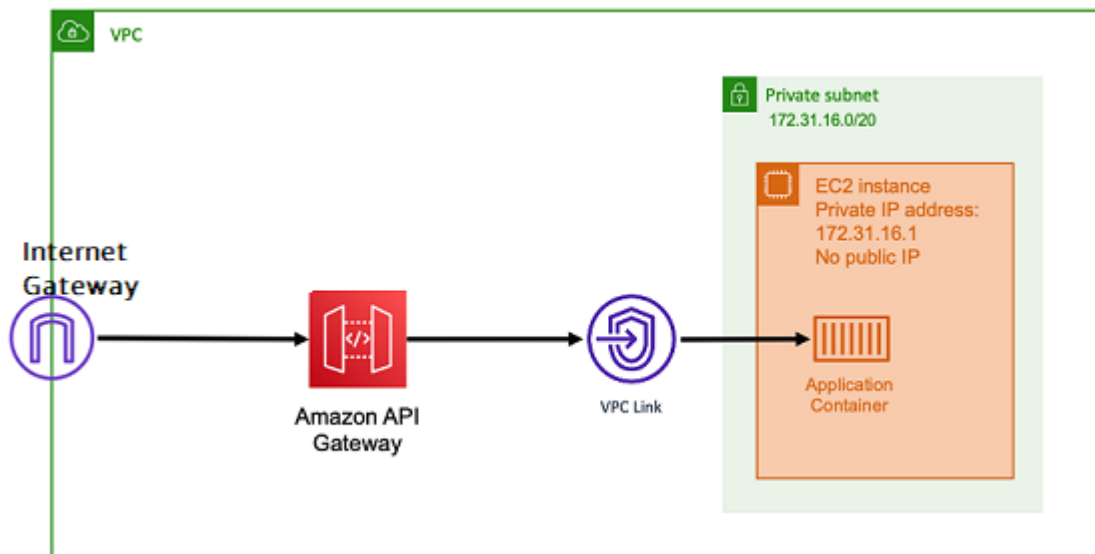
Poiché Network Load Balancer non rispetta il protocollo applicativo ai livelli superiori del modello OSI, non può inviare messaggi di chiusura ai client in tali protocolli. A differenza dell'Application Load Balancer, tali connessioni devono essere chiuse dall'applicazione oppure è possibile configurare Network Load Balancer per chiudere le connessioni di quarto livello quando un'attività viene interrotta o sostituita. Vedi l'impostazione di terminazione della connessione per i gruppi target di Network Load Balancer nella documentazione di [Network Load Balancer](#).

Consentire al Network Load Balancer di chiudere le connessioni al quarto livello può far sì che i client visualizzino messaggi di errore indesiderati, se il client non li gestisce. Per ulteriori informazioni sulla configurazione del client consigliata, consulta la libreria per gli sviluppatori [qui](#).

I metodi per chiudere le connessioni variano a seconda dell'applicazione, tuttavia un modo è garantire che il ritardo di annullamento della registrazione del target di Network Load Balancer sia più lungo del timeout della connessione client. Il client prima andava in timeout e si riconnetteva senza problemi tramite Network Load Balancer all'attività successiva, mentre la vecchia operazione esauriva lentamente tutti i client. [Per ulteriori informazioni sul ritardo di annullamento della registrazione previsto per Network Load Balancer, consulta la documentazione di Network Load Balancer.](#)

API HTTP di Amazon API Gateway

Amazon API Gateway è adatto per applicazioni HTTP con picchi improvvisi di volumi di richieste o volumi di richieste bassi. Per ulteriori informazioni, consulta [Cos'è Amazon API Gateway?](#) nella Guida per sviluppatori di API Gateway.



Il modello di prezzo per Application Load Balancer e Network Load Balancer include una tariffa oraria per mantenere i sistemi di bilanciamento del carico disponibili per accettare connessioni in entrata in ogni momento. Al contrario, API Gateway addebita separatamente ogni richiesta. Ciò ha l'effetto che, se non arriva alcuna richiesta, non ci sono costi. In caso di carichi di traffico elevati, un Application Load Balancer o un Network Load Balancer è in grado di gestire un volume maggiore di richieste a un prezzo per richiesta inferiore rispetto all'API Gateway. Tuttavia, se hai un numero complessivo di richieste limitato o hai periodi di traffico ridotto, il prezzo cumulativo per l'utilizzo dell'API Gateway dovrebbe essere più conveniente rispetto al pagamento di una tariffa oraria per mantenere un sistema di bilanciamento del carico sottoutilizzato. L'API Gateway può anche memorizzare nella cache le risposte API, il che potrebbe comportare una riduzione delle percentuali di richieste di backend.

Funzioni API Gateway che utilizzano un collegamento VPC che consente al servizio AWS gestito di connettersi agli host all'interno della sottorete privata del VPC, utilizzando il relativo indirizzo IP privato. Può rilevare questi indirizzi IP privati esaminando i record di rilevamento dei servizi gestiti da Amazon ECS Service Discovery.

API Gateway supporta le seguenti funzionalità.

- Il funzionamento dell'API Gateway è simile a quello di un sistema di bilanciamento del carico, ma presenta funzionalità aggiuntive esclusive per la gestione delle API
- L'API Gateway offre funzionalità aggiuntive relative all'autorizzazione del client, ai livelli di utilizzo e alla modifica della richiesta/risposta. Per ulteriori informazioni, consulta le [funzionalità di Amazon API Gateway](#).

- L'API Gateway può supportare endpoint gateway API edge, regionali e privati. Gli endpoint Edge sono disponibili tramite una distribuzione gestita CloudFront . Gli endpoint regionali e privati sono entrambi locali rispetto a una regione.
- Terminazione SSL/TLS
- Instradamento di diversi percorsi HTTP verso diversi microservizi di backend

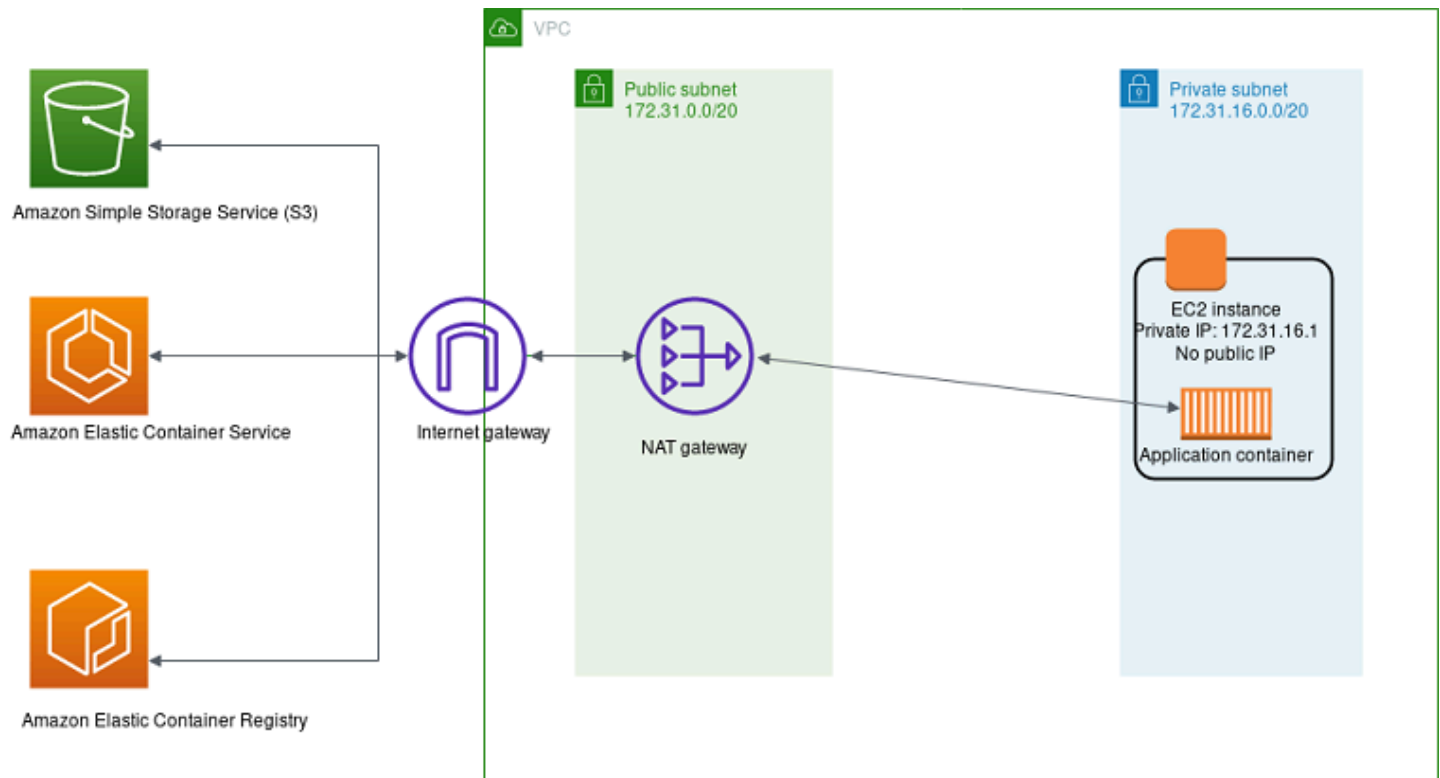
Oltre alle funzionalità precedenti, API Gateway supporta anche l'utilizzo di autorizzatori Lambda personalizzati che puoi utilizzare per proteggere l'API dall'utilizzo non autorizzato. Per ulteriori informazioni, consulta [Field Notes: Serverless Container based with APIs Amazon ECS e Amazon API Gateway](#).

Le migliori pratiche per connettere Amazon ECS ai AWS servizi dall'interno del tuo VPC

Affinché Amazon ECS funzioni correttamente, l'agente container Amazon ECS in esecuzione su ciascun host deve comunicare con il piano di controllo di Amazon ECS. Se memorizzi le immagini dei contenitori in Amazon ECR, EC2 gli host Amazon devono comunicare con l'endpoint del servizio Amazon ECR e con Amazon S3, dove sono archiviati i livelli di immagine. Se utilizzi altri AWS servizi per la tua applicazione containerizzata, come i dati persistenti archiviati in DynamoDB, ricontrolla che anche questi servizi dispongano del supporto di rete necessario.

Gateway NAT

L'utilizzo di un gateway NAT è il modo più semplice per garantire che le attività di Amazon ECS possano accedere ad altri AWS servizi. Per ulteriori informazioni su questo approccio, consulta [Sottorete privata e gateway NAT](#)



Di seguito sono riportati gli svantaggi dell'utilizzo di questo approccio:

- Non è possibile limitare le destinazioni con cui il gateway NAT può comunicare. Inoltre, non puoi limitare le destinazioni con cui il tuo livello di backend può comunicare senza interrompere tutte le comunicazioni in uscita dal tuo VPC.
- I gateway NAT addebitano un costo per ogni GB di dati in transito. Se utilizzi il gateway NAT per una delle seguenti operazioni, ti verrà addebitato un costo per ogni GB di larghezza di banda:
 - Scaricamento di file di grandi dimensioni da Amazon S3
 - Esecuzione di un volume elevato di query di database su DynamoDB
 - Estrarre immagini da Amazon ECR

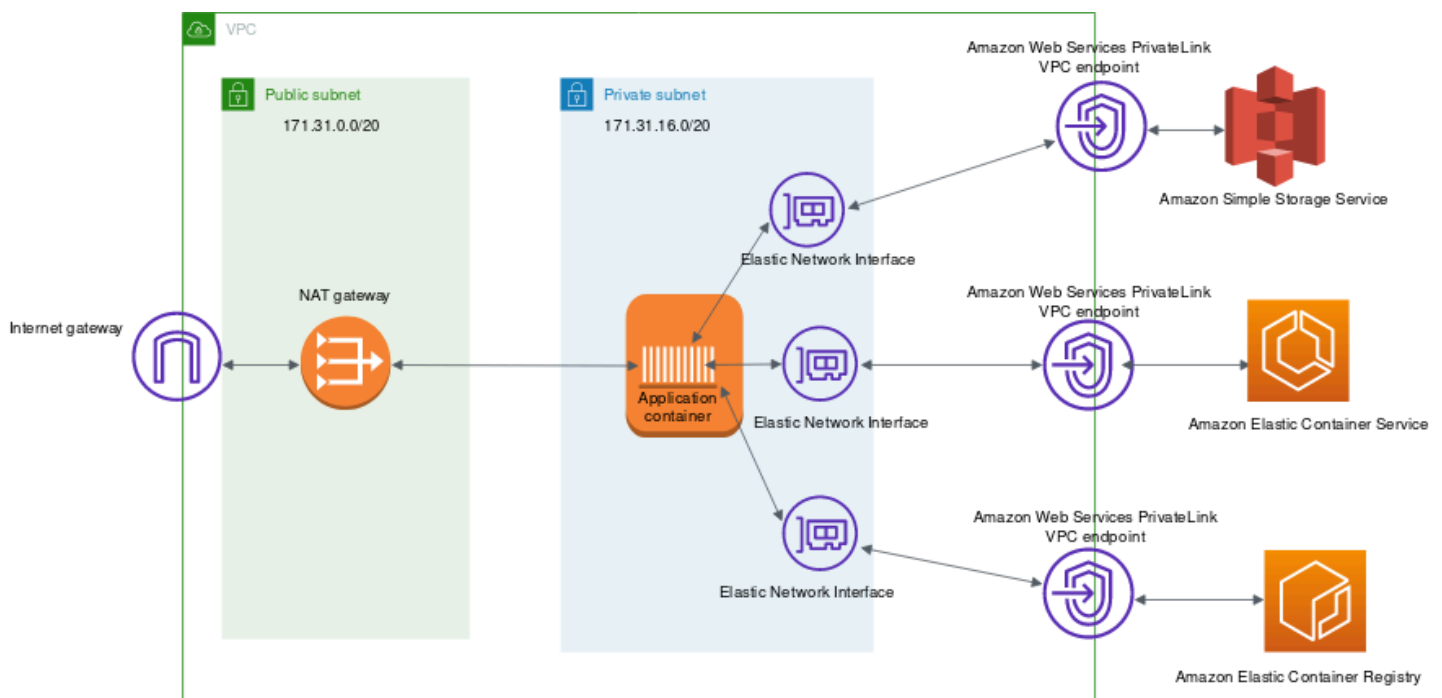
Inoltre, i gateway NAT supportano 5 Gbps di larghezza di banda e scalano automaticamente fino a 45 Gbps. Se si esegue il routing attraverso un singolo gateway NAT, le applicazioni che richiedono connessioni a larghezza di banda molto elevata potrebbero incontrare vincoli di rete. Come soluzione alternativa, puoi dividere il carico di lavoro su più sottoreti e assegnare a ciascuna sottorete il proprio gateway NAT.

AWS PrivateLink

AWS PrivateLink fornisce connettività privata tra i VPCs AWS servizi e le reti locali senza esporre il traffico alla rete Internet pubblica.

Un endpoint VPC consente connessioni private tra il tuo VPC e i servizi supportati AWS e i servizi endpoint VPC. Il traffico tra il VPC e gli altri servizi non lascia la rete Amazon. Un endpoint VPC non richiede un gateway Internet, un gateway privato virtuale, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect EC2Le istanze Amazon nel tuo VPC non richiedono indirizzi IP pubblici per comunicare con le risorse del servizio.

Il diagramma seguente mostra come funziona la comunicazione con AWS i servizi quando si utilizzano endpoint VPC anziché un gateway Internet. AWS PrivateLink fornisce interfacce di rete elastiche (ENIs) all'interno della sottorete e le regole di routing VPC vengono utilizzate per inviare qualsiasi comunicazione al nome host del servizio tramite l'ENI, direttamente al servizio di destinazione. AWS Questo traffico non deve più utilizzare il gateway NAT o il gateway Internet.



Di seguito sono riportati alcuni degli endpoint VPC più comuni utilizzati con il servizio Amazon ECS.

- [Endpoint gateway per Amazon S3](#)
- [Endpoint VPC DynamoDB](#)
- [Endpoint VPC Amazon ECS](#)
- [Endpoint VPC Amazon ECR](#)

Molti altri AWS servizi supportano gli endpoint VPC. Se fai un uso intensivo di qualsiasi AWS servizio, dovresti cercare la documentazione specifica per quel servizio e come creare un endpoint VPC per quel traffico.

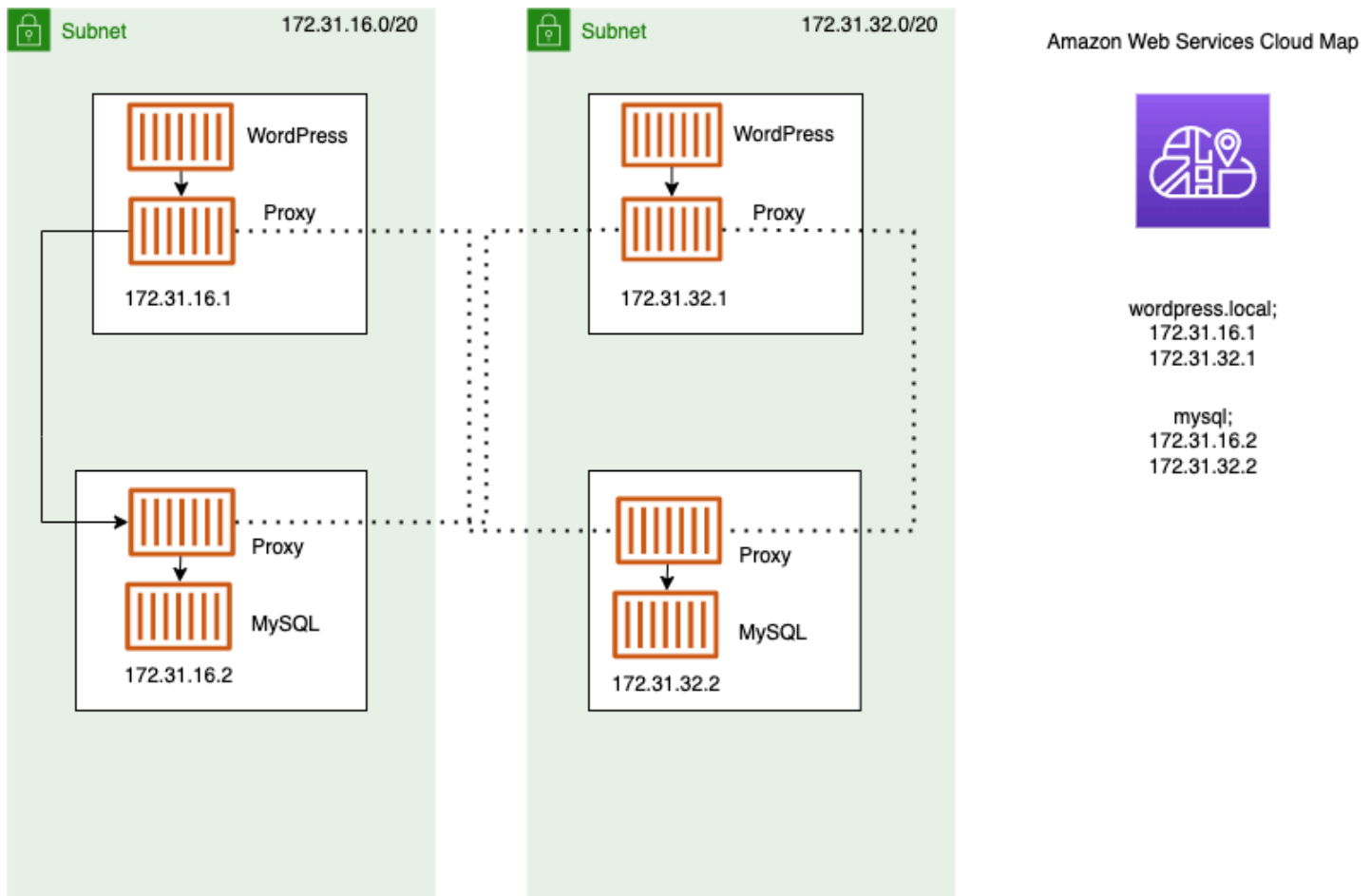
Best practice per connettere i servizi Amazon ECS in un VPC

Utilizzando le attività di Amazon ECS in un VPC, puoi suddividere le applicazioni monolitiche in parti separate che possono essere distribuite e scalate indipendentemente in un ambiente sicuro. Questa architettura si chiama architettura orientata ai servizi (SOA) o microservizi. Tuttavia, può essere difficile assicurarsi che tutte queste parti, sia all'interno che all'esterno di un VPC, possano comunicare tra loro. Esistono diversi approcci per facilitare la comunicazione, tutti con vantaggi e svantaggi diversi.

Utilizzo di Service Connect

Consigliamo Service Connect, che fornisce la configurazione di Amazon ECS per l'individuazione dei servizi, la connettività e il monitoraggio del traffico. Con Service Connect, le applicazioni possono utilizzare nomi brevi e porte standard per connettersi ai servizi nello stesso cluster, VPCs in altri cluster, anche all'interno della stessa regione. Per ulteriori informazioni, consulta [Amazon ECS Service Connect](#).

Quando usi Service Connect, Amazon ECS gestisce tutte le parti del service discovery: crea i nomi che possono essere scoperti, gestisce dinamicamente le voci per ogni attività all'inizio e all'interruzione delle attività, esegue un agente in ogni attività configurato per scoprire i nomi. L'applicazione può cercare i nomi utilizzando la funzionalità standard per i nomi DNS e stabilendo connessioni. Se l'applicazione lo fa già, non è necessario modificarla per utilizzare Service Connect.



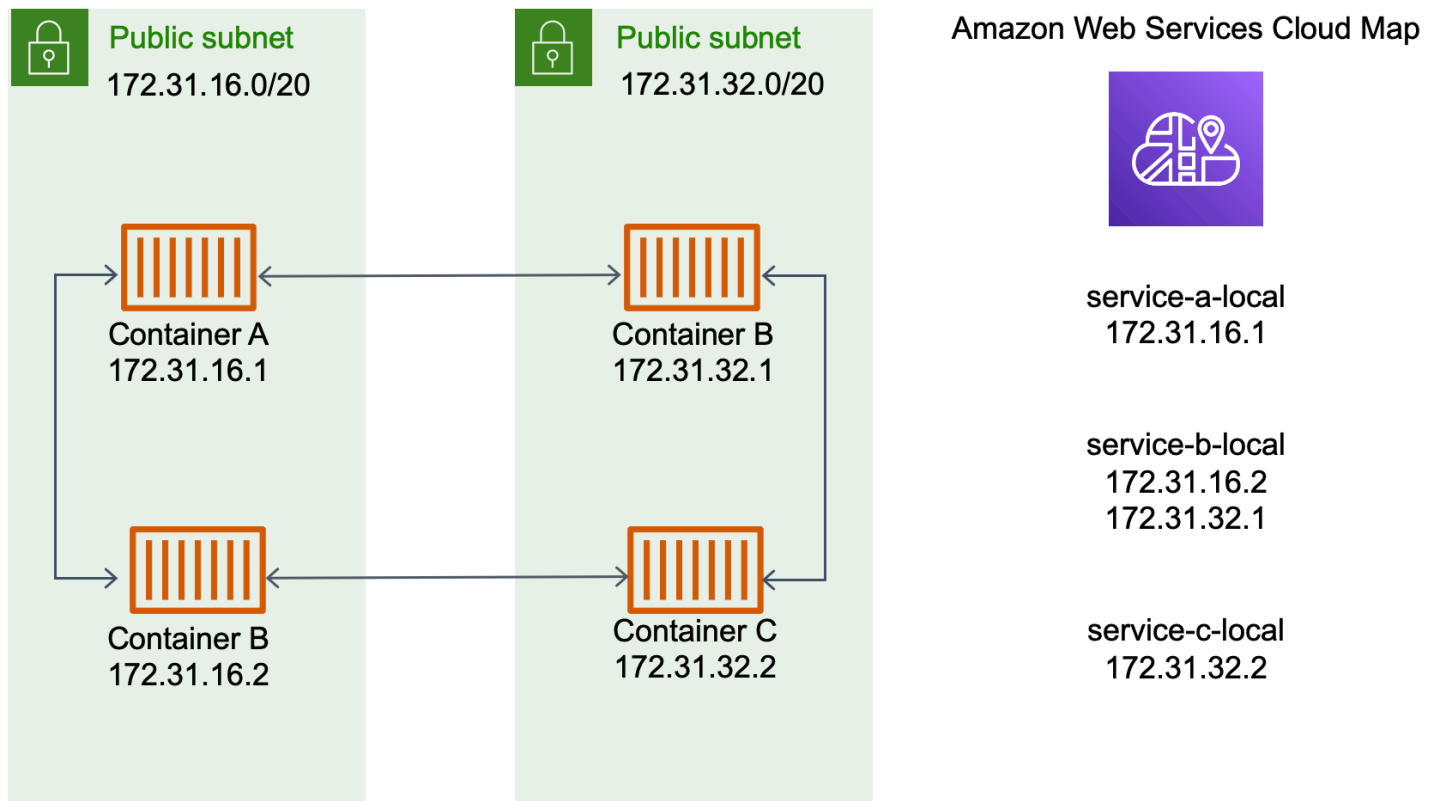
Le modifiche avvengono solo durante le distribuzioni

Fornisci la configurazione completa all'interno di ogni definizione di servizio e attività. Amazon ECS gestisce le modifiche a questa configurazione in ogni distribuzione del servizio, per garantire che tutte le attività di una distribuzione si comportino nello stesso modo. Ad esempio, un problema comune relativo al DNS as service discovery è il controllo di una migrazione. Se si modifica un nome DNS in modo che punti ai nuovi indirizzi IP sostitutivi, potrebbe essere necessario il tempo TTL massimo prima che tutti i client inizino a utilizzare il nuovo servizio. Con Service Connect, la distribuzione del client aggiorna la configurazione sostituendo le attività del client. È possibile configurare l'interruttore automatico di distribuzione e altre configurazioni di distribuzione per influire sulle modifiche di Service Connect allo stesso modo di qualsiasi altra distribuzione.

Utilizzo del rilevamento dei servizi

Un altro approccio alla service-to-service comunicazione è la comunicazione diretta tramite il service discovery. In questo approccio, puoi utilizzare l'integrazione del rilevamento dei AWS Cloud Map servizi con Amazon ECS. Utilizzando Service Discovery, Amazon ECS sincronizza l'elenco delle

attività avviate con AWS Cloud Map, che mantiene un nome host DNS che si risolve negli indirizzi IP interni di una o più attività di quel particolare servizio. Altri servizi in Amazon VPC possono utilizzare questo nome host DNS per inviare traffico direttamente a un altro contenitore utilizzando il suo indirizzo IP interno. Per ulteriori informazioni, consulta [Individuazione dei servizi](#).



Nel diagramma precedente, sono presenti tre servizi. `service-a-local` ha un contenitore e comunica con `service-b-local`, che ha due contenitori. `service-b-local` deve comunicare anche con `service-c-local`, che ha un contenitore. Ogni contenitore di tutti e tre questi servizi può utilizzare i nomi DNS interni di AWS Cloud Map per trovare gli indirizzi IP interni di un contenitore del servizio a valle con cui deve comunicare.

Questo approccio alla service-to-service comunicazione offre una bassa latenza. A prima vista, è anche semplice in quanto non ci sono componenti aggiuntivi tra i contenitori. Il traffico viaggia direttamente da un container all'altro.

Questo approccio è adatto quando si utilizza la modalità `awsvpc` di rete, in cui ogni attività ha il proprio indirizzo IP univoco. La maggior parte dei software supporta solo l'uso di A record DNS, che si risolvono direttamente in indirizzi IP. Quando si utilizza la modalità di `awsvpc` rete, l'indirizzo IP per ogni operazione è un A record. Tuttavia, se utilizzi la modalità di `bridge` rete, è possibile che più contenitori condividano lo stesso indirizzo IP. Inoltre, le mappature dinamiche delle porte fanno sì

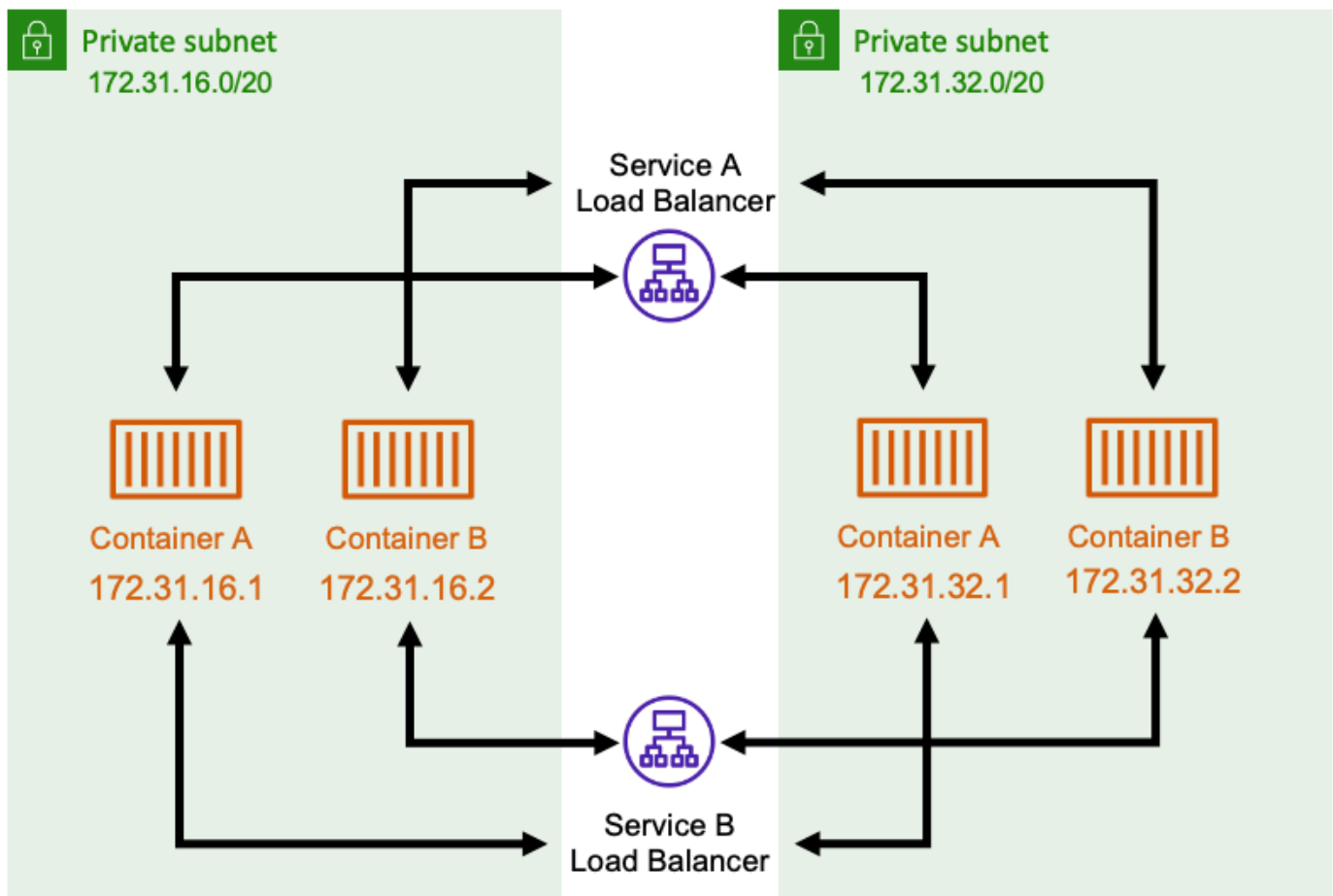
che ai contenitori vengano assegnati in modo casuale i numeri di porta su quel singolo indirizzo IP. A questo punto, un A record non è più sufficiente per l'individuazione del servizio. È inoltre necessario utilizzare un SRV record. Questo tipo di record può tenere traccia sia degli indirizzi IP che dei numeri di porta, ma richiede la configurazione appropriata delle applicazioni. Alcune applicazioni predefinite utilizzate potrebbero non supportare SRV i record.

Un altro vantaggio della modalità di `aws-vpc` rete è che si dispone di un gruppo di sicurezza unico per ogni servizio. È possibile configurare questo gruppo di sicurezza per consentire le connessioni in entrata solo dai servizi upstream specifici che devono comunicare con quel servizio.

Lo svantaggio principale della service-to-service comunicazione diretta tramite Service Discovery è che è necessario implementare una logica aggiuntiva per ripetere i tentativi e gestire gli errori di connessione. I record DNS hanno un periodo time-to-live (TTL) che controlla per quanto tempo vengono memorizzati nella cache. L'aggiornamento del record DNS e la scadenza della cache richiedono del tempo per consentire alle applicazioni di recuperare la versione più recente del record DNS. Pertanto, l'applicazione potrebbe finire per risolvere il record DNS in modo che punti a un altro contenitore che non è più presente. L'applicazione deve gestire i nuovi tentativi e disporre di una logica per ignorare i backend non validi.

Utilizzo di un sistema di bilanciamento del carico interno

Un altro approccio alla service-to-service comunicazione consiste nell'utilizzare un sistema di bilanciamento del carico interno. Un load balancer interno esiste interamente all'interno del tuo VPC ed è accessibile solo ai servizi all'interno del tuo VPC.



Il load balancer mantiene un'elevata disponibilità distribuendo risorse ridondanti in ogni sottorete. Quando un container from serviceA deve comunicare con un container da serviceB, apre una connessione al load balancer. Il sistema di bilanciamento del carico apre quindi una connessione a un contenitore da service B. Il load balancer funge da luogo centralizzato per la gestione di tutte le connessioni tra ciascun servizio.

Se un container di serviceB si ferma, il load balancer può rimuoverlo dal pool. Il load balancer esegue inoltre controlli di integrità su ogni target a valle del relativo pool e può rimuovere automaticamente gli obiettivi danneggiati dal pool finché non tornano a funzionare. Non è più necessario che le applicazioni siano consapevoli del numero di container a valle presenti. Si limitano ad aprire le proprie connessioni al sistema di bilanciamento del carico.

Questo approccio è vantaggioso per tutte le modalità di rete. Il load balancer può tenere traccia degli indirizzi IP delle attività quando si utilizza la modalità di aws-vpc rete, nonché delle combinazioni più avanzate di indirizzi IP e porte quando si utilizza la bridge modalità di rete. Distribuisce in modo

uniforme il traffico su tutte le combinazioni di indirizzi IP e porte, anche se diversi container sono effettivamente ospitati sulla stessa EC2 istanza Amazon, solo su porte diverse.

L'unico svantaggio di questo approccio è il costo. Per garantire un'elevata disponibilità, il sistema di bilanciamento del carico deve disporre di risorse in ogni zona di disponibilità. Ciò comporta costi aggiuntivi a causa del sovraccarico dovuto al pagamento del load balancer e alla quantità di traffico che attraversa il load balancer.

Tuttavia, è possibile ridurre i costi generali facendo in modo che più servizi condividano un sistema di bilanciamento del carico. Ciò è particolarmente adatto per i servizi REST che utilizzano un Application Load Balancer. È possibile creare regole di routing basate sui percorsi che indirizzano il traffico verso servizi diversi. Ad esempio, `/api/user/*` potrebbe indirizzare verso un contenitore che fa parte del `user` servizio, mentre `/api/order/*` potrebbe indirizzare verso il servizio associato `order`. Con questo approccio, paghi solo per un Application Load Balancer e disponi di un URL coerente per la tua API. Tuttavia, puoi suddividere il traffico verso vari microservizi sul backend.

Best practice per il collegamento in rete dei servizi Amazon ECS tra e Account AWS VPCs

Se fai parte di un'organizzazione con più team e divisioni, probabilmente distribuisce i servizi in modo indipendente VPCs all'interno di un ambiente condiviso Account AWS o in VPCs cui sono associati più individui. Account AWS Indipendentemente dal modo in cui distribuisce i tuoi servizi, ti consigliamo di integrare i componenti di rete per indirizzare il traffico tra di loro. VPCs A tal fine, è possibile utilizzare diversi AWS servizi per integrare i componenti di rete esistenti.

- **AWS Transit Gateway:** è necessario considerare prima questo servizio di rete. Questo servizio funge da hub centrale per il routing delle connessioni tra Amazon VPCs e Account AWS le reti locali. Per ulteriori informazioni, consulta [Cos'è un gateway di transito?](#) nella Amazon VPC Transit Gateways Guide.
- **Supporto Amazon VPC e VPN:** puoi utilizzare questo servizio per creare connessioni site-to-site VPN per connettere reti locali al tuo VPC. [Per ulteriori informazioni, consulta Cos'è? AWS Site-to-Site VPN](#) nella Guida AWS Site-to-Site VPN per l'utente.
- **Amazon VPC:** puoi utilizzare il peering Amazon VPC per aiutarti a connetterne più di uno VPCs, nello stesso account o tra più account. Per ulteriori informazioni, consulta [Che cos'è il peering di VPC?](#) nella Amazon VPC Peering Guide.
- **Condiviso VPCs :** puoi utilizzare un VPC e sottoreti VPC su più account. AWS Per ulteriori informazioni, consulta [Working with shared VPCs](#) nella Amazon VPC User Guide.

AWS servizi per la risoluzione dei problemi di rete Amazon ECS

I seguenti servizi e funzionalità possono aiutarvi a ottenere informazioni dettagliate sulle configurazioni di rete e dei servizi. È possibile utilizzare queste informazioni per risolvere i problemi di rete e ottimizzare meglio i servizi.

CloudWatch Container Insights

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Le metriche includono l'utilizzo di risorse come CPU, memoria, disco e rete. Sono disponibili nei dashboard CloudWatch automatici. Per ulteriori informazioni, consulta [Configurazione di Container Insights su Amazon ECS](#) nella Amazon CloudWatch User Guide.

AWS X-Ray

AWS X-Ray è un servizio di tracciamento che puoi utilizzare per raccogliere informazioni sulle richieste di rete effettuate dall'applicazione. Puoi utilizzare l'SDK per strumentare l'applicazione e acquisire i tempi e i codici di risposta del traffico tra i tuoi servizi e tra i tuoi servizi e gli endpoint del servizio. AWS Per ulteriori informazioni, consulta [Che cos'è AWS X-Ray?](#) nella Guida per gli sviluppatori di AWS X-Ray .

Puoi anche esplorare AWS X-Ray i grafici del modo in cui i tuoi servizi si collegano tra loro. Oppure, usali per esplorare le statistiche aggregate sulle prestazioni di ciascun service-to-service link. Infine, puoi approfondire ogni transazione specifica per vedere come i segmenti che rappresentano le chiamate di rete sono associati a quella particolare transazione.

Puoi utilizzare queste funzionalità per identificare se c'è un collo di bottiglia nella rete o se un servizio specifico all'interno della tua rete non funziona come previsto.

Log di flusso VPC

Puoi utilizzare i log di flusso di Amazon VPC per analizzare le prestazioni di rete ed eseguire il debug dei problemi di connettività. Con i log di flusso VPC abilitati, puoi acquisire un registro di tutte le connessioni nel tuo VPC. Queste includono connessioni a interfacce di rete associate a Elastic Load Balancing, Amazon RDS, gateway NAT e altri servizi AWS chiave che potresti utilizzare. Per ulteriori informazioni, consulta [Log di flusso VPC](#) nella Guida per l'utente di Amazon VPC.

Suggerimenti per l'ottimizzazione della rete

È possibile ottimizzare alcune impostazioni per migliorare la rete.

nessun limite di file

Se si prevede che l'applicazione abbia un traffico elevato e gestisca molte connessioni simultanee, è necessario tenere conto della quota di sistema per il numero di file consentiti. Quando ci sono molti socket di rete aperti, ognuno deve essere rappresentato da un descrittore di file. Se la quota di descrittori di file è troppo bassa, limiterà i socket di rete. Ciò si traduce in connessioni o errori non riusciti. Puoi aggiornare la quota specifica del contenitore per il numero di file nella definizione del task di Amazon ECS. Se utilizzi Amazon EC2 (anziché AWS Fargate), potresti dover modificare queste quote anche sull' EC2 istanza Amazon sottostante.

sysctl net

Un'altra categoria di impostazioni regolabili sono le impostazioni di rete. `sysctl` È necessario fare riferimento alle impostazioni specifiche della distribuzione Linux scelta. Molte di queste impostazioni regolano la dimensione dei buffer di lettura e scrittura. Questo può essere utile in alcune situazioni quando si eseguono EC2 istanze Amazon su larga scala che contengono molti contenitori.

Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account

È possibile accedere alle impostazioni dell'account Amazon ECS per attivare o disattivare determinate caratteristiche. Per ogni Regione AWS, puoi attivare o disattivare ogni impostazione dell'account a livello di account o per un utente o ruolo specifico.

È possibile attivare o disattivare determinate caratteristiche se una delle seguenti caratteristiche è rilevante per te:

- Un utente o ruolo può attivare o disattivare specifiche impostazioni del singolo account utente.
- Un utente o ruolo può configurare l'impostazione di attivazione o disattivazione predefinita per tutti gli utenti nell'account.
- L'utente root o un utente con privilegi di amministratore può attivare o disattivare qualsiasi ruolo o utente specifico sull'account. Se l'impostazione dell'account per l'utente root viene modificata, l'impostazione di default viene configurata per tutti gli utenti e i ruoli per i quali non è stata selezionata una singola impostazione dell'account.

Note

Gli utenti federati presuppongono l'impostazione dell'account dell'utente root e non possono avere impostazioni account esplicite impostate per loro separatamente.

Sono disponibili le seguenti impostazioni dell'account. È necessario attivare e disattivare separatamente ciascuna impostazione dell'account.

Nome risorsa	Ulteriori informazioni
containerInsights	Container Insights
serviceLongArnFormat	Nomi di risorse Amazon (ARNs) e IDs
taskLongArnFormat	
containerInstanceLongArnFormat	
tagResourceAuthorization	Autorizzazione all'assegnazione di tag
fargateFIPSMODE	AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)
fargateTaskRetirementWaitPeriod	AWS Fargate tempo di attesa per il ritiro dell'attività
guardDutyActivate	Monitoraggio del runtime (GuardDuty integrazione con Amazon)
dualStackIPv6	VPC a doppio stack IPv6
awsvpcTrunking	Aumenta le interfacce di rete delle istanze di container Linux
defaultLogDriverMode	Modalità predefinita del driver di registro

Nomi di risorse Amazon (ARNs) e IDs

Quando vengono create le risorse Amazon ECS, a ogni risorsa viene assegnato un Amazon Resource Name (ARN) e un identificatore di risorsa (ID). Se utilizzi uno strumento da riga di comando o l'API Amazon ECS per lavorare con Amazon ECS, hai bisogno di risorse ARNs per determinati comandi. IDs Ad esempio, se si utilizza il AWS CLI comando [stop-task](#) per interrompere un'operazione, è necessario specificare l'ARN o l'ID dell'attività nel comando.

Amazon ECS ha introdotto un nuovo formato per Amazon Resource Names (ARNs) e risorse IDs per i servizi, le attività e le istanze di container di Amazon ECS. Lo stato di opt-in per ogni tipo di risorsa determina il formato del nome della risorsa Amazon (ARN) utilizzato dalla risorsa. È necessario attivare il nuovo formato dell'ARN per utilizzare funzionalità quali il tagging di risorse per tale tipo di risorsa.

L'attivazione e la disattivazione del nuovo formato del nome della risorsa Amazon (ARN) e degli ID risorsa è possibile in base alle singole regioni. Attualmente, qualsiasi nuovo account creato viene attivato di default.

Puoi fornire il consenso esplicito o il rifiuto esplicito del nuovo formato dell'Amazon Resource Name (ARN) e dell'ID risorsa in qualsiasi momento. Dopo aver effettuato l'attivazione, tutte le nuove risorse create utilizzeranno il nuovo formato.

Note

Un ID risorsa non può essere modificato dopo la sua creazione. Pertanto, l'attivazione o la disattivazione del nuovo formato non influirà sulla risorsa IDs esistente.

Le seguenti sezioni descrivono in che modo i formati dell'ARN e dell'ID risorsa stanno cambiando. Per ulteriori informazioni sulla transizione ai nuovi formati, consulta [Domande frequenti su Amazon Elastic Container Service](#).

Formato Amazon Resource Name (ARN)

Alcune risorse hanno un nome descrittivo, ad esempio un servizio denominato `production`. In altri casi, è necessario specificare una risorsa utilizzando il formato dell'Amazon Resource Name (ARN). Il nuovo formato dell'ARN per processi, servizi e istanze di container Amazon ECS include il nome del cluster. Per informazioni sull'attivazione con il formato dell'ARN, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

La tabella seguente mostra sia il formato attuale (precedente) sia il nuovo formato per ogni tipo di risorsa.

Tipo di risorsa	ARN
Istanza del container	arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :container-instance/<i>container-instance-id</i> correnti Nuovo: arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :container-instance/<i>cluster-name</i> /<i>container-instance-id</i>
Servizio Amazon ECS	arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :service/<i>service-name</i> correnti Nuovo: arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :service/<i>cluster-name</i> /<i>service-name</i>
Processo di Amazon ECS	arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :task/<i>task-id</i> correnti Nuovo: arn:aws:ecs: <i>region</i>:<i>aws_account_id</i> :task/<i>cluster-name</i> /<i>task-id</i>

Lunghezza dell'ID risorsa

Un ID risorsa è formato da una combinazione univoca di lettere e numeri. I nuovi formati di ID delle risorse includono quelli più brevi IDs per le attività di Amazon ECS e le istanze di container. Il formato dell'ID risorsa precedente è lungo 36 caratteri. I nuovi IDs sono in un formato a 32 caratteri che non include trattini. Per informazioni sull'attivazione con il nuovo formato dell'ID risorsa, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

Il valore predefinito è `enabled`.

Solo le risorse avviate dopo l'attivazione riceveranno il nuovo formato dell'ARN e dell'ID risorsa. Tutte le risorse esistenti non sono interessate. Per eseguire la transizione di servizi e attività Amazon ECS ai nuovi formati di ARN e ID risorsa, è necessario creare nuovamente il servizio o il processo. Per eseguire la transizione di un'istanza di container al nuovo formato dell'ARN e dell'ID risorsa, l'istanza di container deve essere eliminata e ne deve essere avviata registrata una nuova nel cluster.

 Note

Le attività avviate da un servizio Amazon ECS possono ricevere il nuovo formato dell'ARN e dell'ID risorsa solo se il servizio è stato creato a partire dal 16 novembre 2018 e l'utente che ha creato il servizio ha fornito il consenso esplicito al nuovo formato per i processi.

Sequenza temporale formato ARN e ID risorsa

La sequenza temporale per i periodi di accettazione e rifiuto del nuovo formato del nome della risorsa Amazon (ARN) e dell'ID risorsa per le risorse Amazon ECS è terminata il 1° aprile 2021. Per impostazione predefinita, tutti gli account accettano il nuovo formato. Tutte le nuove risorse create ricevono il nuovo formato e non puoi più disattivarle.

Container Insights

Il 2 dicembre 2024, AWS ha rilasciato Container Insights con osservabilità migliorata per Amazon ECS. Questa versione supporta l'osservabilità migliorata per i cluster Amazon ECS utilizzando i tipi di lancio Amazon e EC2 Fargate. Dopo aver configurato Container Insights con una migliore osservabilità su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura dal livello del cluster fino al livello del contenitore nel tuo ambiente e visualizza i tuoi dati in dashboard che mostrano una varietà di metriche e dimensioni. Puoi quindi utilizzare questi out-of-the-box dashboard sulla console Container Insights per comprendere meglio lo stato e le prestazioni dei container e mitigare i problemi più rapidamente identificando le anomalie.

Ti consigliamo di utilizzare Container Insights con una migliore osservabilità anziché Container Insights perché offre una visibilità dettagliata nell'ambiente dei container, riducendo il tempo medio di risoluzione. Per ulteriori informazioni, consulta [Amazon ECS Container Insights con metriche di osservabilità avanzate](#) nella Guida per l'Amazon CloudWatch utente.

L'impostazione predefinita per l'`containerInsights` account è `disabled`

Container Insights con osservabilità migliorata

Usa il seguente comando per attivare Container Insights con una migliore osservabilità.

Imposta l'impostazione `containerInsights` dell'account su `enhanced`.

```
aws ecs put-account-setting --name containerInsights --value enhanced
```

Output di esempio

```
{
  "setting": {
    "name": "containerInsights",
    "value": "enhanced",
    "principalArn": "arn:aws:iam::123456789012:johndoe",
    "type": user
  }
}
```

Dopo aver impostato questa impostazione dell'account, tutti i nuovi cluster utilizzano automaticamente Container Insights con una migliore osservabilità. Usa il `update-cluster-settings` comando per aggiungere Container Insights con osservabilità migliorata a un cluster esistente o per aggiornare un cluster da Container Insights a Container Insights con osservabilità migliorata.

```
aws ecs update-cluster-settings --cluster cluster-name --settings
name=containerInsights,value=enhanced
```

Puoi anche utilizzare la console per configurare Container Insights con una migliore osservabilità. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#)

Container Insights

Quando si imposta l'impostazione dell'`containerInsightsaccount` su `enabled`, tutti i nuovi cluster hanno Container Insights abilitato per impostazione predefinita. È possibile modificare i cluster esistenti utilizzando `update-cluster-settings`

Per utilizzare Container Insights, imposta l'impostazione `containerInsights` dell'account su `enabled`. Usa il seguente comando per attivare Container Insights.

```
aws ecs put-account-setting --name containerInsights --value enabled
```

Output di esempio

```
{
  "setting": {
    "name": "containerInsights",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:johndoe",
  }
}
```

```
    "type": user
  }
}
```

Quando si imposta l'impostazione dell'account `containerInsights` su `enabled`, tutti i nuovi cluster hanno Container Insights abilitato per impostazione predefinita. Usa il `update-cluster-settings` comando per aggiungere Container Insights a un cluster esistente.

```
aws ecs update-cluster-settings --cluster cluster-name --settings
name=containerInsights,value=enabled
```

Puoi anche utilizzare la console per configurare Container Insights. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#)

AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)

Fargate supporta il Federal Information Processing Standard (FIPS-140) che specifica i requisiti di sicurezza previsti per i moduli crittografici che proteggono le informazioni sensibili. È l'attuale standard governativo degli Stati Uniti e del Canada ed è applicabile ai sistemi che devono essere conformi al Federal Information Security Management Act (FISMA) o al Federal Risk and Authorization Management Program (FedRAMP).

Il nome della risorsa è `fargateFIPSMODE`

Il valore predefinito è `disabled`.

Devi attivare la conformità al Federal Information Processing Standard (FIPS-140) su Fargate. Per ulteriori informazioni, consulta [the section called “AWS Fargate Conformità FIPS-140”](#).

Important

L'impostazione dell'account `fargateFIPSMODE` può essere modificata solo utilizzando l'API Amazon ECS o la AWS CLI. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

Esegui `put-account-setting-default` con l'opzione `fargateFIPSMODE` impostata su `enabled`. Per ulteriori informazioni, [put-account-setting-default](#) consulta la sezione Amazon Elastic Container Service API Reference.

- Puoi utilizzare il seguente comando per attivare la conformità FIPS-140.

```
aws ecs put-account-setting-default --name fargateFIPSMODE --value enabled
```

Output di esempio

```
{
  "setting": {
    "name": "fargateFIPSMODE",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root",
    "type": "user"
  }
}
```

Puoi eseguire il comando `list-account-settings` per visualizzare lo stato di conformità FIPS-140 corrente. Utilizza l'opzione `effective-settings` per visualizzare le impostazioni a livello di account.

```
aws ecs list-account-settings --effective-settings
```

Autorizzazione all'assegnazione di tag

Amazon ECS sta introducendo l'autorizzazione all'assegnazione di tag per la creazione di risorse. Gli utenti devono disporre delle autorizzazioni di etichettatura per le azioni che creano la risorsa, ad esempio, `ecsCreateCluster`. Quando crei una risorsa e specifichi i tag per quella risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vi siano le autorizzazioni per creare tag. Pertanto, devi concedere le autorizzazioni esplicite per utilizzare l'operazione `ecs:TagResource`. Per ulteriori informazioni, consulta [the section called “Assegnazione di tag alle risorse al momento della creazione”](#).

Per attivare l'autorizzazione all'assegnazione di tag, esegui il comando `put-account-setting-default` con l'opzione `tagResourceAuthorization` impostata su `enable`. Per ulteriori informazioni, [put-account-setting-default](#) consulta la sezione Amazon Elastic Container Service API Reference. Puoi eseguire il comando `list-account-settings` per visualizzare lo stato attuale dell'autorizzazione all'assegnazione di tag.

- Puoi utilizzare il seguente comando per abilitare l'autorizzazione all'etichettatura.

```
aws ecs put-account-setting-default --name tagResourceAuthorization --value on --  
region region
```

Output di esempio

```
{  
  "setting": {  
    "name": "tagResourceAuthorization",  
    "value": "on",  
    "principalArn": "arn:aws:iam::123456789012:root",  
    "type": user  
  }  
}
```

Dopo aver abilitato l'autorizzazione all'etichettatura, è necessario configurare le autorizzazioni appropriate per consentire agli utenti di taggare le risorse al momento della creazione. Per ulteriori informazioni, consulta [the section called “Assegnazione di tag alle risorse al momento della creazione”](#).

Puoi eseguire il comando `list-account-settings` per visualizzare lo stato attuale dell'autorizzazione all'assegnazione di tag. Utilizza l'opzione `effective-settings` per visualizzare le impostazioni a livello di account.

```
aws ecs list-account-settings --effective-settings
```

Cronologia dell'autorizzazione all'assegnazione di tag

Puoi confermare se l'autorizzazione all'assegnazione di tag è attiva eseguendo il comando `list-account-settings` per visualizzare il valore di `tagResourceAuthorization`. Quando il valore è `on`, indica che l'autorizzazione all'assegnazione di tag è in uso. Per ulteriori informazioni, [list-account-settings](#) consulta la sezione Amazon Elastic Container Service API Reference.

Di seguito sono riportate le date importanti correlate all'autorizzazione all'assegnazione di tag.

- 18 aprile 2023: introduzione dell'autorizzazione all'assegnazione di tag. Tutti gli account nuovi ed esistenti devono aderire per utilizzare la funzionalità. Puoi scegliere di iniziare a utilizzare l'autorizzazione all'etichettatura. Effettuando l'adesione, devi concedere le autorizzazioni appropriate.

- 9 febbraio 2024 - 6 marzo 2024: per tutti i nuovi account e per gli account esistenti non interessati è attivata l'autorizzazione all'etichettatura per impostazione predefinita. Puoi abilitare o disabilitare l'impostazione dell'`tagResourceAuthorization` account per verificare la tua politica IAM.

AWS ha notificato gli account interessati.

Per disabilitare la funzionalità, esegui `put-account-setting-default` con l'`tagResourceAuthorization` opzione impostata su `off`.

- 7 marzo 2024: se hai abilitato l'autorizzazione all'etichettatura, non puoi più disabilitare l'impostazione dell'account.

Ti consigliamo di completare i test delle policy IAM prima di questa data.

- 29 marzo 2024: tutti gli account utilizzano l'autorizzazione all'etichettatura. L'impostazione a livello di account non sarà più disponibile nella console Amazon ECS o. AWS CLI

AWS Fargate tempo di attesa per il ritiro dell'attività

AWS invia notifiche quando le attività di Fargate sono in esecuzione su una revisione della versione della piattaforma contrassegnata come ritirata. Per ulteriori informazioni, consulta [Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS](#).

AWS è responsabile dell'applicazione di patch e della manutenzione dell'infrastruttura sottostante per AWS Fargate. Quando si determina che è necessario un aggiornamento della sicurezza o dell'infrastruttura per un'attività Amazon ECS ospitata su Fargate, le attività devono essere interrotte e avviate nuove attività per sostituirle. Puoi configurare il periodo di attesa prima che le attività vengano ritirate per l'applicazione di patch. È possibile ritirare l'attività immediatamente, attendere 7 giorni di calendario o attendere 14 giorni di calendario.

Questa impostazione si applica a livello di account.

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (0). Quando hai bisogno di un maggiore controllo, ad esempio, quando un'attività può essere interrotta solo durante una determinata finestra, configura l'opzione 7 (7) o 14 giorni (14).

Consigliamo di scegliere un periodo di attesa più breve per ricevere prima le revisioni delle versioni più recenti della piattaforma.

Configura il periodo di attesa eseguendo `put-account-setting-default` o `put-account-setting` come utente `root` o utente amministrativo. Utilizza l'opzione `fargateTaskRetirementWaitPeriod` per il `name` e l'opzione `value` impostata su uno dei seguenti valori:

- 0- AWS invia la notifica e inizia immediatamente a ritirare le attività interessate.
- 7- AWS invia la notifica e attende 7 giorni di calendario prima di iniziare a ritirare le attività interessate.
- 14- AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

L'impostazione di default è 7 giorni.

Per ulteriori informazioni, consulta [put-account-setting-default](#) e consulta il riferimento [put-account-setting](#) all'API di Amazon Elastic Container Service.

Puoi eseguire il comando seguente per impostare il periodo di attesa su 14 giorni.

```
aws ecs put-account-setting-default --name fargateTaskRetirementWaitPeriod --value 14
```

Output di esempio

```
{
  "setting": {
    "name": "fargateTaskRetirementWaitPeriod",
    "value": "14",
    "principalArn": "arn:aws:iam::123456789012:root",
    "type": "user"
  }
}
```

Puoi eseguire il comando `list-account-settings` per visualizzare il tempo di attesa corrente per il ritiro delle attività di Fargate. Utilizza l'opzione `effective-settings`.

```
aws ecs list-account-settings --effective-settings
```

Aumenta le interfacce di rete delle istanze di container Linux

Ogni attività Amazon ECS che utilizza la modalità di `awsvpc` rete riceve la propria interfaccia di rete elastica (ENI), che è collegata all'istanza del contenitore che la ospita. Esiste un limite predefinito al numero di interfacce di rete che possono essere collegate a un' EC2 istanza Amazon e l'interfaccia di rete principale conta come una. Ad esempio, per impostazione predefinita, a un' `c5.large` istanza possono essere ENIs collegate fino a tre istanze. L'interfaccia di rete principale per l'istanza conta come una sola, quindi è possibile collegarne altre due ENIs all'istanza. Perché ogni operazione che utilizza la modalità `awsvpc` di rete richiede un ENI, in genere è possibile eseguire solo due di queste attività su questo tipo di istanza.

Amazon ECS supporta il lancio di istanze di container con maggiori ENI densità utilizzando i tipi di EC2 istanze Amazon supportati. Quando utilizzi questi tipi di istanze e attivi l'impostazione dell'`awsvpcTrunking` account, ne ENIs sono disponibili altre sulle istanze di container appena lanciate. Questa configurazione consente di posizionare più attività su ciascuna istanza di container.

Ad esempio, un' `c5.large` istanza con `awsvpcTrunking` ha un valore aumentato ENI limite di dodici. L'istanza di container avrà un'interfaccia di rete primaria e Amazon ECS crea e collega un'interfaccia di rete "trunk" all'istanza di container. Pertanto, questa configurazione consente di avviare dieci attività sull'istanza di container anziché le due attività correnti.

Monitoraggio del runtime (GuardDuty integrazione con Amazon)

Runtime Monitoring è un servizio intelligente di rilevamento delle minacce che protegge i carichi di lavoro in esecuzione su Fargate EC2 e sulle istanze di container AWS monitorando continuamente i log e l'attività di rete per identificare comportamenti dannosi o non autorizzati.

Il `guardDutyActivate` parametro è di sola lettura in Amazon ECS e indica se il Runtime Monitoring è attivato o disattivato dall'amministratore della sicurezza nel tuo account Amazon ECS. GuardDuty controlla questa impostazione dell'account per tuo conto. Per ulteriori informazioni, consulta [Proteggere i carichi di lavoro Amazon ECS con Runtime Monitoring](#).

Puoi eseguire l'operazione `list-account-settings` per visualizzare l'impostazione di GuardDuty integrazione corrente.

```
aws ecs list-account-settings
```

Output di esempio

```
{
  "setting": {
    "name": "guardDutyActivate",
    "value": "on",
    "principalArn": "arn:aws:iam::123456789012:doej",
    "type": "aws-managed"
  }
}
```

VPC a doppio stack IPv6

Amazon ECS supporta la fornitura di attività con un IPv6 indirizzo oltre all' IPv4 indirizzo privato principale.

Affinché le attività ricevano un IPv6 indirizzo, l'attività deve utilizzare la modalità di awsvpc rete, deve essere avviata in un VPC configurato per la modalità dual-stack e l'impostazione dell'`dualStackIPv6account` deve essere abilitata. Per ulteriori informazioni su altri requisiti, consulta [Utilizzo di un VPC in modalità dual-stack](#) per il tipo di EC2 lancio e [Utilizzo di un VPC in modalità dual-stack](#) per il tipo di lancio Fargate.

Important

L'impostazione dell'account `dualStackIPv6` può essere modificata solo utilizzando l'API Amazon ECS o la AWS CLI. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

Se avevi un'attività in esecuzione utilizzando la modalità di awsvpc rete in una sottorete IPv6 abilitata tra le date del 1° ottobre 2020 e il 2 novembre 2020, l'impostazione predefinita dell'`dualStackIPv6account` nella regione in cui l'attività era in esecuzione è `disabled`. Se tale condizione non viene soddisfatta, l'impostazione `dualStackIPv6` di default nella regione è `enabled`.

Il valore predefinito è `disabled`.

Modalità predefinita del driver di registro

Amazon ECS supporta l'impostazione di una modalità di consegna predefinita dei messaggi di registro da un contenitore al driver di registro scelto. La modalità di consegna influisce sulla stabilità dell'applicazione quando il flusso di log dal contenitore al driver di registro viene interrotto.

L'`defaultLogDriverMode` impostazione supporta due valori: `blocking` e `non-blocking`. Per ulteriori informazioni su queste modalità di distribuzione, consulta [LogConfiguration](#) Amazon Elastic Container Service API Reference.

Se non specifichi una modalità di consegna nelle definizioni del contenitore `logConfiguration`, la modalità specificata utilizzando questa impostazione dell'account verrà utilizzata come predefinita.

Important

L'impostazione dell'account `defaultLogDriverMode` può essere modificata solo utilizzando l'API Amazon ECS o la AWS CLI. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

La modalità di consegna predefinita è `blocking`.

Note

Il 25 giugno 2025, Amazon ECS cambierà la modalità predefinita del driver di registro da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto alla registrazione. Per continuare a utilizzare la `blocking` modalità dopo questa modifica, esegui una delle seguenti operazioni:

- Imposta l'opzione `mode` nella definizione del contenitore `logConfiguration` come `blocking`.
- Imposta l'impostazione `defaultLogDriverMode` dell'account su `blocking`.

Per impostare una modalità di driver di registro predefinita su `non-blocking`, è possibile eseguire il comando seguente.

```
aws ecs put-account-setting-default --name defaultLogDriverMode --value "non-blocking"
```

Visualizzazione delle impostazioni dell'account Amazon ECS tramite la console

Visualizza le impostazioni del tuo account nella console per vedere a quali funzionalità hai accesso.

 Important

Le impostazioni dell'account `dualStackIPv6`, `fargateFIPSMODE` e `fargateTaskRetirementWaitPeriod` possono essere visualizzate o modificate solo utilizzando la AWS CLI.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella barra di navigazione nella parte superiore, seleziona la Regione per la quale visualizzare le impostazioni dell'account.
3. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).

Modifica delle impostazioni dell'account Amazon ECS

Modifica le impostazioni del tuo account per accedere alle funzionalità di Amazon ECS.

Il `guardDutyActivate` parametro è di sola lettura in Amazon ECS e indica se il Runtime Monitoring è attivato o disattivato dall'amministratore della sicurezza nel tuo account Amazon ECS. GuardDuty controlla questa impostazione dell'account per tuo conto. Per ulteriori informazioni, consulta [Proteggere i carichi di lavoro Amazon ECS con Runtime Monitoring](#).

 Important

Le impostazioni dell'account `dualStackIPv6`, `fargateFIPSMODE` e `fargateTaskRetirementWaitPeriod` possono essere visualizzate o modificate solo utilizzando la AWS CLI.

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella barra di navigazione nella parte superiore, seleziona la Regione per la quale visualizzare le impostazioni dell'account.
3. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).
4. Scegli Aggiorna.
5. Per aumentare o diminuire il numero di attività che è possibile eseguire nella modalità di rete `awsipc` per ciascuna EC2 istanza, in `AWSVPC Trunking`, seleziona `Trunking`. `AWSVPC`

6. Per utilizzare o interrompere l'utilizzo predefinito di CloudWatch Container Insights per i cluster, in Osservabilità di CloudWatch Container Insights, scegli una delle seguenti opzioni:
 - Per utilizzare Container Insights con osservabilità migliorata, scegli Container Insights con osservabilità migliorata.
 - Per utilizzare Container Insights, scegli Container Insights.
 - Per smettere di usare Container Insights, scegli Disattivato.
7. Per abilitare o disabilitare l'autorizzazione all'etichettatura delle risorse, in Autorizzazione di etichettatura delle risorse, seleziona o deseleziona Autorizzazione di etichettatura delle risorse.
8. Scegli Save changes (Salva modifiche).
9. Nella schermata di conferma, scegliere Conferma per salvare la selezione.

Passaggi successivi

Se hai attivato Container Insights con osservabilità migliorata o Container Insights, puoi facoltativamente aggiornare i cluster esistenti per utilizzare la funzionalità. Per ulteriori informazioni, consulta [Aggiornamento di un cluster Amazon ECS](#).

Ripristino alle impostazioni predefinite dell'account Amazon ECS

Puoi utilizzare il AWS Management Console per ripristinare le impostazioni predefinite del tuo account Amazon ECS.

L'opzione Revert to account default (Ripristina valori predefiniti account) è disponibile solo quando le impostazioni dell'account non sono più le impostazioni predefinite.

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella barra di navigazione nella parte superiore, seleziona la Regione per la quale visualizzare le impostazioni dell'account.
3. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).
4. Scegli Aggiorna.
5. Scegli Revert to account default (Ripristina valori predefiniti account).
6. Nella schermata di conferma, scegli Conferma per salvare la selezione.

Gestione delle impostazioni dell'account Amazon ECS tramite AWS CLI

Puoi gestire le impostazioni del tuo account utilizzando l'API Amazon ECS AWS CLI oppure SDKs. Le `dualStackIPv6` impostazioni `fargateFIPSPMode` e le impostazioni `fargateTaskRetirementWaitPeriod` dell'account possono essere visualizzate o modificate solo utilizzando tali strumenti.

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

Per informazioni sulle operazioni API disponibili per le definizioni delle attività, consulta [Operazioni delle impostazioni dell'account](#) nella Documentazione di riferimento dell'API di Amazon Elastic Container Service.

Utilizza uno dei comandi seguenti per modificare le impostazioni dell'account di default per tutti gli utenti o i ruoli sull'account. Queste modifiche si applicano all'intero AWS account a meno che un utente o un ruolo non sostituisca esplicitamente queste impostazioni.

- [put-account-setting-default](#) (AWS CLI)

```
aws ecs put-account-setting-default --name serviceLongArnFormat --value enabled --  
region us-east-2
```

Puoi utilizzare questo comando anche per modificare altre impostazioni dell'account. A questo scopo, sostituisci il parametro name con l'impostazione dell'account corrispondente.

- [Write-Setting \(\) ECSAccount](#) AWS Tools for Windows PowerShell

```
Write-ECSAccountSettingDefault -Name serviceLongArnFormat -Value enabled -Region us-  
east-1 -Force
```

Per modificare le impostazioni dell'account per l'account utente (AWS CLI)

Utilizza uno dei comandi seguenti per modificare le impostazioni dell'account per l'utente . Se utilizzi questi comandi come utente root, le modifiche si applicheranno all'intero account AWS , a meno che un utente o ruolo non sovrascriva in modo esplicito e autonomo tali impostazioni.

- [put-account-setting](#) (AWS CLI)

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled --region us-east-1
```

Puoi utilizzare questo comando anche per modificare altre impostazioni dell'account. A questo scopo, sostituisci il parametro name con l'impostazione dell'account corrispondente.

- [ECSAccountImpostazione di scrittura](#) (AWS Tools for Windows PowerShell)

```
Write-ECSAccountSetting -Name serviceLongArnFormat -Value enabled -Force
```

Modifica delle impostazioni dell'account per un utente o un ruolo specifico (AWS CLI)

Utilizza uno dei seguenti comandi e specifica l'ARN di un utente, ruolo o utente root nella richiesta per modificare le impostazioni dell'account per un utente o ruolo specifico.

- [put-account-setting](#) (AWS CLI)

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled --principal-arn arn:aws:iam::aws_account_id:user/principalName --region us-east-1
```

Puoi utilizzare questo comando anche per modificare altre impostazioni dell'account. A questo scopo, sostituisci il parametro name con l'impostazione dell'account corrispondente.

- [ECSAccountImpostazione di scrittura](#) (AWS Tools for Windows PowerShell)

```
Write-ECSAccountSetting -Name serviceLongArnFormat -Value enabled -PrincipalArn arn:aws:iam::aws_account_id:user/principalName -Region us-east-1 -Force
```

Ruoli IAM per Amazon ECS

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità utilizzate. Utilizza la tabella seguente per determinare quali ruoli IAM sono necessari per Amazon ECS.

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri AWS servizi per tuo conto.	La tua attività è ospitata su AWS Fargateo su istanze esterne e: • estrae un'immagine del contenitore da un repository privato Amazon ECR. • estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività. • invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. <code>awslogs</code>	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		La tua attività è ospitata su una AWS Fargate EC2 delle istanze Amazon e: • utilizza l'autenticazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di Secrets Manager o i parametri AWS Systems Manager Parameter Store.	
Ruolo del processo	Questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi.	L'applicazione accede ad altri AWS servizi, come Amazon S3.	Ruolo IAM dell'attività Amazon ECS
Ruolo dell'istanza di container	Questo ruolo consente alle EC2 istanze o alle istanze esterne di registrarsi nel cluster.	La tua attività è ospitata su EC2 istanze Amazon o su un'istanza esterna.	Ruolo IAM delle istanze di container Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di Amazon ECS Anywhere	Questo ruolo consente l'accesso alle istanze esterne. AWS APIs	L'attività è ospitata su istanze esterne.	Ruolo IAM di Amazon ECS Anywhere
Ruolo di Amazon ECS CodeDeploy	Questo ruolo consente CodeDeploy di aggiornare i tuoi servizi.	Si utilizza il tipo di distribuzione CodeDeploy blu/verde per distribuire i servizi.	Ruolo CodeDeploy IAM di Amazon ECS
Ruolo di Amazon ECS EventBridge	Questo ruolo consente EventBridge di aggiornare i tuoi servizi.	Utilizzi le EventBridge regole e gli obiettivi per pianificare le tue attività.	Ruolo EventBridge IAM di Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo dell'infrastruttura Amazon ECS	Questo ruolo consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività. • Desideri utilizzare e Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect. • Vuoi creare gruppi target VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS

Definizioni dei processi di Amazon ECS

Una definizione di attività è uno schema dell'applicazione. Si tratta di un file di testo in formato JSON che descrive i parametri e uno o più container che formano l'applicazione.

Alcuni dei parametri che puoi specificare in una definizione di attività includono:

- Il tipo di avvio da utilizzare, che determina l'infrastruttura in cui vengono ospitate le attività
- Immagine Docker da utilizzare con ogni container nel processo
- La quantità di CPU e di memoria da utilizzare con ogni processo o container all'interno di un processo
- I requisiti di memoria e CPU
- Il sistema operativo del contenitore su cui viene eseguita l'attività
- La modalità di rete Docker da utilizzare per i container nel tuo processo
- La configurazione di registrazione da utilizzare per i tuoi processi
- La possibilità che l'attività continui a essere eseguita in caso di interruzione o esito negativo del container
- Il comando che il container deve eseguire all'avvio
- Tutti i volumi di dati utilizzati con il container nell'attività
- Il ruolo IAM utilizzato dalle attività

Per un elenco completo dei parametri per la definizione di attività, consulta [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#).

Dopo aver creato una definizione di attività, è possibile eseguirla come attività o servizio.

- Si definisce attività la creazione dell'istanza relativa a una definizione di attività all'interno di un cluster. Dopo aver creato una definizione di attività per la tua applicazione all'interno di Amazon ECS, puoi specificare il numero di processi che saranno eseguiti sul tuo cluster.
- un servizio Amazon ECS esegue e mantiene simultaneamente il numero desiderato di attività in un cluster Amazon ECS. Il funzionamento è che, se uno dei processi non riesce o si interrompe per qualsiasi motivo, il pianificatore del servizio Amazon ECS lancia un'altra istanza in base alla definizione di attività. Lo fa per sostituirlo e quindi mantenere il numero desiderato di processi nel servizio.

Argomenti

- [Stati di definizione delle attività di Amazon ECS](#)
- [Progetta la tua applicazione per Amazon ECS](#)
- [Creazione di una definizione di attività Amazon ECS utilizzando la console](#)
- [Aggiornamento di una definizione di attività Amazon ECS tramite la console](#)
- [Annullamento della registrazione di una revisione della definizione di attività di Amazon ECS tramite la console](#)
- [Eliminazione di una revisione della definizione di attività di Amazon ECS tramite la console](#)
- [Casi d'uso per la definizione delle attività di Amazon ECS](#)
- [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#)
- [Parametri di definizione delle attività di Amazon ECS per il tipo di EC2 avvio](#)
- [Modello di definizione delle attività di Amazon ECS](#)
- [Esempi di definizioni di attività Amazon ECS](#)

Stati di definizione delle attività di Amazon ECS

Una definizione di attività cambia lo stato quando la crei, annulli la registrazione o la elimini. È possibile visualizzare lo stato di definizione dell'attività nella console o utilizzando.

`DescribeTaskDefinition`

Di seguito sono riportati i possibili stati di una definizione di attività.

ACTIVE

Una definizione di attività diventa ACTIVE una volta che è stata registrata con Amazon ECS. È possibile utilizzare le definizioni di attività nello stato ACTIVE per eseguire attività o creare servizi.

INACTIVE

Una definizione di attività passa dallo stato ACTIVE allo stato INACTIVE quando si annulla la registrazione di una definizione di attività. È possibile recuperare una definizione di attività INACTIVE attraverso la chiamata `DescribeTaskDefinition`. Non è possibile eseguire nuove attività o creare nuovi servizi con una definizione di attività nello stato INACTIVE. I servizi e le attività esistenti non subiscono alcun impatto.

DELETE_IN_PROGRESS

Una definizione di attività passa dallo stato `INACTIVE` allo stato `DELETE_IN_PROGRESS` dopo la richiesta di eliminazione della definizione di attività. Dopo che la definizione dell'attività ha raggiunto lo stato `DELETE_IN_PROGRESS`, Amazon ECS verifica periodicamente che la definizione dell'attività di destinazione non sia referenziata da alcuna attività o implementazione attiva e successivamente elimina in via definitiva la definizione dell'attività. Non è possibile eseguire nuove attività o creare nuovi servizi con una definizione di attività nello stato `DELETE_IN_PROGRESS`. Una definizione di attività può essere inviata per l'eliminazione in qualsiasi momento senza influire sulle attività e sui servizi esistenti.

Le definizioni di attività nello stato `DELETE_IN_PROGRESS` sono visibili nella console ed è possibile recuperarle effettuando una chiamata `DescribeTaskDefinition`.

Quando elimini tutte le revisioni delle definizioni di attività nello stato `INACTIVE`, il nome della definizione di attività non viene visualizzato nella console e non viene restituito nell'API. Se `DELETE_IN_PROGRESS` lo stato di revisione della definizione dell'attività è attivo, il nome della definizione dell'attività viene visualizzato nella console e restituito nell'API. Il nome della definizione di attività viene mantenuto da Amazon ECS e la revisione viene incrementata durante la prossima creazione di una definizione di attività con tale nome.

Se lo utilizzi AWS Config per gestire le definizioni delle attività, ti AWS Config addebita il costo di tutte le registrazioni delle definizioni delle attività. Viene addebitato solo l'annullamento della registrazione della definizione di attività `ACTIVE` più recente. L'eliminazione di una definizione di attività non prevede alcun costo. Per ulteriori informazioni sui prezzi, consulta [Prezzi di AWS Config](#).

Risorse Amazon ECS in grado di bloccare un'eliminazione

Una richiesta di eliminazione della definizione di attività non viene completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione dell'attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività autonome di Amazon ECS: la definizione dell'attività è necessaria affinché l'attività rimanga integra.
- Attività del servizio Amazon ECS: la definizione dell'attività è necessaria affinché l'attività rimanga integra.
- Distribuzioni di servizi Amazon ECS e set di attività: la definizione dell'attività è richiesta quando viene avviato un evento di scalabilità per una distribuzione o un set di attività di Amazon ECS.

Se la definizione dell'attività rimane invariata `DELETE_IN_PROGRESS`, puoi utilizzare la console o identificare e quindi interrompere AWS CLI le risorse che bloccano l'eliminazione della definizione dell'attività.

Eliminazione della definizione di attività dopo la rimozione della risorsa bloccata

Le seguenti regole si applicano dopo aver rimosso le risorse che bloccano l'eliminazione della definizione di attività:

- Attività Amazon ECS: dopo l'interruzione dell'attività, l'eliminazione della definizione di attività può richiedere fino a 1 ora.
- Distribuzioni di servizi Amazon ECS e set di attività: il completamento dell'eliminazione delle definizioni delle attività può richiedere fino a 24 ore dopo l'eliminazione della distribuzione o del set di attività.

Progetta la tua applicazione per Amazon ECS

L'applicazione viene progettata creando una definizione di attività per l'applicazione. La definizione dell'attività contiene i parametri che definiscono le informazioni sull'applicazione, tra cui:

- Il tipo di avvio da utilizzare, che determina l'infrastruttura su cui sono ospitate le attività.

Quando usi il tipo di EC2 avvio, scegli anche il tipo di istanza. Per alcuni tipi di istanza, come la GPU, è necessario impostare parametri aggiuntivi. Per ulteriori informazioni, consulta [Casi d'uso per la definizione delle attività di Amazon ECS](#).

- L'immagine del contenitore, che contiene il codice dell'applicazione e tutte le dipendenze necessarie per l'esecuzione del codice dell'applicazione.
- La modalità di rete da utilizzare per i contenitori coinvolti nell'attività

La modalità di rete determina il modo in cui l'attività comunica sulla rete.

Per le attività eseguite su EC2 istanza, sono disponibili diverse opzioni, ma è consigliabile utilizzare la modalità di `awsvpc` rete. La modalità `awsvpc` di rete semplifica il networking in container, poiché avete un maggiore controllo sul modo in cui le applicazioni comunicano tra loro e con gli altri servizi all'interno dell'azienda VPCs.

Per le attività eseguite su Fargate, è possibile utilizzare solo la modalità di `awsvpc` rete.

- La configurazione di registrazione da utilizzare per le tue attività.

- Qualsiasi volume di dati utilizzato con i contenitori dell'attività.

Per un elenco completo dei parametri per la definizione di attività, consulta [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#).

Utilizza le linee guida seguenti per creare le definizioni di attività:

- Utilizza ogni famiglia di definizioni delle attività per un solo scopo aziendale.

Se raggruppi più tipi di container delle applicazioni nella stessa definizione di attività, non puoi dimensionare tali container in modo indipendente. Ad esempio, è improbabile che un sito Web e un'API richiedano l'impiego della scalabilità orizzontale alla stessa velocità. Con l'aumento del traffico, sarà necessario un numero diverso di container Web rispetto ai container API. Se questi due container vengono implementati nella stessa definizione di attività, ogni attività esegue lo stesso numero di container Web e container API.

- Associa ogni versione dell'applicazione a una revisione della definizione di attività all'interno di una famiglia di definizioni delle attività.

All'interno di una famiglia di definizioni delle attività, considera ogni revisione della definizione di attività come uno snapshot point-in-time delle impostazioni per una particolare immagine di container. È simile al modo in cui il container può essere considerato uno snapshot di tutti gli elementi necessari per eseguire una particolare versione del codice dell'applicazione.

Assicurati che esista una one-to-one mappatura tra una versione del codice dell'applicazione, un tag di immagine del contenitore e una revisione della definizione dell'attività. Un tipico processo di rilascio prevede un commit git che viene trasformato in un'immagine di container con il tag SHA del commit git. Quindi, il tag dell'immagine di container riceve la propria revisione della definizione di attività di Amazon ECS. Infine, il servizio Amazon ECS viene aggiornato per ordinare l'implementazione della nuova revisione della definizione di attività.

- Utilizza ruoli IAM diversi per ogni famiglia di definizione delle attività.

Definisci ogni definizione delle attività con il proprio ruolo IAM. Questa raccomandazione dovrebbe essere seguita congiuntamente al consiglio di fornire a ogni componente aziendale la propria famiglia di definizione delle attività. Implementando entrambe queste best practice, puoi limitare l'accesso di ciascun servizio alle risorse del tuo AWS account. Ad esempio, puoi concedere al servizio di autenticazione l'accesso per connettersi al database delle password. Allo stesso tempo, puoi garantire l'accesso alle informazioni di pagamento con carta di credito soltanto al servizio.

Best practice per le immagini dei container Amazon ECS

Un'immagine del container è un insieme di istruzioni su come creare il container. Un'immagine del container contiene il codice dell'applicazione e tutte le dipendenze necessarie per l'esecuzione di tale codice. Le dipendenze dell'applicazione includono i pacchetti di codice sorgente su cui si basa il codice dell'applicazione, un runtime del linguaggio per i linguaggi interpretati e i pacchetti binari su cui si basa il codice collegato dinamicamente.

Utilizza le linee guida seguenti per progettare e creare le immagini dei container:

- Rendi complete le immagini del container, archiviando tutte le dipendenze dell'applicazione come file statici all'interno dell'immagine del container.

Se modifichi qualcosa nell'immagine del contenitore, crea una nuova immagine del contenitore con le modifiche.

- Esegui un singolo processo di applicazione all'interno di un container.

La durata del container è pari a quella del processo di applicazione. Amazon ECS sostituisce i processi bloccati e determina dove avviare il processo di sostituzione. Un'immagine completa rende l'implementazione complessiva più resiliente.

- Fai in modo che la tua applicazione sia gestibile. SIGTERM

Quando Amazon ECS interrompe un'attività, invia innanzitutto un segnale SIGTERM all'attività per notificare all'applicazione che deve essere completata e chiusa. Amazon ECS invia quindi un SIGKILL messaggio. Quando le applicazioni lo ignorano SIGTERM, il servizio Amazon ECS deve attendere l'invio del SIGKILL segnale per terminare il processo.

Devi identificare il tempo impiegato dall'applicazione per completare il suo lavoro e assicurarti che le applicazioni gestiscano il segnale. SIGTERM La gestione del segnale dell'applicazione deve impedire all'applicazione di eseguire nuovi lavori e completare il lavoro in corso, oppure salvare il lavoro incompiuto in un archivio esterno all'attività quando il completamento del lavoro richiede troppo tempo.

- Configura le applicazioni containerizzate per la scrittura di log su `stdout` e `stderr`.

Il disaccoppiamento della gestione dei log dal codice dell'applicazione offre la flessibilità necessaria per adattare la gestione dei log a livello di infrastruttura. Un esempio di ciò è la modifica del sistema di registrazione. Invece di modificare i servizi e creare e distribuire una nuova immagine del contenitore, puoi modificare le impostazioni.

- Usa i tag per controllare le versioni delle immagini dei container.

Le immagini dei container vengono archiviate in un registro dei container. Ogni immagine all'interno del registro è identificata da un tag. È presente un tag denominato `latest`. Questo tag funge da puntatore verso la versione più recente dell'immagine del container dell'applicazione, simile al comando `HEAD` in un repository git. Ti consigliamo di utilizzare il tag `latest` solo a scopo di test. Come best practice, assegna alle immagini del container un tag univoco per ogni build. Ti consigliamo di assegnare i tag alle immagini tramite SHA git per il commit git usato per creare l'immagine.

Non è necessario creare un'immagine di container per ogni commit. Tuttavia, ti consigliamo di creare una nuova immagine del container ogni volta che rilasci un particolare commit di codice nell'ambiente di produzione. Ti consigliamo inoltre di assegnare all'immagine un tag che corrisponda al commit git del codice presente nell'immagine. Se hai assegnato all'immagine un commit git, puoi trovare più rapidamente la versione del codice in esecuzione sull'immagine.

Ti consigliamo inoltre di attivare i tag di immagine immutabili in Amazon Elastic Container Registry. Con questa impostazione, non puoi modificare l'immagine del container a cui punta un tag. Amazon ECR impone invece che una nuova immagine debba essere caricata su un nuovo tag. Per ulteriori informazioni, consulta [Mutabilità del tag immagine](#) nella Guida per l'utente di Amazon ECR.

Quando si progetta l'applicazione in modo che venga eseguita AWS Fargate, è necessario decidere se distribuire più contenitori nella stessa definizione di attività e distribuire contenitori separatamente in più definizioni di attività. Quando sono necessarie le seguenti condizioni, si consiglia di implementare più container nella stessa definizione di attività:

- I container condividono un ciclo di vita comune (ovvero vengono avviati e terminati contemporaneamente).
- I container devono essere eseguiti sullo stesso host sottostante (ovvero, un container che fa riferimento all'altro su una porta localhost).
- I contenitori condividono le risorse.
- I tuoi container condividono volumi di dati.

Se queste condizioni non sono richieste, si consiglia di implementare i container separatamente in più definizioni di attività. Ciò consente di scalare, effettuare il provisioning e il deprovisioning dei contenitori separatamente.

Le migliori pratiche per le dimensioni delle attività di Amazon ECS

Le dimensioni dei contenitori e delle attività sono entrambe essenziali per la scalabilità e la pianificazione della capacità. In Amazon ECS, CPU e memoria sono due parametri di risorse utilizzati per determinare la capacità. La CPU viene misurata in unità di 1/1024 di una vCPU completa (dove 1024 unità equivalgono a 1 vCPU intera). La memoria viene misurata in megabyte. Nella definizione dell'attività, è possibile configurare le prenotazioni e i limiti delle risorse.

Quando configuri una prenotazione, stai impostando la quantità minima di risorse richiesta da un'attività. L'attività riceve almeno la quantità di risorse richieste. L'applicazione potrebbe essere in grado di utilizzare più CPU o memoria rispetto alla prenotazione dichiarata. Tuttavia, ciò è soggetto ai limiti che hai anche dichiarato. L'utilizzo di un importo superiore all'importo della prenotazione è noto come *bursting*. In Amazon ECS, le prenotazioni sono garantite. Ad esempio, se utilizzi EC2 istanze Amazon per fornire capacità, Amazon ECS non inserisce un'attività su un'istanza in cui la prenotazione non può essere soddisfatta.

Un limite è la quantità massima di unità CPU o memoria che il contenitore o l'attività può utilizzare. Qualsiasi tentativo di utilizzare più CPU oltre questo limite comporta una limitazione. Qualsiasi tentativo di utilizzare più memoria comporta l'interruzione del contenitore.

La scelta di questi valori può essere difficile. Questo perché i valori più adatti all'applicazione dipendono in larga misura dai requisiti di risorse dell'applicazione. Il test di carico dell'applicazione è la chiave per una corretta pianificazione del fabbisogno di risorse e una migliore comprensione dei requisiti dell'applicazione.

Applicazioni stateless

Per le applicazioni stateless con scalabilità orizzontale, ad esempio un'applicazione con sistema di bilanciamento del carico, consigliamo innanzitutto di determinare la quantità di memoria che l'applicazione consuma per soddisfare le richieste. A tale scopo, è possibile utilizzare strumenti tradizionali come *ps* o soluzioni di monitoraggio come *top* Container Insights. CloudWatch

Nel determinare una prenotazione della CPU, considerate come volete scalare l'applicazione per soddisfare i requisiti aziendali. È possibile utilizzare riserve di CPU più piccole, ad esempio 256 unità CPU (o 1/4 vCPU), per scalare orizzontalmente in modo da ridurre al minimo i costi. Tuttavia, potrebbero non scalare abbastanza velocemente per soddisfare i picchi significativi della domanda. È possibile utilizzare prenotazioni di CPU più grandi per scalare più rapidamente e quindi soddisfare più rapidamente i picchi di domanda. Tuttavia, le prenotazioni di CPU più grandi sono più costose.

Altre applicazioni

Per le applicazioni che non sono scalabili orizzontalmente, come Singleton Worker o server di database, la capacità e i costi disponibili rappresentano le considerazioni più importanti. È consigliabile scegliere la quantità di memoria e CPU in base a ciò che i test di carico indicano che è necessario servire il traffico per raggiungere l'obiettivo del livello di servizio. Amazon ECS garantisce che l'applicazione sia collocata su un host con una capacità adeguata.

Opzioni di task networking di Amazon ECS per il tipo di EC2 avvio

Il comportamento di rete delle attività di Amazon ECS ospitate su EC2 istanze Amazon dipende dalla modalità di rete definita nella definizione dell'attività. Si consiglia di utilizzare la modalità di rete `awsvpc` a meno che non sia necessario utilizzare una modalità di rete diversa.

Di seguito sono riportate le modalità di rete disponibili.

Modalità di rete	Contenitori Linux su EC2	Contenitori Windows attivi EC2	Descrizione
<code>awsvpc</code>	Sì	Sì	All'attività viene assegnata una propria elastic network interface (ENI) e un IPv4 indirizzo privato principale. Ciò conferisce all'attività le stesse proprietà di rete delle EC2 istanze Amazon.
<code>bridge</code>	Sì	No	L'attività utilizza la rete virtuale integrata di Docker su Linux, che viene eseguita all'interno di ogni EC2 istanza Amazon che ospita l'attività. La rete virtuale integrata su Linux utilizza il driver di rete <code>bridge</code> Docker. Questa è la modalità di rete predefinita su Linux se non viene specificata una modalità di rete nella definizione di attività.
<code>host</code>	Sì	No	L'attività utilizza la rete dell'host che aggira la rete virtuale integrata di Docker mappando le porte dei container direttamente all'ENI dell'

Modalità di rete	Contenitori Linux su EC2	Contenitori Windows attivi EC2	Descrizione
			EC2 istanza Amazon che ospita l'attività. Le mappature dinamiche delle porte non possono essere utilizzate in questa modalità di rete. Un container in una definizione di attività che utilizza questa modalità deve specificare un numero di <code>hostPort</code> specifico. Un numero di porta su un host non può essere utilizzato da più attività. Di conseguenza, non puoi eseguire più attività con la stessa definizione di attività su una singola EC2 istanza Amazon.
none	Sì	No	L'attività non dispone di connettività di rete esterna.
default	No	Sì	L'attività utilizza la rete virtuale integrata di Docker su Windows, che viene eseguita all'interno di ogni EC2 istanza Amazon che ospita l'attività. La rete virtuale integrata su Windows utilizza il driver di rete nat Docker. Questa è la modalità di rete predefinita su Windows se non viene specificata una modalità di rete nella definizione di attività.

Per ulteriori informazioni sulla rete Docker su Linux, consulta [Panoramica della rete](#) nella documentazione di Docker.

Per ulteriori informazioni sulla rete Docker su Windows, consulta [Rete di container Windows](#) nella documentazione di Microsoft Containers on Windows.

Assegna un'interfaccia di rete per un'attività Amazon ECS

Le funzionalità di task networking fornite dalla modalità di `awsvpc` rete conferiscono alle attività di Amazon ECS le stesse proprietà di rete delle EC2 istanze Amazon. L'utilizzo della modalità di `awsvpc` rete semplifica il networking in container, poiché hai un maggiore controllo sul modo in cui

le tue applicazioni comunicano tra loro e con gli altri servizi interni. VPCs La modalità `awsvpc` di rete offre inoltre una maggiore sicurezza per i container, consentendovi di utilizzare gruppi di sicurezza e strumenti di monitoraggio della rete a un livello più granulare nell'ambito delle vostre attività. Puoi anche utilizzare altre funzionalità EC2 di rete di Amazon come VPC Flow Logs per monitorare il traffico da e verso le tue attività. Inoltre, i container che appartengono alla stessa attività possono comunicare tramite l'interfaccia `localhost`.

La task elastic network interface (ENI) è una funzionalità completamente gestita di Amazon ECS. Amazon ECS crea l'ENI e lo collega all' EC2 istanza Amazon host con il gruppo di sicurezza specificato. L'attività invia e riceve il traffico di rete sull'ENI nello stesso modo in cui lo fanno EC2 le istanze Amazon con le loro interfacce di rete principali. Per impostazione predefinita, a ciascuna attività ENI viene assegnato un IPv4 indirizzo privato. Se il tuo VPC è abilitato per la modalità dual-stack e utilizzi una sottorete con un blocco IPv6 CIDR, anche l'attività ENI riceverà un indirizzo. IPv6 Ogni processo può avere una sola ENI.

Questi ENIs sono visibili nella EC2 console Amazon del tuo account. Il tuo account non può scollegare o modificare il ENIs. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Puoi visualizzare le informazioni sugli allegati ENI per le attività nella console Amazon ECS o con il funzionamento dell'[DescribeTasks](#) API. Quando l'attività viene interrotta o se il servizio viene ridotto, l'ENI dell'attività viene scollegata ed eliminata.

Se hai bisogno di una maggiore densità ENI, utilizza le impostazioni dell'`awsvpcTrunkingaccount`. Amazon ECS crea e collega anche un'interfaccia di rete «trunk» per l'istanza del contenitore. La rete trunk è completamente gestita da Amazon ECS. L'ENI trunk viene eliminata quando termini o annulli la registrazione dell'istanza di container dal cluster Amazon ECS. Per ulteriori informazioni sull'impostazione dell'`awsvpcTrunkingaccount`, consulta. [Prerequisiti](#)

È necessario specificare `awsvpc` nel `networkMode` parametro della definizione dell'attività. Per ulteriori informazioni, consulta [Modalità di rete](#).

Quindi, quando esegui un'attività o crei un servizio, utilizza il `networkConfiguration` parametro che include una o più sottoreti in cui inserire le attività e uno o più gruppi di sicurezza da collegare a un ENI. Per ulteriori informazioni, consulta [Configurazione della rete](#). Le attività vengono collocate su EC2 istanze Amazon compatibili nelle stesse zone di disponibilità di tali sottoreti e i gruppi di sicurezza specificati sono associati all'ENI predisposto per l'attività.

Considerazioni per Linux

Tieni in considerazione le informazioni seguenti durante l'utilizzo del sistema operativo Linux.

- Se utilizzi un'istanza p5.48xlarge in aws-vpc modalità, non puoi eseguire più di un'attività sull'istanza.
- Le attività e i servizi che utilizzano la modalità di aws-vpc rete richiedono il ruolo collegato ai servizi Amazon ECS per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#). Puoi anche creare il ruolo collegato al servizio con il seguente comando: AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- La tua istanza Amazon EC2 Linux richiede una versione 1.15.0 o successiva dell'agente container per eseguire attività che utilizzano la modalità aws-vpc di rete. Se utilizzi l'AMI ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione 1.15.0-4 del pacchetto ecs-init.
- Amazon ECS popola il nome host del processo con un nome host DNS (interno) fornito da Amazon quando entrambe le opzioni enableDnsHostnames e enableDnsSupport e sono abilitate nel VPC. Se queste opzioni non sono abilitate, il nome host DNS dell'attività è impostato su un nome host casuale. Per ulteriori informazioni sulle impostazioni DNS per un VPC, consulta [Utilizzo del DNS con il VPC](#) nella Guida per l'utente di Amazon VPC.
- Ogni attività Amazon ECS che utilizza la modalità di aws-vpc rete riceve la propria elastic network interface (ENI), collegata all' EC2 istanza Amazon che la ospita. Esiste una quota predefinita per il numero di interfacce di rete che possono essere collegate a un'istanza Amazon EC2 Linux. L'interfaccia di rete principale viene calcolata come unica rispetto a quella quota. Ad esempio, per impostazione predefinita, un'c5.18xlarge istanza può avere solo fino a tre ENIs che possono essere collegate ad essa. L'interfaccia di rete principale per l'istanza viene calcolata come unica. È possibile allegarne altre due ENIs all'istanza. Poiché ogni attività che utilizza la modalità di rete aws-vpc richiede un'ENI, in genere puoi eseguire solo due attività su questo tipo di istanza. Per ulteriori informazioni sui limiti ENI predefiniti per ogni tipo di istanza, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Amazon ECS supporta il lancio di istanze Amazon EC2 Linux che utilizzano tipi di istanze supportati con una maggiore densità ENI. Quando attivi l'impostazione dell'aws-vpcTrunkingaccount e registri istanze Amazon EC2 Linux che utilizzano questi tipi di istanze nel tuo cluster, queste istanze hanno una quota ENI più elevata. L'utilizzo di queste istanze con questa quota più elevata significa che puoi inserire più attività su ciascuna istanza Amazon EC2 Linux. Per utilizzare la maggiore densità ENI con la funzionalità di trunking, l' EC2

istanza Amazon deve utilizzare una versione 1.28.1 o successiva dell'agente container. Se utilizzi l'AMI ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione 1.28.1-2 del pacchetto `ecs-init`. Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsVpcTrunking`, consulta [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#). Per ulteriori informazioni su trunking dell'ENI, consulta [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#).

- Quando si ospitano attività che utilizzano la modalità di `awsVpc` rete su istanze Amazon EC2 Linux, alle attività ENIs non vengono assegnati indirizzi IP pubblici. Per accedere a Internet, le attività devono essere avviate in una sottorete privata configurata per l'utilizzo di un gateway NAT. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC. L'accesso di rete in entrata deve avvenire all'interno di un VPC che utilizza l'indirizzo IP privato oppure deve essere instradato attraverso un load balancer dall'interno del VPC. Le attività avviate all'interno di sottoreti pubbliche non hanno accesso a Internet.
- Amazon ECS riconosce solo ENIs ciò che è collegato alle tue istanze Amazon EC2 Linux. Se ti colleghi manualmente ENIs alle tue istanze, Amazon ECS potrebbe tentare di aggiungere un'attività a un'istanza che non dispone di adattatori di rete sufficienti. Ciò può comportare il timeout dell'attività e il passaggio a uno stato di deprovisioning e quindi allo stato di arresto. Ti consigliamo di non collegarti ENIs alle istanze manualmente.
- Le istanze Amazon EC2 Linux devono essere registrate con la `ecs.capability.task-eni` capacità di essere prese in considerazione per il posizionamento delle attività in modalità `awsVpc` rete. Le istanze che eseguono la versione 1.15.0-4 o successiva di `ecs-init` vengono registrate automaticamente con questo attributo.
- Le istanze create e collegate alle tue istanze Amazon EC2 Linux non possono essere scollegate manualmente o modificate dal tuo account. ENIs Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare il comando ENIs for a task, interrompi l'operazione.
- È possibile specificare solo 16 sottoreti e 5 gruppi di sicurezza in `awsVpcConfiguration` durante l'esecuzione di un'attività o la creazione di un servizio che utilizza la modalità di rete `awsVpc`. Per ulteriori informazioni, consulta il riferimento [AwsVpcConfiguration](#) all'API di Amazon Elastic Container Service.
- Quando un processo viene avviato con la modalità di rete `awsVpc`, l'agente del container di Amazon ECS crea un container pause aggiuntivo per ciascun processo prima di avviare i container nella definizione di attività. Quindi configura lo spazio dei nomi di rete del pause contenitore eseguendo i plugin CNI. [amazon-ecs-cni-plugins](#) Quindi l'agente avvia il resto dei container nell'attività in modo che condividano lo stack di rete del container pause. Questo

significa che tutti i container in un'attività sono indirizzabili tramite gli indirizzi IP dell'ENI e che possono comunicare tra loro sull'interfaccia `localhost`.

- I servizi con attività che utilizzano la modalità di rete `awsvpc` supportano solo Application Load Balancer e Network Load Balancer. Inoltre, quando crei gruppi target per questi servizi, devi scegliere `ip` come tipo di target. Non usare `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'ENI, non a un'istanza Amazon EC2 Linux. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).
- Se il VPC viene aggiornato per modificare le opzioni DHCP utilizzate, non puoi applicare queste modifiche alle attività esistenti. Avvia nuove attività con queste modifiche applicate, verifica che funzionino correttamente e quindi interrompi le attività esistenti per modificare in modo sicuro queste configurazioni di rete.

Considerazioni per Windows

Le seguenti considerazioni sono relative all'utilizzo del sistema operativo Windows:

- Le istanze di container che utilizzano l'AMI Windows Server 2016 ottimizzata per Amazon ECS non possono ospitare attività che utilizzano la modalità di rete `awsvpc`. Se disponi di un cluster che contiene Windows Server 2016 ottimizzato per Amazon ECS AMIs e Windows AMIs che supporta la modalità di `awsvpc` rete, le attività che utilizzano la modalità di `awsvpc` rete non vengono avviate sulle istanze di Windows 2016 Server. Piuttosto, vengono avviati su istanze che supportano le modalità di rete `awsvpc`.
- La tua istanza Amazon EC2 Windows richiede una versione `1.57.1` o successiva dell'agente contenitore per utilizzare i CloudWatch parametri per i contenitori Windows che utilizzano la modalità di `awsvpc` rete.
- Le attività e i servizi che utilizzano la modalità di `awsvpc` rete richiedono il ruolo collegato ai servizi Amazon ECS per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#). Puoi anche creare il ruolo collegato al servizio con il seguente comando. AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- La tua istanza Amazon EC2 Windows richiede una versione `1.54.0` o successiva dell'agente contenitore per eseguire attività che utilizzano la modalità `awsvpc` di rete. Quando avvii l'istanza,

devi configurare le opzioni necessarie per la modalità di rete `awsvpc`. Per ulteriori informazioni, consulta [the section called “Avvio delle istanze dei container”](#).

- Amazon ECS popola il nome host dell'attività con un nome host DNS (interno) fornito da Amazon quando entrambe le opzioni `enableDnsHostnames` e `enableDnsSupport` sono abilitate nel VPC. Se queste opzioni non sono abilitate, il nome host DNS dell'attività sarà un nome host casuale. Per ulteriori informazioni sulle impostazioni DNS per un VPC, consulta [Utilizzo del DNS con il VPC](#) nella Guida per l'utente di Amazon VPC.
- Ogni attività Amazon ECS che utilizza la modalità di `awsvpc` rete riceve la propria elastic network interface (ENI), collegata all'istanza Amazon EC2 Windows che la ospita. Esiste una quota predefinita per il numero di interfacce di rete che possono essere collegate a un'istanza Amazon EC2 Windows. L'interfaccia di rete principale viene calcolata come unica rispetto a questa quota. Ad esempio, per impostazione predefinita, a un'istanza `c5.large` possono essere ENI collegate solo fino a tre istanze. L'interfaccia di rete principale per l'istanza viene calcolata come una di quelle. È possibile allegarne altre due ENI all'istanza. Poiché ogni attività che utilizza la modalità di rete `awsvpc` richiede un'ENI, in genere puoi eseguire solo due attività su questo tipo di istanza. Per ulteriori informazioni sui limiti ENI predefiniti per ogni tipo di istanza, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Quando si ospitano attività che utilizzano la modalità di `awsvpc` rete su istanze Amazon EC2 Windows, alle attività ENI non vengono assegnati indirizzi IP pubblici. Per accedere a Internet, avvia le attività in una sottorete privata configurata per l'utilizzo di un gateway NAT. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC. L'accesso di rete in entrata deve avvenire all'interno del VPC utilizzando l'indirizzo IP privato oppure deve essere instradato attraverso un load balancer all'interno del VPC. Le attività avviate all'interno di sottoreti pubbliche non hanno accesso a Internet.
- Amazon ECS riconosce solo ENI ciò che è collegato alla tua istanza Amazon EC2 Windows. Se ti colleghi manualmente ENI alle tue istanze, Amazon ECS potrebbe tentare di aggiungere un'attività a un'istanza che non dispone di adattatori di rete sufficienti. Ciò può comportare il timeout dell'attività e il passaggio a uno stato di `deprovisioning` e quindi allo stato di arresto. Ti consigliamo di non collegarti ENI alle istanze manualmente.
- Le istanze di Amazon EC2 Windows devono essere registrate per `ecs.capability.task-eni` poter essere prese in considerazione per il posizionamento delle attività in modalità `awsvpc` rete.
- Non puoi modificare o scollegare ENI manualmente le istanze create e collegate alle tue istanze Amazon EC2 Windows. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare il comando `ENIs for a task`, interrompi l'operazione.

- Puoi specificare solo fino a 16 sottoreti e 5 gruppi di sicurezza in `awsVpcConfiguration` quando esegui un'attività o crei un servizio che utilizza la modalità di rete `awsvpc`. Per ulteriori informazioni, consulta il riferimento [AwsVpcConfiguration](#) all'API di Amazon Elastic Container Service.
- Quando un processo viene avviato con la modalità di rete `awsvpc`, l'agente del container di Amazon ECS crea un container pause aggiuntivo per ciascun processo prima di avviare i container nella definizione di attività. Quindi configura lo spazio dei nomi di rete del pause contenitore eseguendo i plugin CNI. [amazon-ecs-cni-plugins](#) Quindi l'agente avvia il resto dei container nell'attività in modo che condividano lo stack di rete del container pause. Questo significa che tutti i container in un'attività sono indirizzabili tramite gli indirizzi IP dell'ENI e che possono comunicare tra loro sull'interfaccia `localhost`.
- I servizi con attività che utilizzano la modalità di rete `awsvpc` supportano solo Application Load Balancer e Network Load Balancer. Quando crei gruppi target per questi servizi, devi scegliere `ip` come tipo di target anziché `instance`. Questo perché le attività che utilizzano la modalità di rete `awsvpc` sono associate a un'ENI, non a un'istanza Amazon EC2 Windows. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).
- Se il VPC viene aggiornato per modificare le opzioni DHCP utilizzate, non puoi applicare queste modifiche alle attività esistenti. Avvia nuove attività con queste modifiche applicate, verifica che funzionino correttamente e quindi interrompi le attività esistenti per modificare in modo sicuro queste configurazioni di rete.
- Quanto segue non è supportato quando si utilizza la modalità `awsvpc` di rete in una configurazione EC2 Windows:
 - Configurazione dual-stack
 - IPv6
 - Trunking ENI

Utilizzo di un VPC in modalità dual-stack

Quando si utilizza un VPC in modalità dual-stack, le attività possono comunicare più IPv4 o meno entrambe. IPv4 e IPv6 gli indirizzi sono indipendenti l'uno dall'altro. Pertanto è necessario configurare il routing e la sicurezza nel VPC separatamente per IPv4 e IPv6. Per ulteriori informazioni su come configurare il tuo VPC per la modalità dual-stack, consulta [Migrating to nella](#) Amazon VPC User IPv6 Guide.

Se hai configurato il tuo VPC con un gateway Internet o un gateway Internet solo in uscita, puoi utilizzare il VPC in modalità dual-stack. In questo modo, le attività a cui viene assegnato un IPv6 indirizzo possono accedere a Internet tramite un gateway Internet o un gateway Internet solo in uscita. I gateway NAT sono opzionali. Per ulteriori informazioni, consulta [Gateway Internet](#) e [Gateway Internet egress-only](#) nella Guida per l'utente di Amazon VPC.

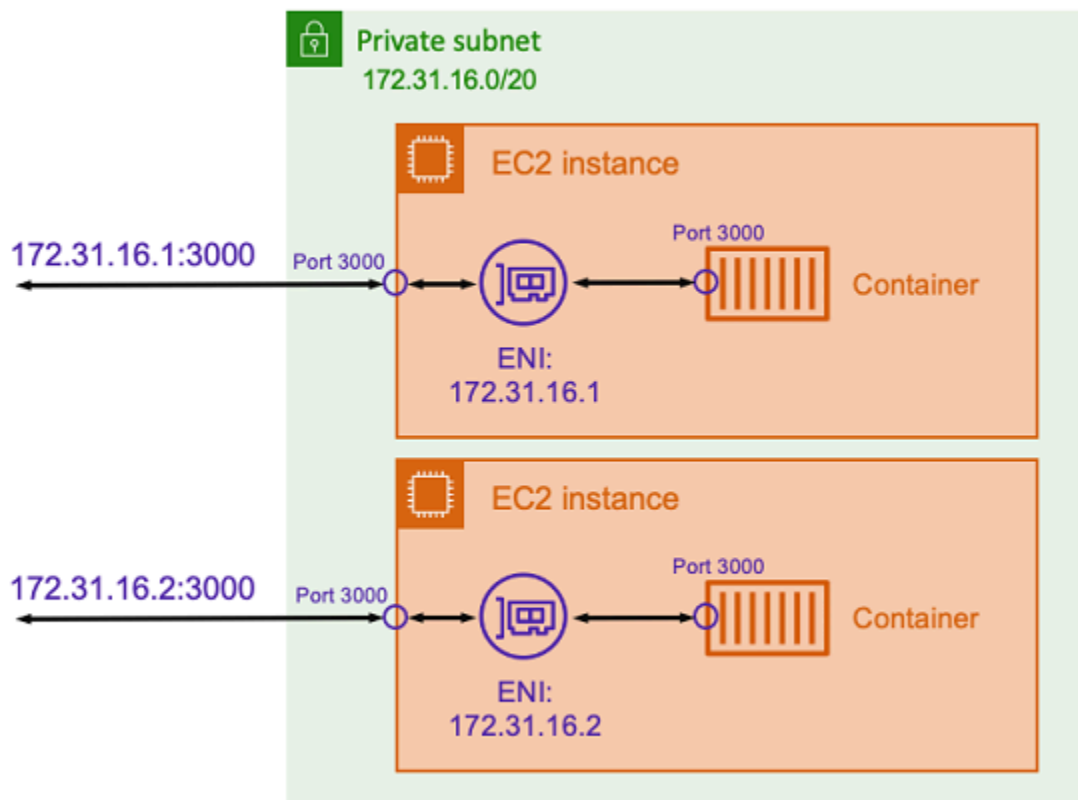
Alle attività di Amazon ECS viene assegnato un IPv6 indirizzo se vengono soddisfatte le seguenti condizioni:

- L'istanza Amazon EC2 Linux che ospita l'attività utilizza una versione 1.45.0 o successiva dell'agente contenitore. Per informazioni sul controllo della versione dell'agente utilizzata dall'istanza e sull'aggiornamento, se necessario, consulta [Aggiornamento dell'agente del container Amazon ECS](#).
- L'impostazione dell'account `dualStackIPv6` è abilitata. Per ulteriori informazioni, consulta [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#).
- Il tuo processo sta usando la modalità di rete `awsvpc`.
- Il VPC e la sottorete sono configurati per IPv6. La configurazione include le interfacce di rete create nella sottorete specificata. Per ulteriori informazioni su come configurare il tuo VPC per la modalità dual-stack, consulta [Migrazione IPv6 e modifica dell'attributo di IPv6 indirizzamento per la tua sottorete nella](#) Amazon VPC User Guide.

Mappa le porte dei container Amazon ECS all'interfaccia di rete dell' EC2 istanza

La modalità `host` di rete è supportata solo per le attività di Amazon ECS ospitate su EC2 istanze Amazon. Non è supportata quando si utilizza Fargate con Amazon ECS.

La modalità di rete `host` è la modalità di rete più semplice supportata in Amazon ECS. Utilizzando la modalità `host`, la rete del container è collegata direttamente all'host sottostante che esegue il container.



Si supponga di eseguire un container Node.js con un'applicazione Express in ascolto sulla porta 3000 simile a quella illustrata nel diagramma precedente. Quando viene utilizzata la modalità di host rete, il contenitore riceve il traffico sulla porta 3000 utilizzando l'indirizzo IP dell' EC2 istanza Amazon host sottostante. Ti consigliamo di non utilizzare questa modalità.

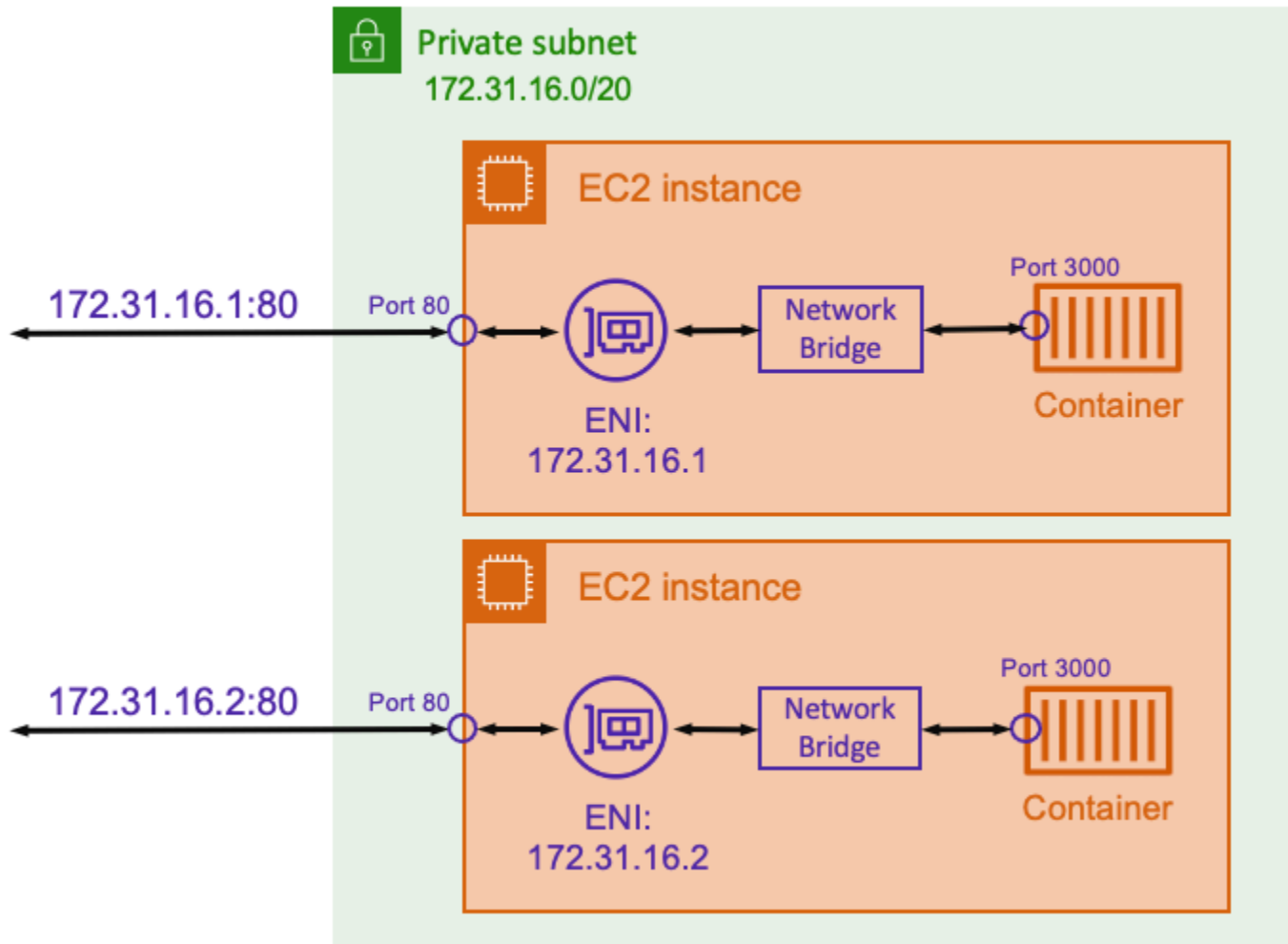
L'utilizzo di questa modalità di rete presenta notevoli svantaggi. È possibile creare solo una singola istanza di un'attività su ciascun host. Questo perché solo la prima attività può collegarsi alla porta richiesta sull' EC2 istanza Amazon. Inoltre, non è possibile rimappare una porta del container quando si utilizza la modalità di rete host. Ad esempio, se un'applicazione deve essere in ascolto su un determinato numero di porta, non è possibile rimappare direttamente il numero di porta. È invece necessario gestire eventuali conflitti di porte modificando la configurazione dell'applicazione.

L'utilizzo della modalità di rete host comporta anche implicazioni in termini di sicurezza. Questa modalità consente ai container di impersonare l'host e di connettersi ai servizi di rete loopback privati sull'host.

Usa la rete virtuale di Docker per le attività di Amazon ECS Linux

La modalità `bridge` di rete è supportata solo per le attività di Amazon ECS ospitate su EC2 istanze Amazon.

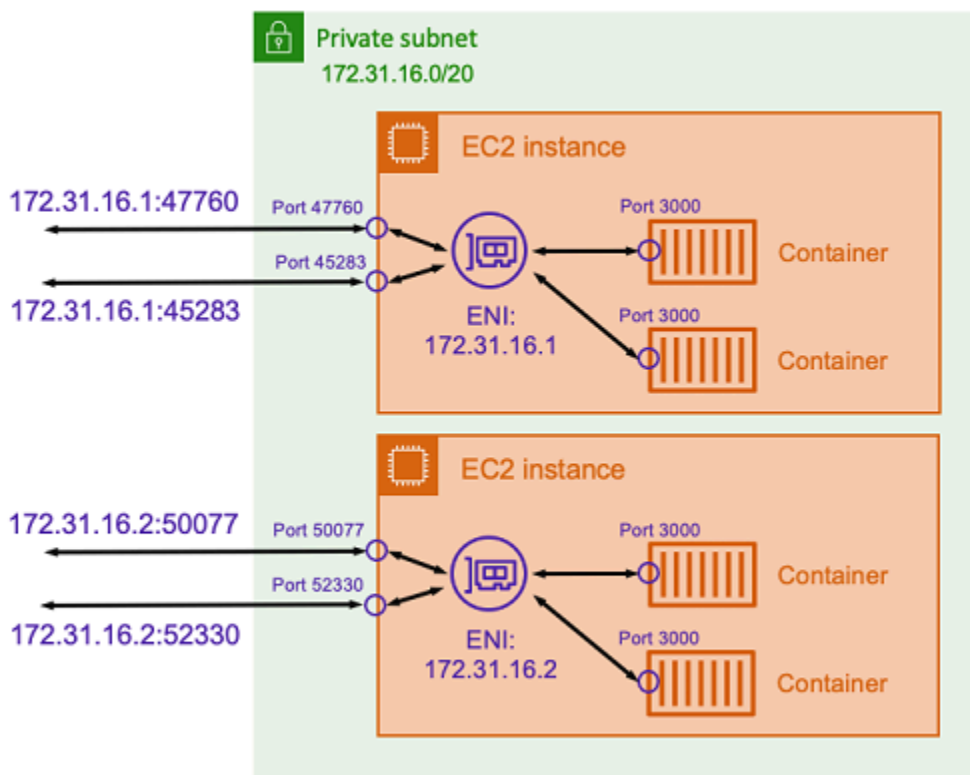
Con la modalità `bridge`, stai utilizzando un bridge di rete virtuale per creare un livello tra l'host e la rete del container. In questo modo, puoi creare mappature delle porte che rimappano una porta host su una porta container. Le mappature possono essere statiche o dinamiche.



Con una mappatura statica delle porte, puoi definire in modo esplicito la porta host da mappare alla porta container. Nell'esempio precedente, la porta 80 dell'host viene mappata sulla porta 3000 del container. Per comunicare con l'applicazione containerizzata, invii il traffico alla porta 80 all'indirizzo IP dell' EC2 istanza Amazon. Dal punto di vista dell'applicazione containerizzata, essa rileva il traffico in entrata sulla porta 3000.

Se vuoi modificare solo la porta del traffico, la mappatura statica rappresenta la soluzione migliore. Tuttavia, presenta lo stesso svantaggio dell'utilizzo della modalità di rete host. È possibile creare solo una singola istanza di un'attività su ciascun host, dal momento che la mappatura statica delle porte consente di mappare solo un solo container sulla porta 80.

Per risolvere questo problema, prendi in considerazione l'utilizzo della modalità di rete bridge con una mappatura dinamica delle porte, come illustrato nel diagramma seguente.



Se non specifichi una porta host nella mappatura delle porte, Docker sceglie una porta casuale e inutilizzata dall'intervallo di porte provvisorie e la assegna come porta host pubblica per il container. Ad esempio, all'applicazione Node.js che ascolta 3000 sulla porta del contenitore potrebbe essere assegnata una porta a numero elevato casuale, ad esempio 47760 sull' EC2 host Amazon. In questo modo è possibile eseguire più copie del container sull'host. Inoltre, a ciascun container può essere assegnata una propria porta sull'host. Ogni copia del container riceve il traffico sulla porta 3000. Tuttavia, i client che inviano traffico a questi container utilizzano le porte host assegnate casualmente.

Amazon ECS consente di tenere traccia delle porte assegnate casualmente per ogni attività. A tale scopo, aggiorna automaticamente i gruppi target del sistema di bilanciamento del carico e il

rilevamento dei AWS Cloud Map servizi in modo da disporre dell'elenco degli indirizzi IP e delle porte delle attività. Ciò semplifica l'utilizzo dei servizi che operano in modalità bridge con porte dinamiche.

Tuttavia, uno svantaggio dell'utilizzo della modalità di rete bridge è la difficoltà di bloccare le comunicazioni tra servizi. Poiché i servizi possono essere assegnati a qualsiasi porta casuale e inutilizzata, è necessario aprire ampi intervalli di porte tra gli host. Tuttavia, non è semplice creare regole specifiche in modo che un determinato servizio possa comunicare solo con un altro servizio specifico. I servizi non dispongono di porte specifiche da utilizzare per le regole di rete dei gruppi di sicurezza.

Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate

Di default, a ogni processo di Amazon ECS su Fargate viene fornita una interfaccia di rete elastica (ENI) con un indirizzo IP privato primario. Quando utilizzi una sottorete pubblica, puoi eventualmente assegnare un indirizzo IP pubblico all'ENI dell'attività. Se il tuo VPC è configurato per la modalità dual-stack e utilizzi una sottorete con un blocco IPv6 CIDR, anche l'ENI dell'attività riceve un indirizzo. IPv6 Un'attività può avere una sola ENI associata in un determinato momento. I container che appartengono alla stessa attività possono comunicare tramite l'interfaccia localhost. Per ulteriori informazioni sulle VPCs sottoreti, consulta [How Amazon VPC works nella Amazon VPC User Guide](#).

Affinché un'attività su Fargate sia in grado di estrarre un'immagine del container, l'attività deve avere un routing verso Internet. Di seguito è descritto come verificare che l'attività abbia un routing verso Internet.

- Quando utilizzi una sottorete pubblica, puoi assegnare un indirizzo IP pubblico all'ENI del processo.
- Quando si utilizza una sottorete privata, la sottorete può avere un gateway NAT collegato.
- Quando utilizzi immagini di container ospitate in Amazon ECR, puoi configurare Amazon ECR per utilizzare un endpoint VPC di interfaccia e il recupero dell'immagine avviene sull'indirizzo privato dell'attività. IPv4 Per ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia Amazon ECR \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.

Poiché ogni attività ottiene la sua ENI, puoi utilizzare funzioni di rete, come i log di flusso VPC, per monitorare il traffico da e verso le tue attività. Per ulteriori informazioni, consulta [Log di flusso VPC](#) nella Guida per l'utente di Amazon VPC.

Puoi anche approfittare di AWS PrivateLink. Puoi configurare un endpoint di interfaccia VPC in modo da poter accedere ad Amazon ECS APIs tramite indirizzi IP privati. AWS PrivateLink limita tutto il

traffico di rete tra il tuo VPC e Amazon ECS alla rete Amazon. Non è richiesto un gateway Internet, un dispositivo NAT o un gateway privato virtuale. Per ulteriori informazioni, consulta [Endpoint AWS PrivateLink VPC dell'interfaccia Amazon ECS](#) ().

Per esempi di come utilizzare la NetworkConfiguration risorsa con AWS CloudFormation, consulta. [the section called “Modelli di esempio AWS CloudFormation”](#)

ENIs Quelli che vengono creati sono completamente gestiti da AWS Fargate. Inoltre, esiste una policy IAM associata che viene utilizzata per concedere autorizzazioni per Fargate. Per le attività che utilizzano la piattaforma Fargate versione 1.4.0 o successiva, l'attività riceve un'unica ENI (denominata ENI di attività) e tutto il traffico di rete scorre attraverso tale ENI all'interno del VPC. Questo traffico viene registrato nei log di flusso VPC. Per le attività che utilizzano la piattaforma Fargate versione 1.3.0 e precedenti, oltre all'ENI di attività, l'attività riceve anche un'ENI di proprietà di Fargate che viene utilizzata per un traffico di rete non visibile nei log di flusso VPC. Di seguito vengono descritti il comportamento del traffico di rete e la policy IAM richiesta per ogni versione della piattaforma.

Azione	Flusso di traffico con piattaforma Linux versione 1.3.0 e versioni precedenti	Flusso di traffico con piattaforma Linux versione 1.4.0	Flusso di traffico con piattaforma Windows versione 1.0.0	Autorizzazione IAM
Recupero delle credenziali di accesso di Amazon ECR	ENI di proprietà Fargate	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Pull immagine	ENI attività	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Invio dei log tramite un driver di log	ENI attività	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo

Azione	Flusso di traffico con piattaforma Linux versione 1.3.0 e versioni precedenti	Flusso di traffico con piattaforma Linux versione 1.4.0	Flusso di traffico con piattaforma Windows versione 1.0.0	Autorizzazione IAM
Invio di log FireLens per Amazon ECS	ENI attività	ENI attività	ENI attività	Ruolo IAM del processo
Recupero di segreti da Secrets Manager o Systems Manager	ENI di proprietà Fargate	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Traffico del file system Amazon EFS	Non disponibile	ENI attività	ENI attività	Ruolo IAM del processo
Traffico delle applicazioni	ENI attività	ENI attività	ENI attività	Ruolo IAM del processo

Considerazioni

Tieni in considerazione le informazioni seguenti quando usi la rete di attività.

- Il ruolo collegato ai servizi Amazon ECS è necessario per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#). Puoi anche creare il ruolo collegato al servizio utilizzando il seguente comando. AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- Amazon ECS popola il nome host dell'attività con un nome host DNS fornito da Amazon quando entrambe le opzioni `enableDnsHostnames` e `enableDnsSupport` sono abilitate nel VPC. Se queste opzioni non sono abilitate, il nome host DNS dell'attività è impostato su un nome host

casuale. Per ulteriori informazioni sulle impostazioni DNS per un VPC, consulta [Utilizzo del DNS con il VPC](#) nella Guida per l'utente di Amazon VPC.

- È possibile specificare solo fino a 16 sottoreti e 5 gruppi di sicurezza per `awsVpcConfiguration`. Per ulteriori informazioni, consulta il riferimento [AwsVpcConfiguration](#) all'API di Amazon Elastic Container Service.
- Non è possibile scollegare o modificare manualmente i ENIs file creati e allegati da Fargate. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare il comando ENIs for a task, interrompete l'operazione.
- Se una sottorete VPC viene aggiornata per modificare la serie di opzioni DHCP utilizzate, non puoi applicare queste modifiche anche alle attività esistenti che utilizzano il VPC. Avvia nuove attività, che riceveranno la nuova impostazione per eseguire la migrazione senza problemi durante il test della nuova modifica e quindi interrompi le attività precedenti, se non è necessario eseguire il rollback.
- Quanto segue si applica alle attività eseguite sulla versione della piattaforma Fargate 1.4.0 o successiva per Linux o 1.0.0 Windows. Le attività avviate in sottoreti dual-stack ricevono un indirizzo e un indirizzo. IPv4 IPv6
- Per le attività che utilizzano la versione della piattaforma 1.4.0 o successiva per Linux o Windows, 1.0.0 l'attività supporta i frame jumbo. ENIs Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimensione del payload più grande che si adatta all'interno di un singolo frame. Più grande è l'MTU, più il payload dell'applicazione può essere adattato all'interno di un singolo fotogramma, riducendo il sovraccarico per fotogramma e aumentando l'efficienza. Il supporto dei frame jumbo riduce il sovraccarico quando il percorso di rete tra l'attività e la destinazione supporta frame jumbo.
- I servizi con attività che utilizzano il tipo di avvio Fargate supportano solo Application Load Balancer e Network Load Balancer. Classic Load Balancer non è supportato. Quando crei gruppi target, devi scegliere `ip` come tipo di target anziché `instance`. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

Utilizzo di un VPC in modalità dual-stack

Quando si utilizza un VPC in modalità dual-stack, le attività possono comunicare più IPv4 o meno entrambe. IPv6 IPv4 e IPv6 gli indirizzi sono indipendenti l'uno dall'altro ed è necessario configurare il routing e la sicurezza nel VPC separatamente IPv4 per e. IPv6 Per ulteriori informazioni sulla configurazione del tuo VPC per la modalità dual-stack, consulta [Migrating to nella IPv6](#) Amazon VPC User Guide.

Se vengono soddisfatte le seguenti condizioni, alle attività di Amazon ECS su Fargate viene assegnato IPv6 un indirizzo:

- L'impostazione del tuo `dualStackIPv6` account Amazon ECS è attivata (`enabled`) per il responsabile IAM che avvia le tue attività nella regione in cui le stai avviando. Questa impostazione può essere modificata solo utilizzando l'API o. AWS CLI Hai la possibilità di attivare questa impostazione per un principale IAM specifico sul tuo account o per l'intero account impostando l'impostazione predefinita dell'account. Per ulteriori informazioni, consulta [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#).
- Il VPC e la sottorete sono abilitati per. IPv6 Per ulteriori informazioni su come configurare il tuo VPC per la modalità dual-stack, consulta [Migrating to nella](#) Amazon VPC User IPv6 Guide.
- La sottorete è abilitata per l'assegnazione automatica IPv6 degli indirizzi. Per ulteriori informazioni su come configurare la sottorete, consulta [Modificare l'attributo di IPv6 indirizzamento per la sottorete](#) nella Amazon VPC User Guide.
- L'attività o il servizio utilizza la versione della piattaforma Fargate 1.4.0 o successiva per Linux.

Se configuri il tuo VPC con un gateway Internet o un gateway Internet solo in uscita, le attività di Amazon ECS su Fargate a cui è assegnato un indirizzo possono accedere a Internet. IPv6 I gateway NAT non sono necessari. Per ulteriori informazioni, consulta [Gateway Internet](#) e [Gateway Internet egress-only](#) nella Guida per l'utente di Amazon VPC.

Opzioni di storage per le attività di Amazon ECS

Amazon ECS ti offre opzioni di archiviazione easy-to-use dei dati flessibili, convenienti e a seconda delle tue esigenze. Amazon ECS supporta le seguenti opzioni di volume di dati per i contenitori:

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Amazon Elastic Block Store (Amazon EBS)	Fargate, Amazon EC2	Linux	Può essere mantenuto se associato a un'attività autonoma. Effimero se associato a	I volumi Amazon EBS forniscono storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro container

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
			un'attività gestita da un servizio.	izzati a uso intensivo di dati. I casi d'uso più comuni includono carichi di lavoro transazionali come database, desktop virtuali e volumi root e carichi di lavoro ad alta intensità di throughput come l'elaborazione dei log e i carichi di lavoro ETL. Per ulteriori informazioni, consulta Usa i volumi Amazon EBS con Amazon ECS.

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Amazon Elastic File System (Amazon EFS)	Fargate, Amazon EC2	Linux	Persistente	I volumi Amazon EFS forniscono uno storage di file condiviso semplice, scalabile e persistente da utilizzare con le attività di Amazon ECS, che cresce e si riduce automaticamente man mano che aggiungi e rimuovi file. I volumi Amazon EFS supportano la concorrenza e sono utili per le applicazioni containerizzate che scalano orizzontalmente e richiedono funzionalità di storage come bassa latenza, throughput elevato e coerenza. read-after-write I casi d'uso più comuni

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				includono carichi di lavoro come analisi dei dati, elaborazione multimediale, gestione dei contenuti e server web. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS .

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
File server Amazon FSx per Windows	Amazon EC2	Windows	Persistente	FSx per i volumi Windows File Server forniscono file server Windows completamente gestiti che è possibile utilizzare e per eseguire il provisioning delle attività Windows che richiedono uno storage di file persistente, distribuito, condiviso e statico. I casi d'uso più comuni includono le applicazioni.NET che potrebbero richiedere cartelle locali come archiviazione persistente per salvare gli output delle applicazioni. Amazon FSx per Windows File Server offre una cartella locale

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				nel contenitore che consente la lettura e la scrittura di più contenuto ri sullo stesso file system supportato da una condivisione SMB. Per ulteriori informazioni, consulta Utilizzo FSx per volumi Windows File Server con Amazon ECS .

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Volumi Docker	Amazon EC2	Windows, Linux	Persistente	I volumi Docker sono una funzionalità del runtime del contenuto re Docker che consente ai contenitori di rendere persistenti i dati montando una directory dal file system dell'host. I driver di volume Docker (noti anche come plugin) vengono utilizzati per integrare i volumi dei container con sistemi di archiviazione esterni. I volumi Docker possono essere gestiti da driver di terze parti o dal driver integrato <code>local</code> . I casi d'uso comuni dei volumi Docker includono la fornitura di

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				volumi di dati persistenti o la condivisione di volumi in posizioni diverse su contenitori diversi sulla stessa istanza di contenitore. Per ulteriori informazioni, consulta Usa i volumi Docker con Amazon ECS .

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Montaggi vincolati	Fargate, Amazon EC2	Windows, Linux	Effimero	I montaggi Bind sono costituiti da un file o una directory sull'host, EC2 ad esempio un'istanza Amazon o AWS Fargate, montata su un contenitore. I casi d'uso più comuni per i bind mount includono la condivisione di un volume da un contenitore di origine con altri contenitori nello stesso compito o il montaggio di un volume host o di un volume vuoto in uno o più contenitori. Per ulteriori informazioni, consulta Usa bind mount con Amazon ECS.

Usa i volumi Amazon EBS con Amazon ECS

I volumi Amazon Elastic Block Store (Amazon EBS) forniscono storage a blocchi ad alta disponibilità, conveniente, durevole e ad alte prestazioni per carichi di lavoro a uso intensivo di dati. I volumi Amazon EBS possono essere utilizzati con le attività di Amazon ECS per applicazioni con throughput elevato e transazioni intensive.

Durante l'avvio di un'attività autonoma, puoi fornire la configurazione che verrà utilizzata per collegare un volume EBS all'attività. Durante la creazione o l'aggiornamento del servizio, puoi fornire la configurazione che verrà utilizzata per allegare un volume EBS per attività a ciascuna attività gestita dal servizio Amazon ECS.

Fornendo la configurazione del volume al momento dell'avvio anziché nella definizione dell'attività, crei definizioni delle attività che non sono limitate a un tipo di volume di dati specifico o a impostazioni specifiche del volume EBS. È quindi possibile riutilizzare le definizioni delle attività in diversi ambienti di runtime. Ad esempio, è possibile fornire un throughput maggiore durante l'implementazione per i carichi di lavoro di produzione rispetto agli ambienti pre-produzione.

I volumi Amazon EBS collegati alle attività di Amazon ECS sono gestiti da Amazon ECS per tuo conto. I volumi possono essere crittografati con chiavi AWS Key Management Service (AWS KMS) per proteggere i dati. È possibile configurare nuovi volumi vuoti per gli allegati oppure utilizzare istantanee per caricare dati da volumi esistenti. Quando si utilizzano le istantanee per configurare i volumi, è possibile specificare un intervallo `volumeInitializationRate`, in MiB/s, in corrispondenza del quale i dati vengono recuperati dall'istanza per creare volumi completamente inizializzati in un periodo di tempo prevedibile. Per ulteriori informazioni sull'inizializzazione dei volumi, consulta [Inizializza i volumi Amazon EBS](#) nella Amazon EBS User Guide. Per ulteriori informazioni sulla configurazione dei volumi Amazon EBS, consulta [Rimanda la configurazione del volume all'ora di avvio in una definizione di attività Amazon ECS](#) e [Specificare la configurazione del volume Amazon EBS nella distribuzione di Amazon ECS](#).

Per monitorare le prestazioni del volume, puoi anche utilizzare i CloudWatch parametri di Amazon. Per ulteriori informazioni sui parametri di Amazon ECS per i volumi Amazon EBS, consulta i parametri di [Metriche di Amazon ECS CloudWatch](#) [Amazon ECS](#) Container Insights.

Il collegamento di un volume Amazon EBS a un'attività è supportato in tutte le aziende commerciali e in Cina [Regioni AWS](#) che supportano Amazon ECS.

Per ulteriori informazioni sui volumi Amazon EBS, consulta i volumi [Amazon EBS](#) nella Amazon EBS User Guide.

Sistemi operativi e tipi di avvio supportati

La tabella seguente fornisce le configurazioni del sistema operativo e del tipo di avvio supportate.

Tipo di avvio	Linux	Windows
Fargate	I volumi Amazon EBS sono supportati nella versione della piattaforma 1.4.0 o successiva (Linux). Per ulteriori informazioni, consulta Versioni della piattaforma Fargate per Amazon ECS .	Non supportato
EC2	Attività ospitate su Nitroistanze Linux basate su Amazon Machine Images AMIs () ottimizzate per Amazon ECS. Per ulteriori informazioni sui tipi di istanze, consulta Tipi di istanza nella Amazon EC2 User Guide. I volumi Amazon EBS sono supportati su 20231219 AMI ottimizzate per ECS o versioni successive. Per ulteriori informazioni, consulta Recupero dei metadati dell'AMI ottimizzata per Amazon ECS.	Attività ospitate su Nitroistanze Linux basate su Amazon Machine Images AMIs () ottimizzate per Amazon ECS. Per ulteriori informazioni sui tipi di istanze, consulta Tipi di istanza nella Amazon EC2 User Guide. I volumi Amazon EBS sono supportati su 20241017 AMI ottimizzate per ECS o versioni successive. Per ulteriori informazioni, consulta Recupero di metadati AMI Windows ottimizzati per Amazon ECS.

Considerazioni

Quando utilizzi i volumi Amazon EBS, considera quanto segue:

- Non è possibile configurare i volumi Amazon EBS da allegare alle attività Amazon ECS di tipo Fargate nella use1-az3 zona di disponibilità.

- Il tipo di volume Amazon EBS magnetico (standard) non è supportato per le attività ospitate su Fargate. Per ulteriori informazioni sui tipi di volume Amazon EBS, consulta [i volumi Amazon EBS](#) nella Amazon EC2 User Guide.
- Un ruolo IAM dell'infrastruttura Amazon ECS è necessario quando si crea un servizio o un'attività autonoma che consiste nella configurazione di un volume al momento della distribuzione. Puoi allegare la policy AmazonECSInfrastructureRolePolicyForVolumes IAM AWS gestita al ruolo oppure puoi utilizzare la policy gestita come guida per creare e allegare la tua policy con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- Puoi allegare al massimo un volume Amazon EBS a ciascuna attività Amazon ECS e deve trattarsi di un nuovo volume. Non puoi collegare un volume Amazon EBS esistente a un'attività. Tuttavia, puoi configurare un nuovo volume Amazon EBS al momento della distribuzione utilizzando lo snapshot di un volume esistente.
- Puoi configurare i volumi Amazon EBS durante la distribuzione solo per i servizi che utilizzano il tipo di distribuzione con aggiornamento continuo e la strategia di pianificazione delle repliche.
- Affinché un contenitore della tua attività possa scrivere sul volume Amazon EBS montato, devi eseguire il contenitore come utente root.
- Amazon ECS aggiunge automaticamente i tag riservati AmazonECSCreated e AmazonECSManaged al volume allegato. Se rimuovi questi tag dal volume, Amazon ECS non sarà in grado di gestire il volume per tuo conto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta [Tagging dei volumi Amazon EBS](#). Per ulteriori informazioni sull'etichettatura delle risorse Amazon ECS, consulta [Tagging your Amazon ECS resources](#).
- Il provisioning di volumi da uno snapshot di un volume Amazon EBS che contiene partizioni non è supportato.
- I volumi collegati alle attività gestite da un servizio non vengono conservati e vengono sempre eliminati al termine dell'attività.
- Non è possibile configurare i volumi Amazon EBS per il collegamento alle attività di Amazon ECS in esecuzione. AWS Outposts

Rimanda la configurazione del volume all'ora di avvio in una definizione di attività Amazon ECS

Per configurare un volume Amazon EBS da allegare alla tua attività, devi specificare la configurazione del punto di montaggio nella definizione dell'attività e assegnare un nome al volume. È inoltre necessario configurare `configuredAtLaunch` impostare `true` perché i volumi Amazon EBS non

possono essere configurati per l'allegato nella definizione dell'attività. Invece, i volumi Amazon EBS sono configurati per l'allegato durante la distribuzione.

Per registrare la definizione dell'attività utilizzando AWS Command Line Interface (AWS CLI), salva il modello come file JSON, quindi passa il file come input per il [register-task-definition](#) comando.

Per creare e registrare una definizione di attività utilizzando il AWS Management Console, vedere [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

La seguente definizione di attività mostra la sintassi degli volumes oggetti mountPoints e nella definizione dell'attività. Per ulteriori informazioni sui parametri di definizione delle attività, vedere [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#). Per utilizzare questo comando, sostituisci *user input placeholders* con le tue informazioni.

Linux

```
{
  "family": "mytaskdef",
  "containerDefinitions": [
    {
      "name": "nginx",
      "image": "public.ecr.aws/nginx/nginx:latest",
      "networkMode": "awsvpc",
      "portMappings": [
        {
          "name": "nginx-80-tcp",
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp",
          "appProtocol": "http"
        }
      ],
      "mountPoints": [
        {
          "sourceVolume": "myEBSVolume",
          "containerPath": "/mount/ebs",
          "readOnly": true
        }
      ]
    }
  ],
  "volumes": [
```

```

    {
      "name": "myEBSVolume",
      "configuredAtLaunch": true
    }
  ],
  "requiresCompatibilities": [
    "FARGATE", "EC2"
  ],
  "cpu": "1024",
  "memory": "3072",
  "networkMode": "awsvpc"
}

```

Windows

```

{
  "family": "mytaskdef",
  "memory": "4096",
  "cpu": "2048",
  "family": "windows-simple-iis-2019-core",
  "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
  "runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
  "requiresCompatibilities": ["EC2"]
  "containerDefinitions": [
    {
      "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html -Type file
-Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your
application is now running on a container in Amazon ECS.</p>'; C:\\ServiceMonitor.exe
w3svc"],
      "entryPoint": [
        "powershell",
        "-Command"
      ],
      "essential": true,
      "cpu": 2048,
      "memory": 4096,
      "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
      "name": "sample_windows_app",
      "portMappings": [
        {

```

```

        "hostPort": 443,
        "containerPort": 80,
        "protocol": "tcp"
    },
    "mountPoints": [
        {
            "sourceVolume": "myEBSVolume",
            "containerPath": "drive:\ebs",
            "readOnly": true
        }
    ]
},
"volumes": [
    {
        "name": "myEBSVolume",
        "configuredAtLaunch": true
    }
],
"requiresCompatibilities": [
    "FARGATE", "EC2"
],
"cpu": "1024",
"memory": "3072",
"networkMode": "awsvpc"
}

```

mountPoints

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato `Volumes` all'API Docker `create-container` e `--volume` all'opzione `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Per le attività eseguite su EC2 istanze che eseguono il sistema operativo Windows, lasciate il valore predefinito di `false`.

`name`

Tipo: string

Campo obbligatorio: no

Nome del volume. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini (`()`) e caratteri di sottolineatura (`-`). `_` A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume mountPoints`

`configuredAtLaunch`

Tipo: Booleano

Obbligatorio: Sì, se desideri utilizzare, collega un volume EBS direttamente a un'attività.

Specifica se un volume è configurabile all'avvio. Se impostato su `true`, è possibile configurare il volume quando si esegue un'attività autonoma o quando si crea o si aggiorna un servizio. Se

impostato su `false`, non sarà possibile fornire un'altra configurazione del volume nella definizione dell'attività. Questo parametro deve essere fornito e impostato su `true` per configurare un volume Amazon EBS da allegare a un'attività.

Crittografa i dati archiviati nei volumi Amazon EBS collegati alle attività di Amazon ECS

Puoi usare AWS Key Management Service (AWS KMS) per creare e gestire chiavi crittografiche che proteggono i tuoi dati. I volumi Amazon EBS vengono crittografati a riposo utilizzando AWS KMS keys. I seguenti tipi di dati sono crittografati:

- Dati archiviati inattivi sul volume
- I/O del disco
- Istantanee create dal volume
- Nuovi volumi creati da istantanee crittografate

I volumi Amazon EBS collegati alle attività possono essere crittografati utilizzando un alias `alias/aws/ebs` predefinito Chiave gestita da AWS o una chiave simmetrica gestita dal cliente specificata nella configurazione del volume. Chiavi gestite da AWS I valori predefiniti sono unici Account AWS per ciascun utente Regione AWS e vengono creati automaticamente. Per creare una chiave simmetrica gestita dal cliente, segui i passaggi descritti in [Creazione di chiavi KMS con crittografia simmetrica](#) nella Guida per gli sviluppatori.AWS KMS

Puoi configurare la crittografia Amazon EBS per impostazione predefinita in modo che tutti i nuovi volumi creati e collegati a un'attività in una determinata operazione Regione AWS vengano crittografati utilizzando la chiave KMS specificata per il tuo account. Per ulteriori informazioni sulla crittografia e la crittografia di Amazon EBS per impostazione predefinita, consulta [Amazon EBS encryption](#) nella Amazon EBS User Guide.

Puoi anche configurare la crittografia a livello di cluster Amazon ECS per lo storage gestito di Amazon ECS quando crei o aggiorni un cluster. La crittografia a livello di cluster può essere utilizzata per crittografare tutti i volumi Amazon EBS collegati alle attività in esecuzione in un cluster specifico utilizzando la chiave KMS specificata a livello di cluster. Per ulteriori informazioni sulla configurazione della crittografia a livello di cluster, consulta il riferimento [ManagedStorageConfiguration](#) alle API di Amazon ECS.

Puoi configurare qualsiasi combinazione di queste chiavi. L'ordine di precedenza delle chiavi KMS è il seguente:

1. La chiave KMS specificata nella configurazione del volume. Quando si specifica una chiave KMS nella configurazione del volume, questa sostituisce l'impostazione predefinita di Amazon EBS e qualsiasi chiave KMS specificata a livello di cluster.
2. La chiave KMS specificata a livello di cluster. Quando si specifica una chiave KMS per la crittografia a livello di cluster dello storage gestito di Amazon ECS, questa sostituisce la crittografia predefinita di Amazon EBS ma non sostituisce alcuna chiave KMS specificata nella configurazione del volume.
3. Crittografia predefinita di Amazon EBS. La crittografia predefinita si applica quando non si specifica né una chiave KMS a livello di cluster né una chiave nella configurazione del volume. Se abiliti la crittografia Amazon EBS per impostazione predefinita, l'impostazione predefinita è la chiave KMS che specifichi per la crittografia per impostazione predefinita. Altrimenti, l'impostazione predefinita è Chiave gestita da AWS con l'alias. `alias/aws/ebs`

Note

Se lo `encrypted` imposti `false` nella configurazione del volume, non specifichi alcuna chiave KMS a livello di cluster e abiliti la crittografia Amazon EBS per impostazione predefinita, il volume verrà comunque crittografato con la chiave specificata per la crittografia Amazon EBS per impostazione predefinita.

Politica delle chiavi KMS gestita dal cliente

Per crittografare un volume EBS collegato all'attività utilizzando una chiave gestita dal cliente, è necessario configurare la politica delle chiavi KMS per garantire che il ruolo IAM utilizzato per la configurazione del volume disponga delle autorizzazioni necessarie per utilizzare la chiave. La policy chiave deve includere le autorizzazioni `kms:CreateGrant` `kms:GenerateDataKey*` Le `kms:ReEncryptFrom` autorizzazioni `kms:ReEncryptTo` e sono necessarie per crittografare i volumi creati utilizzando le istantanee. Se desideri configurare e crittografare solo nuovi volumi vuoti da allegare, puoi escludere le autorizzazioni `and. kms:ReEncryptTo kms:ReEncryptFrom`

Il seguente frammento di codice JSON mostra le principali dichiarazioni politiche che puoi allegare alla tua politica chiave KMS. L'utilizzo di queste istruzioni consentirà ad Amazon ECS di utilizzare la chiave per crittografare il volume EBS. Per utilizzare le dichiarazioni politiche di esempio, sostituiscile *user input placeholders* con le tue informazioni. Come sempre, configura solo le autorizzazioni di cui hai bisogno.

```
{
```

```

    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
    "Action": "kms:DescribeKey",
    "Resource": "*"
  },
  {
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
    "Action": [
      "kms:GenerateDataKey*",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom"
    ],
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:CallerAccount": "aws_account_id",
        "kms:ViaService": "ec2.region.amazonaws.com"
      },
      "ForAnyValue:StringEquals": {
        "kms:EncryptionContextKeys": "aws:ebs:id"
      }
    }
  },
  {
    "Effect": "Allow",
    "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
    "Action": "kms:CreateGrant",
    "Resource": "*",
    "Condition": {
      "StringEquals": {
        "kms:CallerAccount": "aws_account_id",
        "kms:ViaService": "ec2.region.amazonaws.com"
      },
      "ForAnyValue:StringEquals": {
        "kms:EncryptionContextKeys": "aws:ebs:id"
      },
      "Bool": {
        "kms:GrantIsForAWSResource": true
      }
    }
  }
}

```

Per ulteriori informazioni sulle politiche e le autorizzazioni chiave, consulta [Politiche chiave AWS KMS e AWS KMS autorizzazioni nella Guida per gli AWS KMS sviluppatori](#). Per la risoluzione dei problemi relativi agli allegati dei volumi EBS relativi alle autorizzazioni chiave, consulta. [Risoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS](#)

Specificare la configurazione del volume Amazon EBS nella distribuzione di Amazon ECS

Dopo aver registrato una definizione di attività con il `configuredAtLaunch` parametro impostato su `true`, puoi configurare un volume Amazon EBS durante la distribuzione quando esegui un'attività autonoma o quando crei o aggiorni un servizio.

Per configurare un volume, puoi utilizzare Amazon ECS APIs oppure puoi passare un file JSON come input per i seguenti comandi: AWS CLI

- [run-task](#) per eseguire un'attività ECS autonoma.
- [start-task](#) per eseguire un'attività ECS autonoma in un'istanza di contenitore specifica. Questo comando non è applicabile alle attività di tipo avvio di Fargate.
- [create-service](#) per creare un nuovo servizio ECS.
- [update-service](#) per aggiornare un servizio esistente.

Note

Affinché un contenitore della tua attività possa scrivere sul volume Amazon EBS montato, devi eseguire il contenitore come utente root.

Puoi anche configurare un volume Amazon EBS utilizzando AWS Management Console. Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#), [Creazione di un servizio Amazon ECS utilizzando la console](#) e [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Il seguente frammento di codice JSON mostra tutti i parametri di un volume Amazon EBS che può essere configurato al momento della distribuzione. Per utilizzare questi parametri per la configurazione del volume, sostituiscili *user input placeholders* con le tue informazioni. Per ulteriori informazioni su questi parametri, consulta [Configurazioni dei volumi](#).

```
"volumeConfigurations": [
```

```

{
  "name": "ebs-volume",
  "managedEBSVolume": {
    "encrypted": true,
    "kmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "volumeType": "gp3",
    "sizeInGiB": 10,
    "snapshotId": "snap-12345",
    "volumeInitializationRate": 100,
    "iops": 3000,
    "throughput": 125,
    "tagSpecifications": [
      {
        "resourceType": "volume",
        "tags": [
          {
            "key": "key1",
            "value": "value1"
          }
        ],
        "propagateTags": "NONE"
      }
    ],
    "roleArn": "arn:aws:iam::111122223333:role/ecsInfrastructureRole",
    "terminationPolicy": {
      "deleteOnTermination": true//can't be configured for service-
managed tasks, always true
    },
    "filesystemType": "ext4"
  }
}
]

```

Important

Assicurati che `volumeName` quello specificato nella configurazione sia lo stesso `volumeName` specificato nella definizione dell'attività.

Per informazioni sulla verifica dello stato del volume allegato, vedere [Risoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS](#). Per informazioni sul ruolo

dell'infrastruttura Amazon ECS AWS Identity and Access Management (IAM) necessario per l'allegato del volume EBS, consulta. [Ruolo IAM dell'infrastruttura Amazon ECS](#)

Di seguito sono riportati alcuni esempi di snippet JSON che mostrano la configurazione dei volumi Amazon EBS. Questi esempi possono essere utilizzati salvando gli snippet in file JSON e passando i file come parametri (utilizzando il parametro) per i comandi. `--cli-input-json file://filename` AWS CLI Sostituire *user input placeholders* con le proprie informazioni.

Configurare un volume per un'attività autonoma

Il seguente frammento mostra la sintassi per configurare i volumi Amazon EBS per l'allegato a un'attività autonoma. Il seguente frammento di codice JSON mostra la sintassi per la configurazione di, e impostazioni. `volumeType` `sizeInGiB` `encrypted` `kmsKeyId` La configurazione specificata nel file JSON viene utilizzata per creare e allegare un volume EBS all'attività autonoma.

```
{
  "cluster": "mycluster",
  "taskDefinition": "mytaskdef",
  "volumeConfigurations": [
    {
      "name": "datadir",
      "managedEBSVolume": {
        "volumeType": "gp3",
        "sizeInGiB": 100,
        "roleArn": "arn:aws:iam::1111222333:role/ecsInfrastructureRole",
        "encrypted": true,
        "kmsKeyId":
          "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    }
  ]
}
```

Configura un volume al momento della creazione del servizio

Il seguente frammento mostra la sintassi per configurare i volumi Amazon EBS per l'allegato alle attività gestite da un servizio. I volumi provengono dall'istantanea specificata utilizzando il `snapshotId` parametro a una velocità di 200 MiB/s. La configurazione specificata nel file JSON viene utilizzata per creare e allegare un volume EBS a ciascuna attività gestita dal servizio.

```
{
```

```

"cluster": "mycluster",
"taskDefinition": "mytaskdef",
"serviceName": "mysvc",
"desiredCount": 2,
"volumeConfigurations": [
  {
    "name": "myEbsVolume",
    "managedEBSVolume": {
      "roleArn": "arn:aws:iam::1111222333:role/ecsInfrastructureRole",
      "snapshotId": "snap-12345",
      "volumeInitializationRate": 200
    }
  }
]
}

```

Configura un volume durante l'aggiornamento del servizio

Il seguente frammento di codice JSON mostra la sintassi per l'aggiornamento di un servizio che in precedenza non aveva volumi Amazon EBS configurati per il collegamento alle attività. È necessario fornire l'ARN di una revisione della definizione di attività impostata `configuredAtLaunch` su `true`. Il seguente frammento di codice JSON mostra la sintassi per configurare le impostazioni `volumeType`, `sizeInGiB` e `throughput iops` `filesystemType`. Questa configurazione viene utilizzata per creare e collegare un volume EBS a ciascuna attività gestita dal servizio.

```

{
  "cluster": "mycluster",
  "taskDefinition": "mytaskdef",
  "service": "mysvc",
  "desiredCount": 2,
  "volumeConfigurations": [
    {
      "name": "myEbsVolume",
      "managedEBSVolume": {
        "roleArn": "arn:aws:iam::1111222333:role/ecsInfrastructureRole",
        "volumeType": "gp3",
        "sizeInGiB": 100,
        "iops": 3000,
        "throughput": 125,
        "filesystemType": "ext4"
      }
    }
  ]
}

```

```
    }  
  ]  
}
```

Configura un servizio per non utilizzare più i volumi Amazon EBS

Il seguente frammento di codice JSON mostra la sintassi per aggiornare un servizio in modo che non utilizzi più i volumi Amazon EBS. È necessario fornire l'ARN di una definizione di attività con `configuredAtLaunch` set to `false` o una definizione di attività senza il `configuredAtLaunch` parametro. È inoltre necessario fornire un `volumeConfigurations` oggetto vuoto.

```
{  
  "cluster": "mycluster",  
  "taskDefinition": "mytaskdef",  
  "service": "mysvc",  
  "desiredCount": 2,  
  "volumeConfigurations": []  
}
```

Politica di terminazione per i volumi Amazon EBS

Quando un'attività Amazon ECS termina, Amazon ECS utilizza il `deleteOnTermination` valore per determinare se il volume Amazon EBS associato all'attività terminata deve essere eliminato. Per impostazione predefinita, i volumi EBS collegati alle attività vengono eliminati quando l'attività viene terminata. Per le attività autonome, puoi modificare questa impostazione per preservare invece il volume al termine dell'attività.

Note

I volumi collegati alle attività gestite da un servizio non vengono conservati e vengono sempre eliminati al termine dell'attività.

Contrassegna i volumi Amazon EBS

Puoi etichettare i volumi Amazon EBS utilizzando l'`tagSpecifications` oggetto. Utilizzando l'oggetto, puoi fornire tag personalizzati e impostare la propagazione dei tag dalla definizione dell'attività o dal servizio, a seconda che il volume sia collegato a un'attività autonoma o a un'attività in un servizio. Il numero massimo di tag che è possibile allegare a un volume è 50.

⚠ Important

Amazon ECS allega automaticamente i tag AmazonECSManaged riservati a un volume Amazon EBS. AmazonECSCreated Ciò significa che puoi controllare l'allegato di un massimo di 48 tag aggiuntivi a un volume. Questi tag aggiuntivi possono essere tag definiti dall'utente, gestiti da ECS o propagati.

Se desideri aggiungere tag gestiti da Amazon ECS al tuo volume, devi `enableECSTags` impostarlo su `true` nella tua `UpdateService` chiamata `CreateService` `RunTask` o `StartTask`. Se attivi i tag gestiti da Amazon ECS, Amazon ECS taggherà automaticamente il volume con informazioni su cluster e servizi (`aws:ecs:clusterName`). `aws:ecs:serviceName` Per ulteriori informazioni sull'etichettatura delle risorse Amazon ECS, consulta [Tagging your Amazon ECS resources](#).

Il seguente frammento di codice JSON mostra la sintassi per etichettare ogni volume Amazon EBS collegato a ciascuna attività in un servizio con un tag definito dall'utente. Per utilizzare questo esempio per creare un servizio, sostituiscilo con le tue informazioni. *user input placeholders*

```
{
  "cluster": "mycluster",
  "taskDefinition": "mytaskdef",
  "serviceName": "mysvc",
  "desiredCount": 2,
  "enableECSTags": true,
  "volumeConfigurations": [
    {
      "name": "datadir",
      "managedEBSVolume": {
        "volumeType": "gp3",
        "sizeInGiB": 100,
        "tagSpecifications": [
          {
            "resourceType": "volume",
            "tags": [
              {
                "key": "key1",
                "value": "value1"
              }
            ]
          }
        ],
        "propagateTags": "NONE"
      }
    }
  ]
}
```



```

    }
  ],
  "roleArn": "arn:aws:iam:1111222333:role/ecsInfrastructureRole",
  "encrypted": true,
  "kmsKeyId":
    "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
]
}

```

Important

È necessario specificare un tipo di volume risorsa per etichettare i volumi Amazon EBS.

Prestazioni dei volumi Amazon EBS per le attività on-demand di Fargate

Il volume, gli IOPS e il throughput di base di Amazon EBS disponibili per un'attività on-demand di Fargate dipendono dal numero totale di unità CPU richieste per l'attività. Se richiedi 0,25, 0,5 o 1 unità CPU virtuale (vCPU) per l'attività Fargate, ti consigliamo di configurare un volume SSD per uso generico gp2 (gp3o) o un volume Hard Disk Drive (HDD) (o). st1 sc1 Se richiedi più di 1 vCPU per l'attività Fargate, i seguenti limiti prestazionali di base si applicano a un volume Amazon EBS collegato all'attività. È possibile ottenere temporaneamente prestazioni EBS superiori ai seguenti limiti. Tuttavia, ti consigliamo di pianificare il tuo carico di lavoro in base a questi limiti.

Unità CPU richieste (in vCPUs)	IOPS Amazon EBS di base (16 KiB di I/O)	Throughput di base di Amazon EBS (in, MiBps 128 KB di I/O)	Larghezza di banda di base (in Mbps)
2	3.000	75	360
4	5.000	120	1.150
8	10.000	250	2.300
16	15.000	500	4.500

 Note

Quando configuri un volume Amazon EBS per il collegamento a un'attività Fargate, il limite di prestazioni di Amazon EBS per l'attività Fargate viene condiviso tra lo storage temporaneo dell'attività e il volume allegato.

Prestazioni dei volumi Amazon EBS per le attività EC2

Amazon EBS offre tipi di volume, che si differenziano per caratteristiche prestazionali e prezzo, in modo da poter personalizzare le prestazioni e i costi di storage in base alle esigenze delle applicazioni. Per informazioni sulle prestazioni, inclusi IOPS per volume e throughput per volume, consulta i [tipi di volume di Amazon EBS](#) nella Amazon Elastic Block Store User Guide.

Risoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS

Potrebbe essere necessario risolvere i problemi o verificare il collegamento dei volumi Amazon EBS alle attività di Amazon ECS.

Verifica lo stato degli allegati al volume

Puoi utilizzare il AWS Management Console per visualizzare lo stato dell'allegato di un volume Amazon EBS a un'attività Amazon ECS. Se l'attività inizia e l'allegato non riesce, vedrai anche un motivo dello stato che puoi utilizzare per risolvere i problemi. Il volume creato verrà eliminato e l'operazione verrà interrotta. Per ulteriori informazioni sui motivi dello stato, vedere [Motivi dello stato dell'allegato del volume Amazon EBS alle attività di Amazon ECS](#).

Per visualizzare lo stato degli allegati di un volume e il motivo dello stato utilizzando la console

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella pagina Cluster, scegli il cluster in cui è in esecuzione l'attività. Viene visualizzata la pagina dei dettagli del cluster.
3. Nella pagina dei dettagli del cluster, scegli la scheda Attività.
4. Scegli l'attività per la quale desideri visualizzare lo stato degli allegati al volume. Potrebbe essere necessario utilizzare Filtra lo stato desiderato e scegliere Interrotto se l'attività che desideri esaminare è stata interrotta.
5. Nella pagina dei dettagli dell'attività, scegli la scheda Volumi. Potrai visualizzare lo stato degli allegati del volume Amazon EBS in Attachment status. Se il volume non riesce a collegarsi all'attività, puoi scegliere lo stato in Attachment status per visualizzare la causa dell'errore.

Puoi anche visualizzare lo stato degli allegati al volume di un'attività e il motivo dello stato associato utilizzando l'[DescribeTasks](#)API.

Guasti del servizio e delle attività

Potresti riscontrare errori di servizio o attività non specifici dei volumi Amazon EBS che possono influire sull'allegato del volume. Per ulteriori informazioni, consulta la pagina

- [Messaggi di evento relativi ai servizi](#)
- [Codici di errore delle attività interrotte](#)
- [Motivi di errore dell'API](#)

Motivi dello stato dell'allegato del volume Amazon EBS alle attività di Amazon ECS

Utilizza il seguente riferimento per risolvere i problemi che potresti riscontrare sotto forma di motivi di stato AWS Management Console durante la configurazione dei volumi Amazon EBS da allegare alle attività di Amazon ECS. Per ulteriori informazioni su come individuare questi motivi di status nella console, consulta. [Verifica lo stato degli allegati al volume](#)

ECS non è stato in grado di assumere il ruolo di infrastruttura ECS configurato 'arn:aws:iam: :role/ '. **111122223333 ecsInfrastructureRole** Verifica che il ruolo assegnato abbia un rapporto di fiducia adeguato con Amazon ECS

Questo motivo di status appare nei seguenti scenari.

- Fornisci un ruolo IAM senza la necessaria policy di fiducia allegata. Amazon ECS non può accedere al ruolo IAM dell'infrastruttura Amazon ECS fornito da te se il ruolo non dispone della politica di fiducia necessaria. L'attività può rimanere bloccata nello DEPROVISIONING stato. Per ulteriori informazioni sulla politica di fiducia necessaria, vedere [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- Il tuo utente IAM non è autorizzato a trasferire il ruolo dell'infrastruttura Amazon ECS ad Amazon ECS. L'attività può rimanere bloccata nello DEPROVISIONING stato. Per evitare questo problema, puoi allegare l'PassRole autorizzazione al tuo utente. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- Il tuo ruolo IAM non dispone delle autorizzazioni necessarie per gli allegati del volume Amazon EBS. L'attività può rimanere bloccata nello DEPROVISIONING stato. Per ulteriori informazioni sulle autorizzazioni specifiche necessarie per collegare i volumi Amazon EBS alle attività, consulta. [Ruolo IAM dell'infrastruttura Amazon ECS](#)

 Note

Potresti anche visualizzare questo messaggio di errore a causa di un ritardo nella propagazione dei ruoli. Se riprovando a utilizzare il ruolo dopo aver atteso qualche minuto il problema non risolve il problema, è possibile che tu abbia configurato erroneamente la politica di attendibilità per il ruolo.

ECS non è riuscito a configurare il volume EBS. Incontrato `IdempotentParameterMismatch` «; «Il token client che hai fornito è associato a una risorsa che è già stata eliminata. Utilizza un token client diverso.»

I seguenti scenari AWS KMS chiave possono portare alla `IdempotentParameterMismatch` visualizzazione di un messaggio:

- Specificate un ARN, un ID o un alias della chiave KMS che non è valido. In questo scenario, l'attività potrebbe sembrare avviata correttamente, ma alla fine l'attività non riesce perché AWS autentica la chiave KMS in modo asincrono. Per ulteriori informazioni, consulta la [crittografia Amazon EBS](#) nella Amazon EC2 User Guide.
- Fornisci una chiave gestita dal cliente priva delle autorizzazioni che consentono al ruolo IAM dell'infrastruttura Amazon ECS di utilizzare la chiave per la crittografia. Per evitare problemi di autorizzazione relativi alle policy chiave, consulta la policy AWS KMS chiave di esempio in [Data encryption for Amazon EBS](#) Volumes.

Puoi configurare Amazon EventBridge per inviare eventi di volume Amazon EBS e eventi di modifica dello stato delle attività di Amazon ECS a una destinazione, ad esempio gruppi Amazon CloudWatch . Puoi quindi utilizzare questi eventi per identificare lo specifico problema relativo alla chiave gestita dal cliente che ha influito sull'allegato del volume. Per ulteriori informazioni, consulta la pagina

- [Come posso creare un gruppo di CloudWatch log da utilizzare come destinazione per una EventBridge regola?](#) su AWS Re:post.
- Eventi di [modifica dello stato dell'attività](#).
- [EventBridge Eventi Amazon per Amazon EBS nella Guida](#) per l'utente di Amazon EBS.

ECS si è verificato un timeout durante la configurazione dell'allegato del volume EBS al tuo Task.

I seguenti scenari di formato del file system generano questo messaggio.

- Il formato del file system specificato durante la configurazione non è compatibile con il [sistema operativo dell'attività](#).

- Si configura un volume Amazon EBS da creare da uno snapshot e il formato del file system dello snapshot non è compatibile con il sistema operativo dell'operazione. Per i volumi creati da uno snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dello snapshot.

Puoi utilizzare i log degli agenti container di Amazon ECS per risolvere questo messaggio per attività di tipo avvio di Amazon. EC2 Per ulteriori informazioni, consulta le [posizioni dei file di registro di Amazon ECS](#) e [Amazon ECS log collector](#).

Usa i volumi Amazon EFS con Amazon ECS

Amazon Elastic File System (Amazon EFS) offre archiviazione file semplice e scalabile da utilizzare con i processi Amazon ECS. Con Amazon EFS, la capacità di storage è elastica. La capacità di storage aumenta e si riduce automaticamente quando si aggiungono e si rimuovono i file. Le tue applicazioni possono disporre dello storage di cui hanno bisogno nel momento in cui ne hanno bisogno.

Puoi utilizzare i file system Amazon EFS con Amazon ECS per esportare i dati del file system all'interno del tuo parco istanze di container. In questo modo, le tue attività hanno accesso allo stesso storage persistente, indipendentemente dall'istanza in cui si trovano. Inoltre, per utilizzare il file system, le tue definizioni di attività devono fare riferimento ai montaggi di volume inerenti l'istanza di container.

Per un tutorial, vedere [Configurazione dei file system Amazon EFS per Amazon ECS tramite la console](#).

Considerazioni

Quando usi i volumi Amazon EFS, tieni presente quanto segue:

- Per le attività che utilizzano il tipo di EC2 avvio, il supporto del file system Amazon EFS è stato aggiunto come anteprima pubblica con la versione AMI ottimizzata per Amazon ECS 20191212 con agente container versione 1.35.0. Tuttavia, il supporto del file system Amazon EFS è entrato nella disponibilità generale con l'AMI ottimizzata per Amazon ECS versione 20200319 con l'agente del container versione 1.38.0, che conteneva il punto di accesso Amazon EFS e le funzionalità di autorizzazione IAM. Per utilizzare queste funzionalità, ti consigliamo di utilizzare la versione AMI ottimizzata per Amazon ECS 20200319 o versioni successive. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

 Note

Se crei la tua AMI, devi utilizzare l'agente container 1.38.0 o successiva, `ecs-init` versione 1.38.0-1 o successiva ed eseguire i seguenti comandi sull'istanza Amazon EC2 per abilitare il plugin di volume Amazon ECS. I comandi dipendono dal fatto che si stia usando Amazon Linux 2 o Amazon Linux come immagine di base.

Amazon Linux 2

```
yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
yum install amazon-efs-utils
sudo shutdown -r now
```

- Per le attività ospitate su Fargate, i file system Amazon EFS sono supportati sulla piattaforma versione 1.4.0 o successiva (Linux). Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).
- Quando utilizzi volumi Amazon EFS per attività ospitate su Fargate, Fargate crea un container supervisor responsabile della gestione del volume Amazon EFS. Il contenitore supervisor utilizza una piccola quantità di memoria e CPU dell'attività. Il container supervisor è visibile quando si sottopone a query l'endpoint dei metadati dell'attività versione 4. Inoltre, è visibile in CloudWatch Container Insights come nome `aws-fargate-supervisor` del contenitore. Per ulteriori informazioni sull'utilizzo del tipo di EC2 avvio Amazon, consulta [Endpoint di metadati delle attività Amazon ECS versione 4](#). Per ulteriori informazioni sull'utilizzo del tipo di lancio Fargate, vedere [Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate](#).
- L'uso di volumi Amazon EFS o la specifica di una `EFSVolumeConfiguration` non sono supportati su istanze esterne.
- Consigliamo di impostare il parametro `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION` nel file di configurazione dell'agente su un valore inferiore a quello predefinito (circa 1 ora). Questa modifica aiuta a prevenire la scadenza delle credenziali di montaggio EFS e consente la pulizia dei supporti che non sono in uso. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Usa i punti di accesso Amazon EFS

I punti di accesso Amazon EFS sono punti di accesso specifici dell'applicazione in un file system EFS per la gestione dell'accesso dell'applicazione ai set di dati condivisi. Per ulteriori informazioni sui punti di accesso Amazon EFS e su come controllare l'accesso a tali punti, consulta [Working with Amazon EFS Access Points](#) (Utilizzo dei punti di accesso Amazon EFS) nella Guida per l'utente di Amazon Elastic File System.

I punti di accesso possono applicare un'identità utente, inclusi i gruppi dell'utente POSIX, per tutte le richieste al file system effettuate tramite il punto di accesso. I punti di accesso possono inoltre applicare una directory root diversa per il file system. In questo modo i client possono accedere solo ai dati nella directory specificata o nelle relative sottodirectory.

Note

Quando crei un punto di accesso EFS, è necessario specificare un percorso nel file system da utilizzare come directory root. Quando si fa riferimento al file system EFS con un ID punto di accesso nella definizione di attività Amazon ECS, la directory root deve essere omessa o impostata su / in modo da forzare il percorso impostato sul punto di accesso EFS.

Puoi utilizzare un ruolo IAM del processo Amazon ECS per imporre che applicazioni specifiche utilizzino un punto di accesso specifico. Combinando le policy IAM con i punti di accesso, puoi fornire accesso sicuro a set di dati specifici per le applicazioni. Per ulteriori informazioni su come utilizzare i ruoli IAM dell'attività, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Best practice per l'utilizzo dei volumi Amazon EFS con Amazon ECS

Prendi nota dei seguenti consigli sulle best practice quando usi Amazon EFS con Amazon ECS.

Sicurezza e controlli di accesso per i volumi Amazon EFS

Amazon EFS offre funzionalità di controllo degli accessi che puoi utilizzare per garantire che i dati archiviati in un file system Amazon EFS siano sicuri e accessibili solo dalle applicazioni che ne hanno bisogno. Puoi proteggere i dati abilitando la crittografia a riposo e in transito. Per ulteriori informazioni, consulta [Crittografia dati in Amazon EFS](#) nella Amazon Elastic File System User Guide.

Oltre alla crittografia dei dati, puoi anche utilizzare Amazon EFS per limitare l'accesso a un file system. Esistono tre modi per implementare il controllo degli accessi in EFS.

- **Gruppi di sicurezza:** con gli obiettivi di montaggio di Amazon EFS, puoi configurare un gruppo di sicurezza utilizzato per consentire e negare il traffico di rete. Puoi configurare il gruppo di sicurezza collegato ad Amazon EFS per consentire il traffico NFS (porta 2049) dal gruppo di sicurezza collegato alle tue istanze Amazon ECS o, quando utilizzi la modalità di aws-vpc rete, il task Amazon ECS.
- **IAM:** puoi limitare l'accesso a un file system Amazon EFS utilizzando IAM. Una volta configurate, le attività di Amazon ECS richiedono un ruolo IAM per l'accesso al file system per montare un file system EFS. Per ulteriori informazioni, consulta [Using IAM per controllare l'accesso ai dati del file system](#) nella Amazon Elastic File System User Guide.

Le policy IAM possono anche applicare condizioni predefinite, come richiedere a un client di utilizzare TLS durante la connessione a un file system Amazon EFS. Per ulteriori informazioni, consulta le [chiavi di condizione di Amazon EFS per i client](#) nella Amazon Elastic File System User Guide.

- **Punti di accesso Amazon EFS:** i punti di accesso Amazon EFS sono punti di ingresso specifici dell'applicazione in un file system Amazon EFS. Puoi utilizzare i punti di accesso per applicare l'identità di un utente, inclusi i gruppi POSIX dell'utente, per tutte le richieste del file system effettuate tramite il punto di accesso. I punti di accesso possono inoltre applicare una directory root diversa per il file system. In questo modo i client possono accedere solo ai dati nella directory specificata o nelle sue sottodirectory.

Policy IAM

Puoi utilizzare le policy IAM per controllare l'accesso al file system Amazon EFS.

È possibile specificare le seguenti operazioni per i client in un file system utilizzando una policy del file system.

Azione	Descrizione
<code>elasticfilesystem:ClientMount</code>	Fornisce un accesso in sola lettura a un file system.
<code>elasticfilesystem:ClientWrite</code>	Fornisce le autorizzazioni di scrittura su un file system.

Azione	Descrizione
<code>elasticfilesystem:ClientRootAccess</code>	Fornisce la possibilità di utilizzare l'utente root quando si accede a un file system.

È necessario specificare ogni azione in una policy. Le politiche possono essere definite nei seguenti modi:

- Basato sul client: associa la policy al ruolo dell'attività

Imposta l'opzione di autorizzazione IAM quando crei la definizione dell'attività.

- Basato sulle risorse: collega la policy al file system Amazon EFS

Se la politica basata sulle risorse non esiste, per impostazione predefinita al momento della creazione del file system l'accesso è concesso a tutti i principali (*).

Quando imposti l'opzione di autorizzazione IAM, uniamo la policy associata al task role e quella basata sulle risorse Amazon EFS. L'opzione di autorizzazione IAM trasmette l'identità dell'attività (il ruolo dell'attività) con la policy ad Amazon EFS. Ciò consente alla policy basata sulle risorse di Amazon EFS di avere un contesto per l'utente o il ruolo IAM specificato nella policy. Se non imposti l'opzione, la policy a livello di risorsa di Amazon EFS identifica l'utente IAM come «anonimo».

Prendi in considerazione l'implementazione di tutti e tre i controlli di accesso su un file system Amazon EFS per la massima sicurezza. Ad esempio, puoi configurare il gruppo di sicurezza collegato a un punto di montaggio Amazon EFS per consentire solo il traffico NFS in ingresso da un gruppo di sicurezza associato all'istanza del contenitore o all'attività Amazon ECS. Inoltre, puoi configurare Amazon EFS per richiedere un ruolo IAM per accedere al file system, anche se la connessione proviene da un gruppo di sicurezza consentito. Infine, puoi utilizzare gli access point Amazon EFS per applicare le autorizzazioni utente POSIX e specificare le directory root per le applicazioni.

Il seguente frammento di definizione delle attività mostra come montare un file system Amazon EFS utilizzando un punto di accesso.

```
"volumes": [
  {
    "efsVolumeConfiguration": {
      "fileSystemId": "fs-1234",
      "authorizationConfig": {
```

```
    "accessPointId": "fsap-1234",  
    "iam": "ENABLED"  
  },  
  "transitEncryption": "ENABLED",  
  "rootDirectory": ""  
},  
"name": "my-fileSystem"  
}  
]
```

Prestazioni dei volumi Amazon EFS

Amazon EFS offre due modalità di prestazioni: General Purpose e Max I/O. General Purpose è adatto per applicazioni sensibili alla latenza come i sistemi di gestione dei contenuti e i CI/CD tools. In contrast, Max I/O file system sono adatti per carichi di lavoro come analisi dei dati, elaborazione multimediale e apprendimento automatico. Questi carichi di lavoro devono eseguire operazioni parallele da centinaia o addirittura migliaia di container e richiedono il throughput aggregato e gli IOPS più elevati possibili. Per ulteriori informazioni, consulta le [modalità di prestazione di Amazon EFS](#) nella Amazon Elastic File System User Guide.

Alcuni carichi di lavoro sensibili alla latenza richiedono sia i livelli di I/O più elevati forniti dalla modalità di prestazioni Max I/O sia la latenza più bassa fornita dalla modalità prestazionale General Purpose. Per questo tipo di carico di lavoro, consigliamo di creare più file system in modalità prestazionale per uso generico. In questo modo, è possibile distribuire il carico di lavoro delle applicazioni su tutti questi file system, purché il carico di lavoro e le applicazioni siano in grado di supportarlo.

Velocità effettiva dei volumi Amazon EFS

Tutti i file system Amazon EFS hanno un throughput misurato associato determinato dalla quantità di throughput assegnato per i file system che utilizzano Provisioned Throughput o dalla quantità di dati archiviati nella classe di storage EFS Standard o One Zone per i file system che utilizzano Bursting Throughput. Per ulteriori informazioni, consulta [Understanding metered throughput](#) nella Amazon Elastic File System User Guide.

La modalità di throughput predefinita per i file system Amazon EFS è la modalità bursting. Con la modalità bursting, la velocità effettiva disponibile per un file system aumenta o diminuisce man mano che il file system cresce. Poiché i carichi di lavoro basati su file in genere registrano picchi, richiedendo livelli di throughput elevati per periodi di tempo e livelli di throughput inferiori per il resto del tempo, Amazon EFS è progettato per consentire livelli di throughput elevati per periodi di tempo.

Inoltre, poiché molti carichi di lavoro richiedono un elevato livello di lettura, le operazioni di lettura vengono misurate con un rapporto di 1:3 rispetto ad altre operazioni NFS (come la scrittura).

Tutti i file system Amazon EFS offrono prestazioni di base costanti di 50 MB/s for each TB of Amazon EFS Standard or Amazon EFS One Zone storage. All file systems (regardless of size) can burst to 100 MB/s. File systems with more than 1TB of EFS Standard or EFS One Zone storage can burst to 100 MB/s for each TB. Because read operations are metered at a 1:3 ratio, you can drive up to 300 MiBs/s per ogni TIB di throughput di lettura. Man mano che aggiungi dati al file system, il throughput massimo disponibile per il file system si ridimensiona in modo lineare e automatico con lo storage nella classe di storage Amazon EFS Standard. Se hai bisogno di un throughput superiore a quello che puoi ottenere con la quantità di dati archiviati, puoi configurare Provisioned Throughput sulla quantità specifica richiesta dal tuo carico di lavoro.

La velocità effettiva del file system è condivisa tra tutte le EC2 istanze Amazon connesse a un file system. Ad esempio, un file system da 1 TB che può arrivare a 100. MB/s of throughput can drive 100 MB/s from a single Amazon EC2 instance can each drive 10 MB/s Per ulteriori informazioni, consulta le [prestazioni di Amazon EFS](#) nella Amazon Elastic File System User Guide.

Ottimizzazione dei costi per i volumi Amazon EFS

Amazon EFS semplifica la scalabilità dello storage per te. I file system Amazon EFS crescono automaticamente man mano che aggiungi più dati. Soprattutto con la modalità Amazon EFS Bursting Throughput, la velocità effettiva su Amazon EFS si ridimensiona all'aumentare delle dimensioni del file system nella classe di storage standard. Per migliorare la velocità effettiva senza pagare costi aggiuntivi per la velocità effettiva assegnata su un file system EFS, puoi condividere un file system Amazon EFS con più applicazioni. Utilizzando i punti di accesso Amazon EFS, puoi implementare l'isolamento dello storage in file system Amazon EFS condivisi. In questo modo, anche se le applicazioni condividono ancora lo stesso file system, non possono accedere ai dati a meno che tu non li autorizzi.

Man mano che i dati crescono, Amazon EFS ti aiuta a spostare automaticamente i file a cui si accede raramente in una classe di storage inferiore. La classe di storage Amazon EFS Standard-Infrequent Access (IA) riduce i costi di storage per i file a cui non si accede ogni giorno. Lo fa senza sacrificare l'elevata disponibilità, l'elevata durabilità, l'elasticità e l'accesso al file system POSIX forniti da Amazon EFS. Per ulteriori informazioni, consulta le [classi di storage EFS](#) nella Amazon Elastic File System User Guide.

Prendi in considerazione l'utilizzo delle policy del ciclo di vita di Amazon EFS per risparmiare automaticamente denaro spostando i file a cui si accede raramente nello storage Amazon EFS IA.

Per ulteriori informazioni, consulta [Gestione del ciclo di vita EFS](#) nella Guida per l'utente di Amazon Elastic File System.

Quando crei un file system Amazon EFS, puoi scegliere se Amazon EFS replica i dati su più zone di disponibilità (Standard) o archivarli in modo ridondante all'interno di un'unica zona di disponibilità. La classe di storage Amazon EFS One Zone può ridurre i costi di storage di un margine significativo rispetto alle classi di storage Amazon EFS Standard. Prendi in considerazione l'utilizzo della classe di storage Amazon EFS One Zone per carichi di lavoro che non richiedono resilienza Multi-AZ. Puoi ridurre ulteriormente il costo dello storage Amazon EFS One Zone spostando i file a cui si accede raramente su Amazon EFS One Zone-Infrequent Access. Per ulteriori informazioni, consulta la sezione [Amazon EFS Infrequent Access](#).

Protezione dei dati dei volumi Amazon EFS

Amazon EFS archivia i dati in modo ridondante su più zone di disponibilità per file system che utilizzano classi di storage Standard. Se selezioni le classi di storage Amazon EFS One Zone, i tuoi dati vengono archiviati in modo ridondante all'interno di un'unica zona di disponibilità. Inoltre, Amazon EFS è progettato per fornire il 99,99999% (11 9) di durabilità in un determinato anno.

Come per qualsiasi ambiente, è consigliabile disporre di un backup e creare protezioni contro l'eliminazione accidentale. Per i dati di Amazon EFS, tale best practice include un backup funzionante e regolarmente testato utilizzando AWS Backup. I file system che utilizzano le classi di storage Amazon EFS One Zone sono configurati per eseguire automaticamente il backup dei file per impostazione predefinita al momento della creazione del file system, a meno che non si scelga di disabilitare questa funzionalità. Per ulteriori informazioni, consulta [Backup up file system EFS](#) nella Amazon Elastic File System User Guide.

Specificare un file system Amazon EFS in una definizione di attività Amazon ECS

Per utilizzare i volumi del file system Amazon EFS per i container, è necessario specificare le configurazioni di volume e punto di montaggio nella definizione di attività. Il seguente frammento JSON della definizione di attività illustra la sintassi degli oggetti `volumes` e `mountPoints` per un container.

```
{
  "containerDefinitions": [
    {
      "name": "container-using-efs",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
      "entryPoint": [
```

```

        "sh",
        "-c"
    ],
    "command": [
        "ls -la /mount/efs"
    ],
    "mountPoints": [
        {
            "sourceVolume": "myEfsVolume",
            "containerPath": "/mount/efs",
            "readOnly": true
        }
    ]
},
"volumes": [
    {
        "name": "myEfsVolume",
        "efsVolumeConfiguration": {
            "fileSystemId": "fs-1234",
            "rootDirectory": "/path/to/my/data",
            "transitEncryption": "ENABLED",
            "transitEncryptionPort": integer,
            "authorizationConfig": {
                "accessPointId": "fsap-1234",
                "iam": "ENABLED"
            }
        }
    }
]
}

```

efsVolumeConfiguration

Tipo: oggetto

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

fileSystemId

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

Tipo: string

Campo obbligatorio: no

La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omissso, viene utilizzata la radice del volume Amazon EFS. La specifica di `/` avrà lo stesso effetto dell'omissione di questo parametro.

 Important

Se un punto di accesso EFS è specificato in `authorizationConfig`, il parametro della directory root deve essere omissso o impostato su `/` per applicare il percorso impostato sul punto di accesso EFS.

`transitEncryption`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se abilitare o meno la crittografia per i dati Amazon EFS in transito tra l'host Amazon ECS e il server Amazon EFS. Se si utilizza l'autorizzazione IAM di Amazon EFS, è necessario abilitare la crittografia di transito. Se questo parametro viene omissso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni, consulta [Crittografia dei dati in transito](#) nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: integer

Campo obbligatorio: no

La porta da utilizzare per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non si specifica una porta di crittografia di transito, verrà utilizzata la strategia di

selezione della porta utilizzata dall'helper per il montaggio di Amazon EFS. Per ulteriori informazioni, consulta [Assistente per il montaggio di EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: oggetto

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

Tipo: string

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory root specificato in `efsVolumeConfiguration` deve essere omesso o impostato su `/` per applicare il percorso impostato sul punto di accesso EFS. Se si utilizza un punto di accesso, la crittografia di transito deve essere abilitata in `EFSTransitEncryptionConfiguration`. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Amazon EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`iam`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se utilizzare o meno il ruolo IAM dell'attività Amazon ECS riportato in una definizione di attività durante il montaggio del file system Amazon EFS. Se abilitato, la crittografia di transito deve essere abilitata nella casella `EFSTransitEncryptionConfiguration`. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni consulta [Ruoli IAM per le attività](#).

Configurazione dei file system Amazon EFS per Amazon ECS tramite la console

Scopri come usare i file system Amazon Elastic File System (Amazon EFS) con Amazon ECS.

Fase 1: Creazione di un cluster Amazon ECS

Utilizza la procedura seguente per creare un cluster Amazon ECS.

Creazione di un nuovo cluster (console Amazon ECS)

Prima di iniziare, assegna l'autorizzazione IAM appropriata. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#).

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Configurazione del cluster, per Nome cluster, inserisci `EFS-tutorial` come nome del cluster.
6. (Facoltativo) Per modificare il VPC e le sottoreti in cui vengono avviati i processi e i servizi, in Networking (Reti), esegui una qualunque di queste operazioni:
 - Per rimuovere una sottorete, in Subnets (Sottoreti), scegli X per ogni sottorete da rimuovere.
 - Per passare a un VPC diverso da quello di default, in VPC, scegli un VPC esistente, poi in Subnets (Sottoreti), seleziona ciascuna sottorete.
7. Per aggiungere EC2 istanze Amazon al tuo cluster, espandi l'infrastruttura, quindi seleziona le EC2 istanze Amazon. Successivamente, configura il gruppo Auto Scaling che funge da provider di capacità:
 - Per creare un gruppo Auto Scaling, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona Create new group (Crea nuovo gruppo) e quindi fornisci i seguenti dettagli sul gruppo:
 - Per Sistema operativo/architettura, seleziona Amazon Linux 2.
 - Ad EC2 esempio, scegli il tipo. `t2.micro`
 - In SSH key pair (Coppia di chiavi SSH), scegli la coppia che dimostra la tua identità quando ti connetti all'istanza.
 - In Capacità, inserisci 1.
8. Scegli Create (Crea).

Fase 2: creare un gruppo di sicurezza per EC2 le istanze Amazon e il file system Amazon EFS

In questa fase, crei un gruppo di sicurezza per le tue EC2 istanze Amazon che consente il traffico di rete in entrata sulla porta 80 e il tuo file system Amazon EFS che consente l'accesso in entrata dalle istanze di container.

Crea un gruppo di sicurezza per le tue EC2 istanze Amazon con le seguenti opzioni:

- Nome gruppo di sicurezza: immetti un nome univoco per il gruppo di sicurezza.
- VPC: il VPC identificato in precedenza per il cluster.
- Regola in entrata
 - Tipo: HTTP
 - Origine: 0.0.0.0/0.

Crea un gruppo di sicurezza per il file system Amazon EFS con le seguenti opzioni:

- Nome gruppo di sicurezza: immetti un nome univoco per il gruppo di sicurezza. Ad esempio, EFS-access-for-sg-*dc025fa2*.
- VPC: il VPC identificato in precedenza per il cluster.
- Regola in entrata
 - Tipo: NFS
 - Origine: personalizzato con l'ID del gruppo di sicurezza creato per le istanze.

Per informazioni su come creare un gruppo di sicurezza, consulta [Creare un gruppo di sicurezza per la tua EC2 istanza Amazon](#) nella Amazon EC2 User Guide.

Fase 3: Creazione di un file system Amazon EFS

In questa fase, viene creato un file system Amazon EFS.

Come creare un file system Amazon EFS per i processi di Amazon ECS

1. Apri la console Amazon Elastic File System all'indirizzo <https://console.aws.amazon.com/efs/>.
2. Scegliere Create file system (Crea file system).
3. Inserisci un nome per il file system, quindi scegli il VPC in cui sono ospitate le istanze di container. Di default, ciascuna sottorete nel VPC specificato riceve un target di montaggio che utilizza il gruppo di sicurezza predefinito per tale VPC. Quindi, scegli Personalizza.

 Note

Questo tutorial presuppone che il file system Amazon EFS, il cluster Amazon ECS, le istanze di container e le attività si trovino nello stesso VPC. Per ulteriori informazioni sul montaggio di un file system da un VPC diverso, consulta la [procedura dettagliata: montare un file system da un VPC diverso nella Amazon EFS User Guide](#).

4. Nella pagina Impostazioni del file system, configura le impostazioni facoltative, quindi in Impostazioni delle prestazioni, scegli Ottimizzazione come modalità di velocità di trasmissione effettiva per il file system. Dopo aver configurato le impostazioni, seleziona Avanti.
 - a. (Opzionale) Aggiungi tag al tuo file system. Ad esempio, puoi specificare un nome di file system univoco inserendolo nella colonna Value (Valore) accanto alla chiave Name (Nome).
 - b. (Facoltativo) Abilita la gestione del ciclo di vita per risparmiare denaro in storage con accessi non frequenti. Per ulteriori informazioni, consulta [Gestione del ciclo di vita EFS](#) nella Amazon Elastic File System User Guide.
 - c. (Facoltativo) Abilita la crittografia. Seleziona la casella di controllo per abilitare la crittografia del file system Amazon EFS inattivo.
5. Nella pagina Accesso di rete, in Target di montaggio, sostituisci la configurazione del gruppo di sicurezza esistente per ogni zona di disponibilità con il gruppo di sicurezza creato per il file system in [Fase 2: creare un gruppo di sicurezza per EC2 le istanze Amazon e il file system Amazon EFS](#), quindi scegli Avanti.
6. Non è necessario configurare la Policy del file system per questo tutorial, quindi puoi saltare la sezione scegliendo Avanti.
7. Esamina le opzioni del file system e scegli Crea per completare il processo.
8. Dalla schermata File system, registra ID del file system. Nel passaggio successivo, si farà riferimento a questo valore nella definizione di attività di Amazon ECS.

Fase 4: Aggiunta di contenuti al file system Amazon EFS

In questa fase, monti il file system Amazon EFS su un' EC2 istanza Amazon e aggiungi contenuti ad essa. Questo è per scopi di test in questo tutorial, per illustrare la natura persistente dei dati. Quando si utilizza questa funzione è necessario disporre della propria applicazione o di un altro metodo di scrittura dei dati nel file system Amazon EFS.

Per creare un' EC2 istanza Amazon e montare il file system Amazon EFS

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Scegliere Launch Instance (Avvia istanza).
3. In Immagini di applicazioni e sistema operativo (Amazon Machine Image), seleziona AMI (HVM) di Amazon Linux 2.
4. In Tipo di istanza, mantieni il tipo di istanza predefinito t2.micro.
5. In Coppia di chiavi (registrazione), seleziona una coppia di chiavi per l'accesso SSH all'istanza.
6. In Impostazioni di rete, seleziona il VPC specificato per il file system Amazon EFS e il cluster Amazon ECS. Seleziona una sottorete e il gruppo di sicurezza dell'istanza creato in [Fase 2: creare un gruppo di sicurezza per EC2 le istanze Amazon e il file system Amazon EFS](#). Configura il gruppo di sicurezza dell'istanza. Assicurati che l'opzione Assegna automaticamente IP pubblico sia abilitata.
7. In Configura storage, scegli il pulsante Modifica per i file system, quindi scegli EFS. Seleziona il file system creato in [Fase 3: Creazione di un file system Amazon EFS](#). Puoi modificare facoltativamente il punto di montaggio o lasciare il valore predefinito.

 Important

Devi selezionare una sottorete prima di poter aggiungere un file system all'istanza.

8. Deseleziona l'opzione Crea e allega automaticamente i gruppi di sicurezza. Lascia selezionata l'altra casella di controllo. Scegli Add shared file system (Aggiungi file system condiviso).
9. In Dettagli avanzati, assicurati che lo script dati utente venga popolato automaticamente con le fasi di montaggio del file system Amazon EFS.
10. In Riepilogo, assicurati che Numero di istanze sia 1. Scegliere Launch Instance (Avvia istanza).
11. Nella pagina Avvia un'istanza, scegli Visualizza tutte le istanze per visualizzare lo stato delle istanze. Inizialmente, Stato dell'istanza è PENDING. Quando lo stato cambia in RUNNING e l'istanza supera tutti i controlli di stato, l'istanza è pronta per l'uso.

Ora ti connetti all' EC2 istanza Amazon e aggiungi contenuti al file system Amazon EFS.

Per connettersi all' EC2 istanza Amazon e aggiungere contenuti al file system Amazon EFS

1. SSH verso l' EC2 istanza Amazon che hai creato. Per ulteriori informazioni, consulta [Connect to your Linux using SSH](#) nella Amazon EC2 User Guide.

2. Dalla finestra del terminale, esegui il comando `df -T` per verificare che il file system Amazon EFS sia montato. Nel seguente output, abbiamo evidenziato il montaggio del file system Amazon EFS.

```
$ df -T
Filesystem      Type          1K-blocks    Used        Available Use% Mounted on
devtmpfs        devtmpfs      485468        0           485468    0% /dev
tmpfs           tmpfs         503480        0           503480    0% /dev/shm
tmpfs           tmpfs         503480        424         503056    1% /run
tmpfs           tmpfs         503480        0           503480    0% /sys/fs/
cgroup
/dev/xvda1      xfs           8376300 1310952        7065348   16% /
127.0.0.1:/     nfs4          9007199254739968 0 9007199254739968 0% /mnt/efs/fs1
tmpfs           tmpfs         100700        0           100700    0% /run/
user/1000
```

3. Passa alla directory in cui è montato il file system Amazon EFS. Nell'esempio precedente è `/mnt/efs/fs1`.
4. Crea un file denominato `index.html` con i seguenti contenuti:

```
<html>
  <body>
    <h1>It Works!</h1>
    <p>You are using an Amazon EFS file system for persistent container
storage.</p>
  </body>
</html>
```

Fase 5: creazione di una definizione di attività

La seguente definizione di attività crea un volume di dati denominato `efs-html`. Il container `nginx` monta il volume di dati host nella radice NGINX, `/usr/share/nginx/html`.

Per creare una nuova definizione di attività utilizzando la console Amazon ECS

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).

4. Nella casella dell'editor JSON, copia e incolla il seguente testo JSON, sostituendo `fileSystemId` con l'ID del file system Amazon EFS.

```
{
  "containerDefinitions": [
    {
      "memory": 128,
      "portMappings": [
        {
          "hostPort": 80,
          "containerPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "mountPoints": [
        {
          "containerPath": "/usr/share/nginx/html",
          "sourceVolume": "efs-html"
        }
      ],
      "name": "nginx",
      "image": "public.ecr.aws/docker/library/nginx:latest"
    }
  ],
  "volumes": [
    {
      "name": "efs-html",
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1324abcd",
        "transitEncryption": "ENABLED"
      }
    }
  ],
  "family": "efs-tutorial",
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole"
}
```

Note

Puoi aggiungere le seguenti autorizzazioni al tuo ruolo IAM di esecuzione delle attività di Amazon ECS per consentire all'agente Amazon ECS di individuare e montare un file system Amazon EFS su un'attività all'avvio.

- `elasticfilesystem:ClientMount`
- `elasticfilesystem:ClientWrite`
- `elasticfilesystem:DescribeMountTargets`
- `elasticfilesystem:DescribeFileSystems`

5. Scegli Create (Crea).

Fase 6: Esecuzione di un processo e visualizzazione dei risultati

Ora che il file system Amazon EFS è stato creato e che è presente del contenuto Web per il container NGINX da servire, è possibile eseguire un processo utilizzando la definizione di attività creata. I server Web NGINX gestiscono la pagina HTML semplice. Se aggiorni il contenuto del tuo file system Amazon EFS, le modifiche vengono propagate a qualsiasi container in cui è montato lo stesso file system.

L'attività viene eseguita nella sottorete definita per il cluster.

Esecuzione di un processo e visualizzazione dei risultati attraverso la console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Cluster, seleziona il cluster per eseguire il processo autonomo contenuto.

Determina la risorsa da cui avviare il servizio.

Per avviare un servizio da	Fasi	
Cluster	a. Nella pagina Cluster, seleziona il cluster in cui creare il servizio.	

Per avviare un servizio da	Fasi	
	b. Dalla scheda Processi, scegli Esegui nuovo processo.	
Tipo di avvio	a. Nella pagina Task (Attività), scegli la definizione di attività. b. Se è presente più di una revisione, selezionane una. c. Scegli Create (Crea), Run task (Esegui attività).	

3. (Facoltativo) Scegli come viene distribuita l'attività pianificata nell'infrastruttura cluster. Espandi Compute configuration (Configurazione di calcolo), quindi effettua le seguenti operazioni:

Metodo di distribuzione	Fasi	
Tipo di avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Tipo di avvio, scegli EC2.	

4. Per Tipo di applicazione, scegli Processo.
5. Per Definizione di attività, seleziona la definizione dell'attività `efs-tutorial` che hai creato in precedenza.
6. Per Attività desiderate, inserisci 1.
7. Scegli Create (Crea).
8. Nella pagina Cluster, seleziona Infrastruttura.
9. In Istanze di container, seleziona l'istanza di container a cui effettuare la connessione.
10. Nella pagina Istanza di container, nella sezione Reti, registra l'IP pubblico per l'istanza.

11. Apri un browser e inserisci l'indirizzo IP pubblico. Dovresti visualizzare un messaggio simile al seguente:

It works!
You are using an Amazon EFS file system for persistent container storage.

 Note

In caso contrario, assicurati che il gruppo di sicurezza per l'istanza di container consenta il traffico di rete in entrata sulla porta 80 e che il gruppo di sicurezza per il file system consenta l'accesso in entrata dall'istanza di container.

Utilizzo FSx per volumi Windows File Server con Amazon ECS

FSx per Windows File Server fornisce file server Windows completamente gestiti, supportati da un file system Windows. Quando si utilizza FSx for Windows File Server insieme a ECS, è possibile effettuare il provisioning delle attività di Windows con uno storage di file statico persistente, distribuito, condiviso. Per ulteriori informazioni, consulta [What Is FSx for Windows File Server?](#).

 Note

EC2 le istanze che utilizzano l'AMI Windows Server 2016 Full ottimizzata FSx per Amazon ECS non supportano i volumi di attività ECS di Windows File Server. Non è possibile utilizzarlo FSx per i volumi di Windows File Server in un contenitore Windows nella configurazione Fargate. È invece possibile [modificare i contenitori per montarli all'avvio](#).

È possibile utilizzare Windows File Server FSx per distribuire carichi di lavoro Windows che richiedono l'accesso allo storage esterno condiviso, allo storage regionale ad alta disponibilità o allo storage ad alto throughput. Puoi montare uno o più FSx volumi del file system Windows File Server in un contenitore Amazon ECS eseguito su un'istanza Amazon ECS Windows. Puoi condividere FSx i volumi del file system Windows File Server tra più contenitori Amazon ECS all'interno di una singola attività Amazon ECS.

Per abilitare l'uso di FSx per Windows File Server con ECS, includi l'ID del file system FSx per Windows File Server e le informazioni correlate in una definizione di attività. Ciò viene riportato nel

frammento JSON della definizione di attività di esempio di seguito riportato. Per poter creare ed eseguire una definizione di attività, è necessario quanto riportato di seguito.

- Un' EC2 istanza ECS Windows aggiunta a un dominio valido. Può essere ospitato da un [AWS Directory Service for Microsoft Active Directory](#) Active Directory locale o da Active Directory ospitato autonomamente su Amazon. EC2
- Un parametro AWS Secrets Manager segreto o Systems Manager che contiene le credenziali utilizzate per aggiungere il dominio Active Directory e collegare il file system FSx per Windows File Server. I valori delle credenziali sono le credenziali di nome e password immesse durante la creazione di Active Directory.

Per un tutorial correlato, consulta [Scopri come configurare FSx i file system Windows File Server per Amazon ECS](#).

Considerazioni

Quando si utilizza FSx per volumi Windows File Server, si consideri quanto segue:

- FSx per Windows File Server con Amazon ECS supporta solo EC2 istanze Amazon Windows. EC2Le istanze Amazon Linux non sono supportate.
- FSx per Windows File Server con Amazon ECS non supporta AWS Fargate.
- FSx per Windows File Server con Amazon ECS con modalità awsvpc di rete richiede una versione 1.54.0 o successiva dell'agente contenitore.
- Il numero massimo di lettere di unità che possono essere utilizzate per un processo Amazon ECS è 23. A ogni attività con un volume FSx per Windows File Server viene assegnata una lettera di unità.
- Per impostazione predefinita, il tempo di pulizia delle risorse dell'attività è di tre ore dopo la fine dell'attività. Anche se non viene utilizzata da alcuna attività, una mappatura di file creata da un'attività persiste per 3 ore. Il tempo di pulizia di default può essere configurato utilizzando la variabile di ambiente ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION di Amazon ECS. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).
- Le attività in genere vengono eseguite solo nello stesso VPC del file system FSx per Windows File Server. Tuttavia, è possibile avere il supporto cross-VPC se esiste una connettività di rete consolidata tra il VPC del cluster Amazon ECS e il file system FSx per Windows File Server tramite peering VPC.
- È possibile controllare l'accesso a un file system FSx per Windows File Server a livello di rete configurando i gruppi di sicurezza VPC. Solo le attività ospitate su EC2 istanze unite al dominio

Active Directory con gruppi di sicurezza Active Directory correttamente configurati possono accedere alla condivisione di file FSx per Windows File Server. Se i gruppi di sicurezza non sono configurati correttamente, Amazon ECS non riesce ad avviare l'attività con il seguente messaggio di errore: `unable to mount file system fs-id.`»

- FSx per Windows File Server è integrato con AWS Identity and Access Management (IAM) per controllare le azioni che gli utenti e i gruppi IAM possono intraprendere su risorse specifiche FSx per Windows File Server. Con l'autorizzazione del client, i clienti possono definire ruoli IAM che consentono o negano l'accesso a file system specifici FSx per Windows File Server, richiedono facoltativamente l'accesso in sola lettura e facoltativamente consentono o impediscono l'accesso root al file system da parte del client. Per ulteriori informazioni, consulta la sezione [Sicurezza](#) nella Amazon FSx Windows User Guide.

Procedure consigliate FSx per l'utilizzo per Windows File Server con Amazon ECS

Prendi nota delle seguenti best practice consigliate quando utilizzi FSx Windows File Server con Amazon ECS.

Controlli di sicurezza e accesso FSx per Windows File Server

FSx per Windows File Server offre le seguenti funzionalità di controllo dell'accesso che è possibile utilizzare per garantire che i dati archiviati in un file system FSx per Windows File Server siano sicuri e accessibili solo dalle applicazioni che ne hanno bisogno.

Crittografia dei dati FSx per volumi Windows File Server

FSx per Windows File Server supporta due forme di crittografia per i file system. Sono la crittografia dei dati in transito e la crittografia a riposo. La crittografia dei dati in transito è supportata nelle condivisioni di file mappate su un'istanza di contenitore che supporta il protocollo SMB 3.0 o versioni successive. La crittografia dei dati inattivi viene abilitata automaticamente durante la creazione di un FSx file system Amazon. Amazon crittografa FSx automaticamente i dati in transito utilizzando la crittografia SMB quando accedi al tuo file system senza che tu debba modificare le tue applicazioni. Per ulteriori informazioni, consulta la sezione [Crittografia dei dati FSx in Amazon](#) nella Guida FSx per l'utente di Amazon per Windows File Server.

Usa Windows ACLs per il controllo degli accessi a livello di cartella

L' EC2 istanza Amazon Windows accede alle condivisioni di FSx file Amazon utilizzando le credenziali di Active Directory. Utilizza elenchi di controllo degli accessi standard di Windows (ACLs)

per un controllo granulare degli accessi a livello di file e cartella. È possibile creare più credenziali, ognuna per una cartella specifica all'interno della condivisione associata a un'attività specifica.

Nell'esempio seguente, l'attività ha accesso alla cartella App01 utilizzando una credenziale salvata in Secrets Manager. Il suo Amazon Resource Name (ARN) è. 1234

```
"rootDirectory": "\\path\\to\\my\\data\\App01",
"credentialsParameter": "arn-1234",
"domain": "corp.fullyqualified.com",
```

In un altro esempio, un'attività ha accesso alla cartella App02 utilizzando una credenziale salvata in Secrets Manager. Il suo ARN è 6789.

```
"rootDirectory": "\\path\\to\\my\\data\\App02",
"credentialsParameter": "arn-6789",
"domain": "corp.fullyqualified.com",
```

Specificare un file system FSx per Windows File Server in una definizione di attività Amazon ECS

Da utilizzare FSx per i volumi del file system Windows File Server per i contenitori, specificate le configurazioni del volume e del punto di montaggio nella definizione dell'attività. Il seguente frammento JSON della definizione di attività illustra la sintassi degli oggetti `volumes` e `mountPoints` per un container.

```
{
  "containerDefinitions": [
    {
      "entryPoint": [
        "powershell",
        "-Command"
      ],
      "portMappings": [],
      "command": ["New-Item -Path C:\\fsx-windows-dir\\index.html -ItemType file -Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>It Works!</h2> <p>You are using Amazon FSx for Windows File Server file system for persistent container storage.</p>' -Force"],
      "cpu": 512,
      "memory": 256,
      "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2019",
```

```

        "essential": false,
        "name": "container1",
        "mountPoints": [
            {
                "sourceVolume": "fsx-windows-dir",
                "containerPath": "C:\\fsx-windows-dir",
                "readOnly": false
            }
        ],
    },
    {
        "entryPoint": [
            "powershell",
            "-Command"
        ],
        "portMappings": [
            {
                "hostPort": 443,
                "protocol": "tcp",
                "containerPort": 80
            }
        ],
        "command": ["Remove-Item -Recurse C:\\inetpub\\wwwroot\\* -Force; Start-Sleep -Seconds 120; Move-Item -Path C:\\fsx-windows-dir\\index.html -Destination C:\\inetpub\\wwwroot\\index.html -Force; C:\\ServiceMonitor.exe w3svc"],
        "mountPoints": [
            {
                "sourceVolume": "fsx-windows-dir",
                "containerPath": "C:\\fsx-windows-dir",
                "readOnly": false
            }
        ],
        "cpu": 512,
        "memory": 256,
        "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2019",
        "essential": true,
        "name": "container2"
    }
],
"family": "fsx-windows",
"executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
"volumes": [
    {

```

```
    "name": "fsx-windows-dir",
    "fsxWindowsFileServerVolumeConfiguration": {
      "fileSystemId": "fs-0eeb5730b2EXAMPLE",
      "authorizationConfig": {
        "domain": "example.com",
        "credentialsParameter": "arn:arn-1234"
      },
      "rootDirectory": "share"
    }
  ]
}
```

FSxWindowsFileServerVolumeConfiguration

Tipo: oggetto

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizza FSx il file system [Windows File Server](#) per l'archiviazione delle attività.

fileSystemId

Tipo: stringa

Campo obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

rootDirectory

Tipo: stringa

Campo obbligatorio: sì

La directory all'interno del FSx file system Windows File Server da montare come directory principale all'interno dell'host.

authorizationConfig

credentialsParameter

Tipo: stringa

Campo obbligatorio: sì

Le opzioni delle credenziali di autorizzazione:

- L'Amazon Resource Name (ARN) di un segreto di [Secrets Manager](#).
- Amazon Resource Name (ARN) di un parametro di [Systems Manager](#).

domain

Tipo: stringa

Campo obbligatorio: sì

Un nome di dominio completo ospitato da una directory [AWS Directory Service for Microsoft Active Directory](#) (AWS Managed Microsoft AD) o da un Active Directory ospitato autonomamente EC2 .

Metodi per l'archiviazione delle FSx credenziali del volume Windows File Server

Esistono due metodi diversi per archiviare le credenziali da utilizzare con il parametro delle credenziali.

- AWS Secrets Manager segreto

Questa credenziale può essere creata nella AWS Secrets Manager console utilizzando la categoria segreta Altro tipo. Aggiungi una riga per ognuna key/value pair, username/admin e *password* password/.

- Parametro di Systems Manager

Queste credenziali possono essere create nella console dei parametri di Systems Manager immettendo del testo nel modulo illustrato nel frammento di codice di esempio seguente.

```
{
  "username": "admin",
  "password": "password"
}
```

credentialsParameter nel parametro FSxWindowsFileServerVolumeConfiguration della definizione di attività conterrà l'ARN del segreto o l'ARN del parametro di Systems Manager. Per ulteriori informazioni, consulta [Cos'è AWS Secrets Manager](#) nella Guida per l'utente di Secrets Manager e [Archivio parametri di Systems Manager](#) nella Guida per l'utente di Systems Manager.

Scopri come configurare FSx i file system Windows File Server per Amazon ECS

Scopri come avviare un'istanza Windows ottimizzata per Amazon ECS che ospita un file system FSx per Windows File Server e contenitori in grado di accedere al file system. A tale scopo, è innanzitutto necessario creare un Microsoft Active Directory AWS Directory Service AWS gestito. Quindi, crei un file system e un cluster FSx per Windows File Server File Server con un' EC2 istanza Amazon e una definizione di attività. Configurate la definizione delle attività per i contenitori in modo che utilizzino il file system FSx per Windows File Server. Infine, esegui il test del file system.

Occorrono dai 20 ai 45 minuti ogni volta che si avvia o si elimina il file system Active Directory o il file system FSx per Windows File Server. Considera almeno 90 minuti per completare il tutorial oppure completalo in più sessioni.

Prerequisiti per il tutorial

- Un utente amministratore. Consultare [Configurazione per l'uso di Amazon ECS](#).
- (Facoltativo) Una PEM key pair per la connessione all'istanza di EC2 Windows tramite accesso RDP. Per informazioni su come creare coppie di chiavi, consulta le coppie di [EC2 chiavi Amazon e EC2 le istanze Amazon](#) nella Amazon EC2 User Guide.
- Un VPC con almeno una sottorete pubblica e una sottorete privata e un gruppo di sicurezza. È possibile utilizzare il VPC di default. Non è necessario un gateway o un dispositivo NAT. AWS Directory Service non supporta Network Address Translation (NAT) con Active Directory. Affinché ciò funzioni, il file system Active Directory, FSx per Windows File Server, il cluster ECS e l'EC2 istanza devono trovarsi all'interno del VPC. Per ulteriori informazioni su VPCs Active Directory, consulta [Creazione di un VPC e Prerequisiti per la creazione di un Microsoft AWS AD gestito](#).
- Le autorizzazioni IAM `ecsInstanceRole` e `ecsTaskExecution Role` sono associate al tuo account. Questi ruoli collegati ai servizi consentono ai servizi di effettuare chiamate API e accedere a container, segreti, directory e server di file per tuo conto.

Fase 1: Creazione dei ruoli di accesso IAM

Crea un cluster con la AWS Management Console.

1. Verifica se ne possiedi uno `ecsInstanceRole` e scopri come puoi crearne uno se non ne hai uno. [Ruolo IAM delle istanze di container Amazon ECS](#)
2. Consigliamo di personalizzare le policy dei ruoli per ottenere autorizzazioni minime in un ambiente di produzione effettivo. Per completare questo tutorial, verifica che la seguente politica AWS gestita sia allegata al tuo `ecsInstanceRole`. Collega la policy, se non è già collegata.

- EC2ContainerServiceforEC2Ruolo di Amazon
- Amazon SSMManged InstanceCore
- Amazon SSMDirectory ServiceAccess

Per allegare politiche AWS gestite.

- Aprire la [console IAM](#).
 - Nel pannello di navigazione, selezionare Ruoli.
 - Scegli un ruolo gestito da AWS .
 - Scegli Autorizzazioni, Collega policy.
 - Per limitare le policy disponibili da collegare, utilizza Filtro.
 - Seleziona la policy appropriata, quindi scegli Collega policy.
- Verifica se hai un ecsTaskExecution ruolo e scopri come puoi crearne uno se non ne hai uno.
[Ruolo IAM di esecuzione di attività Amazon ECS](#)

Consigliamo di personalizzare le policy dei ruoli per ottenere autorizzazioni minime in un ambiente di produzione effettivo. Per completare questo tutorial, verifica che le seguenti politiche AWS gestite siano allegate al tuo ecsTaskExecution ruolo. Collega le policy, se non sono già collegate. Utilizza la procedura indicata nella sezione precedente per allegare le politiche AWS gestite.

- SecretsManagerReadWrite
- Amazon FSx ReadOnlyAccess
- Amazon SSMRead OnlyAccess
- Amazon ECSTask ExecutionRolePolicy

Fase 2: Creazione di Windows Active Directory (AD)

- Segui i passaggi descritti in [Creazione di AWS Managed Microsoft AD](#) nella AWS Directory Service Administration Guide. Utilizza il VPC designato per questo tutorial. Nella fase 3 della creazione di AWS Managed Microsoft AD, salva il nome utente e la password di amministratore da utilizzare nel passaggio successivo. Inoltre, prendi nota del nome DNS di directory completo per i passaggi futuri. È possibile completare il passaggio seguente durante la creazione di Active Directory.

2. Crea un segreto di AWS Secrets Manager da utilizzare nei passaggi seguenti. Per ulteriori informazioni, consulta Guida [introduttiva a Secrets Manager](#) nella Guida per l'utente di AWS Secrets Manager.
 - a. Apri la [console Secrets Manager](#).
 - b. Fai clic su Archivia un nuovo segreto.
 - c. Seleziona Altro tipo di segreti.
 - d. Per Chiave/valore segreto, nella prima riga, crea una chiave **username** con valore **admin**. Fai clic su + Aggiungi riga.
 - e. Nella nuova riga, crea una chiave **password**. Per ulteriori informazioni, digitate la password inserita nel passaggio 3 di Create Your AWS Managed AD Directory.
 - f. Fai clic sul pulsante Successivo.
 - g. Specifica un nome e una descrizione per il segreto. Fai clic su Next (Successivo).
 - h. Fai clic su Next (Successivo). Fai clic su Archivia.
 - i. Dall'elenco nella pagina Segreti, seleziona il segreto appena creato.
 - j. Salva l'ARN del nuovo segreto per utilizzarlo nelle fasi seguenti.
 - k. Durante la creazione di Active Directory, è possibile procedere alla fase successiva.

Fase 3: Verifica e aggiornamento del gruppo di sicurezza

In questa fase è possibile verificare e aggiornare le regole per il gruppo di sicurezza in uso. Pertanto è possibile utilizzare il gruppo di sicurezza di default creato per il VPC.

Verifica e aggiorna il gruppo di sicurezza.

È necessario creare o modificare il gruppo di sicurezza per inviare dati da e verso le porte, descritti in [Amazon VPC Security Groups](#) nella Guida FSx per l'utente di Windows File Server. A tale scopo, è possibile creare la regola in ingresso del gruppo di sicurezza visualizzata nella prima riga della tabella di regole in entrata riportata di seguito. Questa regola consente il traffico in ingresso dalle interfacce di rete (e le relative istanze associate) assegnate al gruppo di sicurezza. Tutte le risorse cloud create si trovano all'interno dello stesso VPC e collegate allo stesso gruppo di sicurezza. Pertanto, questa regola consente l'invio di traffico da e verso il file system FSx per Windows File Server, Active Directory e l'istanza ECS, se necessario. Le altre regole in entrata consentono al traffico di servire il sito Web e l'accesso RDP per la connessione all'istanza ECS.

Nella tabella seguente vengono illustrate le regole in entrata del gruppo di sicurezza necessarie per questo tutorial.

Tipo	Protocollo	Intervallo porte	Origine
Tutto il traffico	Tutti	Tutti	<i>sg-securi tygroup</i>
HTTPS	TCP	443	0.0.0.0/0
RDP	TCP	3389	indirizzo IP del laptop

Nella tabella seguente vengono illustrate le regole in uscita del gruppo di sicurezza necessarie per questo tutorial.

Tipo	Protocollo	Intervallo porte	Destinazione
Tutto il traffico	Tutti	Tutti	0.0.0.0/0

1. Apri la [EC2 console](#) e seleziona Security Groups dal menu a sinistra.
2. Dall'elenco dei gruppi di sicurezza ora visualizzati, seleziona la casella di controllo a sinistra del gruppo di sicurezza utilizzato per questo tutorial.

Vengono visualizzati i dettagli del gruppo di sicurezza.

3. Modifica le regole in ingresso e in uscita selezionando le schede Inbound rules (Regole in entrata) o Outbound rules (Regole in uscita) e scegliendo l'opzione Edit inbound rules (Modifica regole in entrata) o Edit outbound rules (Modifica regole in uscita). Modifica le regole in base a quelle visualizzate nelle tabelle precedenti. Dopo aver creato l' EC2 istanza più avanti in questo tutorial, modifica l'origine RDP della regola in entrata con l'indirizzo IP pubblico dell' EC2 istanza, come descritto in [Connect to your Windows using RDP](#) dalla Amazon EC2 User Guide.

Passaggio 4: crea un file system FSx per Windows File Server

Dopo aver verificato e aggiornato il gruppo di sicurezza e aver creato Active Directory e aver raggiunto lo stato attivo, crea il file system FSx per Windows File Server nello stesso VPC di Active Directory. Utilizza i passaggi seguenti per creare un file system FSx per Windows File Server per le tue attività Windows.

Creare il tuo primo file system.

1. Apri la [FSx console Amazon](#).
2. Nel pannello di controllo, scegli Crea file system per avviare la procedura guidata di creazione del file system.
3. Nella pagina Seleziona il tipo di file system, scegli FSx Windows File Server, quindi scegli Avanti. Viene visualizzata la pagina Crea file system.
4. Nella sezione Dettagli file system, specifica un nome per il file system. La denominazione dei file system ne semplifica la ricerca e la gestione. Puoi utilizzare fino a 256 caratteri Unicode. I caratteri consentiti sono lettere, numeri, spazi e caratteri speciali segno più (+), segno meno (-), segno uguale (=), punto (.), trattino basso (_), due punti (:) e barra (/).
5. Per Tipo di implementazione, scegli Single-AZ per implementare un file system distribuito in una zona di disponibilità singola. Single-AZ 2 è l'ultima generazione di file system a zona di disponibilità singola e supporta l'archiviazione SSD e HDD.
6. Per Tipo di archiviazione, scegli HDD.
7. In Capacità di archiviazione, specifica la capacità di archiviazione minima.
8. Mantieni Capacità di velocità effettiva sul valore di default.
9. Nella sezione Rete e sicurezza, scegli lo stesso Amazon VPC che hai scelto per la tua AWS Directory Service directory.
10. Per Gruppi di sicurezza VPC, scegli il gruppo di sicurezza verificato in Fase 3: Verifica e aggiornamento del gruppo di sicurezza.
11. Per Autenticazione Windows, scegli AWS Managed Microsoft Active Directory, quindi seleziona la tua directory AWS Directory Service dall'elenco.
12. Per Crittografia, mantieni l'impostazione di default Chiave di crittografia su aws/fsx (default).
13. Mantieni le impostazioni di default per Preferenze di manutenzione.
14. Fai clic sul pulsante Successivo.

15. Rivedi la configurazione del file system riportata nella pagina Crea file system. Come riferimento, prendi nota delle impostazioni del file system che è possibile modificare dopo la sua creazione. Scegliere Create file system (Crea file system).
16. Prendi nota dell'ID file system. Sarà utile in una fase successiva.

Puoi passare ai passaggi successivi per creare un cluster e un' EC2istanza durante la creazione del file system FSx per Windows File Server.

Fase 5: Creazione di un cluster Amazon ECS

Creazione di un cluster tramite la console Amazon ECS

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Configurazione del cluster, per Nome cluster, inserisci windows-fsx-cluster.
6. Espandi l'infrastruttura, cancella AWS Fargate (senza server) e seleziona le EC2 istanze Amazon.
 - Per creare un gruppo Auto Scaling, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona Create new group (Crea nuovo gruppo) e quindi fornisci i seguenti dettagli sul gruppo:
 - Per Sistema operativo/architettura, scegli Windows Server 2019 Core.
 - Ad EC2 esempio, scegli t2.medium o t2.micro.
7. Scegli Create (Crea).

Fase 6: creare un'istanza Amazon ottimizzata per Amazon EC2 ECS

Creazione di un'istanza di container Windows di Amazon ECS

Creazione di un'istanza Amazon ECS

1. Usa il comando `aws ssm get-parameters` per recuperare il nome dell'AMI per la regione che ospita il VPC. Per ulteriori informazioni, consulta [Recupero dei metadati dell'AMI ottimizzata per Amazon ECS](#).

2. Usa la EC2 console Amazon per avviare l'istanza.
 - a. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
 - b. Seleziona la Regione da utilizzare nella barra di navigazione.
 - c. Dalla EC2 dashboard, scegli Launch instance.
 - d. Per Name (Nome), inserisci un nome univoco.
 - e. Per Immagini di applicazioni e sistema operativo (Amazon Machine Image), nel campo cerca, inserisci il nome dell'AMI recuperata.
 - f. In Tipo di istanza, scegli t2.medium o t2.micro.
 - g. Per Key pair (login) (Coppia di chiavi [accesso]), scegli una coppia di chiavi. Se non specifichi una coppia di chiavi,
 - h. In Impostazioni di rete, per VPC e Sottorete, scegli il VPC e una sottorete pubblica.
 - i. In Network settings (Impostazioni di rete), per Security group (Gruppo di sicurezza), scegli un gruppo di sicurezza esistente o creane uno nuovo. Assicurati che il gruppo di sicurezza scelto disponga delle regole in entrata e in uscita definite in [Prerequisiti per il tutorial](#)
 - j. In Network settings (Impostazioni di rete), per Auto-assign Public IP (Assegna automaticamente un IP pubblico), seleziona Enable (Abilita).
 - k. Espandi Dettagli avanzati e in Directory aggiunta dominio, seleziona l'ID dell'Active Directory creata. Il dominio di questa opzione si aggiunge al tuo AD all'avvio dell' EC2 istanza.
 - l. In Dettagli avanzati, per il profilo dell'istanza IAM, scegli ecsInstanceRole.
 - m. Configura la tua istanza di container Amazon ECS con i seguenti dati utente. In Dettagli avanzati, incolla lo script seguente nel campo Dati utente, sostituendolo *cluster_name* con il nome del cluster.

```
<powershell>  
Initialize-ECSAgent -Cluster windows-fsx-cluster -EnableTaskIAMRole  
</powershell>
```

- n. Quando sei pronto, seleziona il campo di conferma e scegli Launch Instances (Avvia istanze).
 - o. Una pagina di conferma indicherà che l'istanza si sta avviando. Scegliere View Instances (Visualizza istanze) per chiudere la pagina di conferma e tornare alla console.
3. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
4. Nel riquadro di navigazione, scegli Cluster, quindi scegli. windows-fsx-cluster

5. Scegli la scheda Infrastruttura e verifica che l'istanza sia stata registrata nel `windows-fsx-clustercluster`.

Fase 7: Registrazione di una definizione di attività di Windows

Prima di eseguire i container Windows nel cluster Amazon ECS, devi registrare una definizione di attività. Il seguente esempio di definizione di attività mostra una pagina Web semplice. L'attività avvia due contenitori che hanno accesso al FSx file system. Il primo container scrive un file HTML nel file system. Il secondo container scarica il file HTML dal file system e serve la pagina Web.

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).
4. Nella casella dell'editor JSON, sostituisci i valori per il tuo ruolo di esecuzione delle attività e i dettagli sul tuo FSx file system, quindi scegli Salva.

```
{
  "containerDefinitions": [
    {
      "entryPoint": [
        "powershell",
        "-Command"
      ],
      "portMappings": [],
      "command": ["New-Item -Path C:\\fsx-windows-dir\\index.html -ItemType
file -Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body
{margin-top: 40px; background-color: #333;} </style> </head><body> <div
style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>It
Works!</h2> <p>You are using Amazon FSx for Windows File Server file system for
persistent container storage.</p>' -Force"],
      "cpu": 512,
      "memory": 256,
      "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
      "essential": false,
      "name": "container1",
      "mountPoints": [
        {
          "sourceVolume": "fsx-windows-dir",
```

```

        "containerPath": "C:\\fsx-windows-dir",
        "readOnly": false
    }
],
},
{
    "entryPoint": [
        "powershell",
        "-Command"
    ],
    "portMappings": [
        {
            "hostPort": 443,
            "protocol": "tcp",
            "containerPort": 80
        }
    ],
    "command": ["Remove-Item -Recurse C:\\inetpub\\wwwroot\\* -Force;
Start-Sleep -Seconds 120; Move-Item -Path C:\\fsx-windows-dir\\index.html -
Destination C:\\inetpub\\wwwroot\\index.html -Force; C:\\ServiceMonitor.exe
w3svc"],
    "mountPoints": [
        {
            "sourceVolume": "fsx-windows-dir",
            "containerPath": "C:\\fsx-windows-dir",
            "readOnly": false
        }
    ],
    "cpu": 512,
    "memory": 256,
    "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
    "essential": true,
    "name": "container2"
}
],
"family": "fsx-windows",
"executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
"volumes": [
    {
        "name": "fsx-windows-dir",
        "fsxWindowsFileServerVolumeConfiguration": {
            "filesystemId": "fs-0eeb5730b2EXAMPLE",
            "authorizationConfig": {

```

```
        "domain": "example.com",
        "credentialsParameter": "arn:arn-1234"
    },
    "rootDirectory": "share"
}
]
```

Fase 8: Esecuzione di un processo e visualizzazione dei risultati

Prima di eseguire l'operazione, verificate che lo stato del file system FSx per Windows File Server sia Disponibile. Una volta disponibile, è possibile eseguire un processo utilizzando la definizione di attività creata. Il processo inizia creando container che mescolano un file HTML utilizzando il file system. Dopo il mescolamento, un server Web serve la pagina HTML semplice.

Note

Potresti non essere in grado di connetterti al sito Web da una VPN.

Esegui un'attività e visualizza i risultati tramite la console Amazon ECS.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel riquadro di navigazione, scegli Cluster, quindi scegli windows-fsx-cluster
3. Seleziona la scheda Attività, quindi Esegui nuova attività.
4. Per Tipo di avvio, scegli EC2.
5. In Configurazione dell'implementazione, per Definizione dell'attività, scegli fsx-windows, quindi scegli Crea.
6. Quando lo stato dell'attività è IN ESECUZIONE, scegli l'ID attività.
7. In Container, quando lo stato di container1 è ARRESTATO, seleziona container2 per visualizzarne i dettagli.
8. In Dettagli del container per container2, seleziona Associazioni di rete e fai clic sull'indirizzo IP esterno associato al container. Si aprirà il browser e sarà visualizzato il seguente messaggio.

```
Amazon ECS Sample App
It Works!
```


You are using Amazon FSx for Windows File Server file system for persistent container storage.

 Note

Possono essere necessari alcuni minuti per visualizzare il messaggio. Se dopo qualche minuto non viene visualizzato il messaggio, assicurati che non stai utilizzando una VPN e che il gruppo di sicurezza per l'istanza di container consenta il traffico di rete in entrata sulla porta 443.

Fase 9: Pulizia

 Note

Sono necessari dai 20 ai 45 minuti FSx per eliminare il file system Windows File Server o l'AD. È necessario attendere il completamento delle operazioni di eliminazione del file system FSx for Windows File Server prima di iniziare le operazioni di eliminazione di AD.

Elimina FSx per il file system Windows File Server.

1. Apri la [FSx console Amazon](#)
2. Scegli il pulsante di opzione a sinistra del file system FSx per Windows File Server che hai appena creato.
3. Scegli Azioni.
4. Seleziona Elimina file system.

Elimina l'AD.

1. Apri la [AWS Directory Service console](#).
2. Scegli il pulsante di opzione a sinistra dell'AD appena creato.
3. Scegli Azioni.
4. Seleziona Elimina directory.

Elimina il cluster.

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nel riquadro di navigazione, scegli Cluster, quindi scegli fsx-windows-cluster
3. Scegli Delete cluster (Elimina cluster).
4. Inserisci la frase, quindi scegli Elimina.

Termina l'istanza. EC2

1. Apri la [EC2 console Amazon](#).
2. Nel menu a sinistra, seleziona Istanze.
3. Seleziona la casella a sinistra dell' EC2 istanza che hai creato.
4. Fai clic su Stato istanza, Termina istanza.

Elimina il segreto.

1. Apri la [console Secrets Manager](#).
2. Seleziona il segreto creato per questa procedura.
3. Fai clic su Operazioni.
4. Seleziona Elimina segreto.

Usa i volumi Docker con Amazon ECS

Quando utilizzi i volumi Docker, puoi usare il driver `local` integrato o un driver di volumi di terze parti. I volumi Docker sono gestiti da Docker e viene creata una directory in `/var/lib/docker/volumes` sull'istanza di container che contiene i dati del volume.

Per usare i volumi Docker, specifica `dockerVolumeConfiguration` nella definizione di attività. Per ulteriori informazioni, consulta [Volumes in Docker documentazione](#).

Alcuni casi di utilizzo comune per i volumi Docker sono i seguenti:

- Offrire volumi di dati persistenti per l'utilizzo con i container
- Condividere un volume di dati definito in diverse posizioni su differenti container nella stessa istanza di container

- Definire un volume di dati vuoto, non persistente e montarlo su più container all'interno della stessa attività
- Per fornire un volume di dati per l'attività gestita da un driver di terze parti

Considerazioni sull'utilizzo dei volumi Docker

Quando usi volumi Docker, tieni presenti le considerazioni seguenti:

- I volumi Docker sono supportati solo quando si utilizza il tipo di EC2 avvio o istanze esterne.
- I container Windows supportano solo l'uso del driver `local`.
- Se viene utilizzato un driver di terze parti, assicurati che sia installato e attivo sull'istanza di container prima dell'avvio dell'agente del container. Se il driver di terze parti non è attivo prima dell'avvio dell'agente, puoi riavviare l'agente del container utilizzando uno dei seguenti comandi:
 - Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS:

```
sudo stop ecs && sudo start ecs
```

Per informazioni su come specificare un volume Docker in una definizione di attività, consulta.

[Specificare un volume Docker in una definizione di attività Amazon ECS](#)

Specificare un volume Docker in una definizione di attività Amazon ECS

Prima che i container possano utilizzare i volumi di dati, è necessario specificare le configurazioni del punto di montaggio e del volume nella definizione di attività. Questa sezione descrive la configurazione del volume per un container. Per le attività che usano un volume Docker, specifica `dockerVolumeConfiguration`. Per le attività che usano un volume host di montaggio vincolato, specifica `host` e facoltativamente `sourcePath`.

Il seguente JSON della definizione di attività illustra la sintassi degli oggetti `volumes` e `mountPoints` per un container.

```
{  
  "containerDefinitions": [  

```

```
{
  "mountPoints": [
    {
      "sourceVolume": "string",
      "containerPath": "/path/to/mount_volume",
      "readOnly": boolean
    }
  ]
},
"volumes": [
  {
    "name": "string",
    "dockerVolumeConfiguration": {
      "scope": "string",
      "autoprovision": boolean,
      "driver": "string",
      "driverOpts": {
        "key": "value"
      },
      "labels": {
        "key": "value"
      }
    }
  }
]
}
```

name

Tipo: string

Campo obbligatorio: no

Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e caratteri di sottolineatura (-). _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. sourceVolume mountPoints

dockerVolumeConfiguration

Tipo: oggetto [DockerVolumeConfiguration](#)

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker. I volumi Docker sono supportati solo quando si eseguono attività su EC2 istanze. I contenitori Windows supportano solo l'uso del `local` driver. Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

Tipo: stringa

Valori validi: `task` | `shared`

Campo obbligatorio: no

L'ambito del volume Docker che determina il suo ciclo di vita. I volumi Docker che rientrano nell'ambito `task` vengono automaticamente assegnati all'avvio del processo e distrutti quando il processo viene arrestato. I volumi Docker che vengono definiti come `shared` vengono mantenuti dopo l'arresto del processo.

`autoprovision`

Tipo: Booleano

Valore predefinito: `false`

Campo obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo viene utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omissso.

`driver`

Tipo: string

Campo obbligatorio: no

Il driver del volume Docker da utilizzare. Il valore del driver deve corrispondere al nome del driver fornito da Docker perché questo nome viene utilizzato per il posizionamento delle attività. Se il driver è stato installato utilizzando la CLI del plug-in Docker, `docker plugin ls` utilizzalo per recuperare il nome del driver dall'istanza del contenitore. Se il driver è stato installato utilizzando un altro metodo, utilizza Docker Plugin Discovery per recuperare il nome del driver.

`driverOpts`

Tipo: string

Campo obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da esaminare. Questo parametro viene mappato `DriverOpts` alla sezione Crea un volume di Docker.

`labels`

Tipo: string

Campo obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato `Volumes` all'API Docker `create-container` e `--volume` all'opzione `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Per le attività eseguite su EC2 istanze che eseguono il sistema operativo Windows, lasciate il valore predefinito di `false`.

Esempi di volumi Docker per Amazon ECS

Gli esempi seguenti mostrano come fornire storage temporaneo per un contenitore e come fornire un volume condiviso per più container e come fornire storage persistente NFS per un contenitore.

Fornire uno storage temporaneo per un contenitore utilizzando un volume Docker

In questo esempio, un container utilizza un volume di dati vuoto che viene smaltito al termine dell'attività. Ad esempio, potresti avere un container che deve accedere alla posizione di storage di alcuni file temporanei durante un'attività. Questa attività può essere eseguita utilizzando un volume Docker.

1. Nella sezione `volumes` della definizione di attività, definisci un volume di dati con i valori `name` e `DockerVolumeConfiguration`. In questo esempio, specifichiamo l'ambito come `task` in modo che il volume venga eliminato dopo l'arresto dell'attività e venga utilizzato il driver `local` incorporato.

```
"volumes": [  
  {  
    "name": "scratch",  
    "dockerVolumeConfiguration" : {  
      "scope": "task",  
      "driver": "local",  
      "labels": {  
        "scratch": "space"  
      }  
    }  
  }  
]
```

2. Nella sezione `containerDefinitions`, definisci un container con valori `mountPoints` che faccia riferimento al nome del volume definito e al valore `containerPath` per montare il volume sul container.

```
"containerDefinitions": [  

```

```
{
  "name": "container-1",
  "mountPoints": [
    {
      "sourceVolume": "scratch",
      "containerPath": "/var/scratch"
    }
  ]
}
```

Fornire spazio di archiviazione persistente per più contenitori utilizzando un volume Docker

In questo esempio, desideri un volume condiviso per più container da utilizzare e che persista dopo l'interruzione di qualsiasi singola attività che lo utilizza. Il driver `local` integrato è in uso. Per questo motivo, il volume è ancora legato al ciclo di vita dell'istanza di container.

1. Nella sezione `volumes` della definizione di attività, definisci un volume di dati con i valori `name` e `DockerVolumeConfiguration`. In questo esempio, specificare un ambito `shared` in modo che il volume persista, imposta il provisioning automatico su `true`. In questo modo il volume viene creato per l'uso. Quindi, utilizza anche il driver `local` integrato.

```
"volumes": [
  {
    "name": "database",
    "dockerVolumeConfiguration" : {
      "scope": "shared",
      "autoprovision": true,
      "driver": "local",
      "labels": {
        "database": "database_name"
      }
    }
  ]
}
```

2. Nella sezione `containerDefinitions`, definisci un container con valori `mountPoints` che faccia riferimento al nome del volume definito e al valore `containerPath` per montare il volume sul container.

```
"containerDefinitions": [
```



```

{
  "name": "container-1",
  "mountPoints": [
    {
      "sourceVolume": "database",
      "containerPath": "/var/database"
    }
  ]
},
{
  "name": "container-2",
  "mountPoints": [
    {
      "sourceVolume": "database",
      "containerPath": "/var/database"
    }
  ]
}
]

```

Fornire lo spazio di archiviazione persistente per un container utilizzando un volume Docker

In questo esempio, un container utilizza un volume di dati NFS che viene montato automaticamente all'avvio dell'attività e viene smontato al termine. Questo utilizza il driver `local` integrato in Docker. Un esempio di caso d'uso potrebbe essere quello in cui hai un'archiviazione NFS locale e hai la necessità di accedervi attraverso un'attività ECS Anywhere. Ciò può essere ottenuto utilizzando un volume Docker con opzione driver NFS.

1. Nella sezione `volumes` della definizione di attività, definisci un volume di dati con i valori `name` e `DockerVolumeConfiguration`. In questo esempio, specifica un ambito `task` in modo che il volume venga smontato al termine dell'attività. Usa il driver `local` e configura le `driverOpts` con le opzioni `type`, `device` e `o` di conseguenza. Sostituisci `NFS_SERVER` con l'endpoint del server NFS.

```

"volumes": [
  {
    "name": "NFS",
    "dockerVolumeConfiguration" : {
      "scope": "task",
      "driver": "local",
      "driverOpts": {

```

```

        "type": "nfs",
        "device": "$NFS_SERVER:/mnt/nfs",
        "o": "addr=$NFS_SERVER"
      }
    }
  ]

```

2. Nella sezione `containerDefinitions`, definisci un container con valori di `mountPoints` che facciano riferimento al nome del volume definito e al valore di `containerPath` per montare il volume sul container.

```

"containerDefinitions": [
  {
    "name": "container-1",
    "mountPoints": [
      {
        "sourceVolume": "NFS",
        "containerPath": "/var/nfsmount"
      }
    ]
  }
]

```

Usa bind mount con Amazon ECS

Con bind mount, un file o una directory su un host, ad esempio un' EC2 istanza Amazon, viene montato in un contenitore. I bind mount sono supportati per le attività ospitate su istanze Fargate e Amazon. EC2 I bind mount sono legati al ciclo di vita del contenitore che li utilizza. Una volta arrestati tutti i container che utilizzano un montaggio vincolato, ad esempio quando viene arrestata un'attività, i dati vengono rimossi. Per le attività ospitate su EC2 istanze Amazon, i dati possono essere collegati al ciclo di vita dell' EC2 istanza Amazon host specificando un `host sourcePath` valore opzionale nella definizione dell'attività. Per ulteriori informazioni, consulta [Bind mounts](#) nella documentazione Docker.

I seguenti sono casi d'uso comune dei montaggi vincolati.

- Per fornire un volume di dati vuoto da montare in uno o più container.
- Per fornire un volume di dati host in uno o più container.

- Per condividere un volume di dati da un container di origine con altri container nello stesso processo.
- Per esporre un percorso e il relativo contenuto da un Dockerfile a uno o più container.

Considerazioni su quando utilizzare i montaggi vincolati

Quando usi i montaggi vincolati, tieni presente le seguenti considerazioni.

- Per impostazione predefinita, le attività ospitate AWS Fargate utilizzando una versione della piattaforma 1.4.0 o successiva (Linux) 1.0.0 o successiva (Windows) ricevono un minimo di 20 GiB di storage temporaneo per i bind mount. È possibile aumentare la quantità totale di storage temporaneo fino a un massimo di 200 GiB specificando il parametro nella definizione dell'ephemeralStorageattività.
- Per esporre i file da un Dockerfile a un volume di dati quando viene eseguito un processo, il piano dati di Amazon ECS cerca una direttiva VOLUME. Se il percorso assoluto specificato nella direttiva VOLUME è lo stesso presente nel containerPath specificato nella definizione di attività, i dati nel percorso della direttiva VOLUME vengono copiati sul volume di dati. Nell'esempio Dockerfile seguente, un file denominato `examplefile` nella directory `/var/log/exported` viene scritto sull'host e quindi montato all'interno del container.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest
RUN mkdir -p /var/log/exported
RUN touch /var/log/exported/examplefile
VOLUME ["/var/log/exported"]
```

Di default, le autorizzazioni dei volumi sono impostate su 0755 e il proprietario è root. Queste autorizzazioni possono essere personalizzate nel Dockerfile. L'esempio seguente definisce il proprietario della directory come `node`.

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest
RUN yum install -y shadow-utils && yum clean all
RUN useradd node
RUN mkdir -p /var/log/exported && chown node:node /var/log/exported
RUN touch /var/log/exported/examplefile
USER node
VOLUME ["/var/log/exported"]
```

- Per le attività ospitate su EC2 istanze Amazon, quando un `sourcePath` valore `host` and non è specificato, il demone Docker gestisce il `bind mount` per te. Quando nessun container fa riferimento a questo montaggio vincolato, viene alla fine eliminato dal servizio di pulizia dell'attività dell'agente del container Amazon ECS. Per impostazione predefinita, ciò avviene tre ore dopo la chiusura del container. Tuttavia, puoi configurare questa durata con la variabile dell'agente `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION`. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#). Se è necessario che questi dati vengano conservati oltre il ciclo di vita del container, specifica un valore `sourcePath` per il montaggio vincolato.

Specificare un bind mount in una definizione di attività Amazon ECS

Per le attività di Amazon ECS ospitate su istanze Fargate o EC2 Amazon, il seguente frammento JSON di definizione di attività mostra la sintassi `mountPoints`, e gli oggetti per `volumes` una definizione di attività. `ephemeralStorage`

```
{
  "family": "",
  ...
  "containerDefinitions" : [
    {
      "mountPoints" : [
        {
          "containerPath" : "/path/to/mount_volume",
          "sourceVolume" : "string"
        }
      ],
      "name" : "string"
    }
  ],
  ...
  "volumes" : [
    {
      "name" : "string"
    }
  ],
  "ephemeralStorage": {
    "sizeInGiB": integer
  }
}
```

Per le attività di Amazon ECS ospitate su EC2 istanze Amazon, puoi utilizzare il `host` parametro opzionale e `sourcePath` quando specifichi i dettagli del volume delle attività. Quando viene specificato, lega il montaggio vincolato al ciclo di vita dell'attività anziché al container.

```
"volumes" : [  
  {  
    "host" : {  
      "sourcePath" : "string"  
    },  
    "name" : "string"  
  }  
]
```

Di seguito sono riportate descrizioni più dettagliate per ogni parametro di definizione di attività.

name

Tipo: string

Campo obbligatorio: no

Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e caratteri di sottolineatura (-). _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume mountPoints`

host

Campo obbligatorio: no

Il `host` parametro viene utilizzato per collegare il ciclo di vita del `bind mount` all' EC2 istanza Amazon `host`, anziché all'attività e alla posizione in cui è archiviata. Se il parametro `host` è vuoto, il daemon Docker assegna un percorso `host` per il tuo volume di dati, ma non è garantito che i dati vengano mantenuti dopo che viene interrotta l'esecuzione del container a essi associato.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`.

Note

Il `sourcePath` parametro è supportato solo quando si utilizzano attività ospitate su EC2 istanze Amazon.

sourcePath

Tipo: string

Campo obbligatorio: no

Quando viene utilizzato il `host` parametro, specifica `sourcePath` a per dichiarare il percorso sull' EC2 istanza Amazon host che viene presentata al contenitore. Se questo parametro è vuoto, il daemon Docker assegna automaticamente un percorso host. Se il `host` parametro contiene una posizione di `sourcePath` file, il volume di dati persiste nella posizione specificata sull' EC2 istanza Amazon host finché non lo elimini manualmente. Se il `sourcePath` valore non esiste sull' EC2 istanza Amazon host, il demone Docker lo crea. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

mountPoints

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato `Volumes` all'API Docker `create-container` e `--volume` all'opzione `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

sourceVolume

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

containerPath

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Per le attività eseguite su EC2 istanze che eseguono il sistema operativo Windows, lasciate il valore predefinito di `false`.

`ephemeralStorage`

Tipo: oggetto

Campo obbligatorio: no

La quantità di archiviazione temporanea da allocare per il processo. Questo parametro viene utilizzato per espandere la quantità totale di spazio di archiviazione temporanea disponibile, oltre la quantità predefinita, per le attività ospitate AWS Fargate utilizzando la versione della piattaforma 1.4.0 o successiva (Linux) 1.0.0 o successiva (Windows).

È possibile utilizzare l'interfaccia della riga di comando di Copilot CloudFormation, l' AWS SDK o la CLI per specificare lo storage temporaneo per un bind mount.

Esempi di montaggio Bind per Amazon ECS

I seguenti esempi coprono i casi d'uso comuni dell'utilizzo di un bind mount per i contenitori.

Come allocare una maggiore quantità di spazio di archiviazione temporanea per un processo Fargate

Per le attività Amazon ECS ospitate su Fargate che utilizzano la versione della piattaforma 1.4.0 o successiva (Linux) o 1.0.0 o successiva (Windows), è possibile allocare più della quantità predefinita di storage temporaneo per i container nell'attività da utilizzare. Questo esempio può essere incorporato negli altri esempi per allocare più spazio di archiviazione temporanea per i processi Fargate.

- Nella definizione di attività, definisci un oggetto `ephemeralStorage`. La `sizeInGiB` deve essere un numero intero compreso tra i valori di 21 e 200 ed è espresso in GiB.

```
"ephemeralStorage": {  
  "sizeInGiB": integer  
}
```

Come fornire un volume di dati vuoto per uno o più container

In alcuni casi, si può fornire ai container in un processo un po' di spazio scratch. Ad esempio, potresti avere due container di database che devono accedere alla stessa posizione di storage dei file temporanei durante un'attività. Questo può essere ottenuto utilizzando un montaggio vincolato.

1. Nella sezione `volumes` della definizione di attività, definisci un montaggio vincolato con il nome `database_scratch`.

```
"volumes": [  
  {  
    "name": "database_scratch"  
  }  
]
```

2. Nella sezione `containerDefinitions`, crea le definizioni di container del database. in modo che montino il volume.

```
"containerDefinitions": [  
  {  
    "name": "database1",  
    "image": "my-repo/database",  
    "cpu": 100,  
    "memory": 100,  
    "essential": true,  
    "mountPoints": [  
      {  
        "sourceVolume": "database_scratch",  
        "containerPath": "/var/scratch"  
      }  
    ]  
  },  
  {  
    "name": "database2",  
    "image": "my-repo/database",  
    "cpu": 100,
```



```

    "memory": 100,
    "essential": true,
    "mountPoints": [
      {
        "sourceVolume": "database_scratch",
        "containerPath": "/var/scratch"
      }
    ]
  }
]

```

Come utilizzare un percorso e il relativo contenuto in un Dockerfile in un container

In questo esempio, hai un Dockerfile che scrive i dati che si desidera montare all'interno di un container. Questo esempio funziona per le attività ospitate su istanze Fargate o Amazon EC2 .

1. Crea un Dockerfile. L'esempio seguente utilizza l'immagine del container Amazon Linux 2 e crea un file denominato `examplefile` nella directory `/var/log/exported` che vogliamo montare all'interno del container. La direttiva `VOLUME` dovrebbe specificare un percorso assoluto.

```

FROM public.ecr.aws/amazonlinux/amazonlinux:latest
RUN mkdir -p /var/log/exported
RUN touch /var/log/exported/examplefile
VOLUME ["/var/log/exported"]

```

Di default, le autorizzazioni dei volumi sono impostate su `0755` e il proprietario è `root`. Queste autorizzazioni possono essere modificate nel Dockerfile. Nell'esempio seguente il proprietario della directory `/var/log/exported` è impostato su `node`.

```

FROM public.ecr.aws/amazonlinux/amazonlinux:latest
RUN yum install -y shadow-utils && yum clean all
RUN useradd node
RUN mkdir -p /var/log/exported && chown node:node /var/log/exported
USER node
RUN touch /var/log/exported/examplefile
VOLUME ["/var/log/exported"]

```

2. Nella sezione `volumes` della definizione di attività, definire un volume con il nome `application_logs`.

```
"volumes": [  
  {  
    "name": "application_logs"  
  }  
]
```

3. Nella sezione `containerDefinitions`, crea le definizioni di container dell'applicazione. in modo che montino lo storage. Il valore `containerPath` deve corrispondere al percorso assoluto specificato nella direttiva `VOLUME` dal Dockerfile.

```
"containerDefinitions": [  
  {  
    "name": "application1",  
    "image": "my-repo/application",  
    "cpu": 100,  
    "memory": 100,  
    "essential": true,  
    "mountPoints": [  
      {  
        "sourceVolume": "application_logs",  
        "containerPath": "/var/log/exported"  
      }  
    ]  
  },  
  {  
    "name": "application2",  
    "image": "my-repo/application",  
    "cpu": 100,  
    "memory": 100,  
    "essential": true,  
    "mountPoints": [  
      {  
        "sourceVolume": "application_logs",  
        "containerPath": "/var/log/exported"  
      }  
    ]  
  }  
]
```

Fornire un volume di dati vuoto per un contenitore legato al ciclo di vita dell'istanza Amazon host EC2

Per le attività ospitate su EC2 istanze Amazon, puoi utilizzare bind mount e collegare i dati al ciclo di vita dell'istanza Amazon host. EC2 Puoi farlo utilizzando il parametro `host` e specificando un valore `sourcePath`. Tutti i file esistenti nel `sourcePath` vengono presentati ai container con il valore `containerPath`. Tutti i file scritti nel `containerPath` valore vengono scritti nel `sourcePath` valore sull' EC2 istanza Amazon host.

Important

Amazon ECS non sincronizza lo storage tra le EC2 istanze Amazon. Le attività che utilizzano lo storage persistente possono essere collocate su qualsiasi EC2 istanza Amazon del cluster con capacità disponibile. [Se le tue attività richiedono uno storage persistente dopo l'arresto e il riavvio, specifica sempre la stessa EC2 istanza Amazon al momento dell'avvio dell'attività con il AWS CLI comando `start-task`](#). Puoi utilizzare anche volumi Amazon EFS per l'archiviazione persistente. Per ulteriori informazioni, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).

1. Nella sezione `volumes` della definizione di attività, definisci un montaggio vincolato con i valori `name` e `sourcePath`. Nell'esempio seguente, l' EC2 istanza Amazon host contiene i dati `/ecs/webdata` che desideri montare all'interno del contenitore.

```
"volumes": [  
  {  
    "name": "webdata",  
    "host": {  
      "sourcePath": "/ecs/webdata"  
    }  
  }  
]
```

2. Nella sezione `containerDefinitions`, definisci un container con i valori `mountPoints` che faccia riferimento al nome del montaggio vincolato definito e al valore `containerPath` per montare il montaggio vincolato sul container.

```
"containerDefinitions": [  
  {  
    "name": "web",  
    "image": "public.ecr.aws/docker/library/nginx:latest",  
    "mountPoints": [  
      {  
        "containerPath": "/usr/share/nginx/html",  
        "sourcePath": "webdata"  
      }  
    ]  
  }  
]
```

```
"cpu": 99,
"memory": 100,
"portMappings": [
  {
    "containerPort": 80,
    "hostPort": 80
  }
],
"essential": true,
"mountPoints": [
  {
    "sourceVolume": "webdata",
    "containerPath": "/usr/share/nginx/html"
  }
]
}
```

Per montare un volume definito su più container in diverse posizioni

Puoi definire un volume di dati in una definizione di attività e montarlo in posizioni diverse su container diversi. Ad esempio, il container host ha una cartella di dati del sito Web in `/data/webroot`. Si potrebbe voler montare il volume di dati in sola lettura su due server Web diversi che hanno radici di documenti diverse.

1. Nella sezione `volumes` della definizione di attività, definisci un volume di dati con il nome `webroot` e il percorso di origine `/data/webroot`.

```
"volumes": [
  {
    "name": "webroot",
    "host": {
      "sourcePath": "/data/webroot"
    }
  }
]
```

2. Nella sezione `containerDefinitions`, definisci un container per ciascun server Web con i valori `mountPoints` che associano il volume `webroot` al valore `containerPath` puntando alla radice documento per tale container.

```
"containerDefinitions": [  
  {  
    "name": "web-server-1",  
    "image": "my-repo/ubuntu-apache",  
    "cpu": 100,  
    "memory": 100,  
    "portMappings": [  
      {  
        "containerPort": 80,  
        "hostPort": 80  
      }  
    ],  
    "essential": true,  
    "mountPoints": [  
      {  
        "sourceVolume": "webroot",  
        "containerPath": "/var/www/html",  
        "readOnly": true  
      }  
    ]  
  },  
  {  
    "name": "web-server-2",  
    "image": "my-repo/sles11-apache",  
    "cpu": 100,  
    "memory": 100,  
    "portMappings": [  
      {  
        "containerPort": 8080,  
        "hostPort": 8080  
      }  
    ],  
    "essential": true,  
    "mountPoints": [  
      {  
        "sourceVolume": "webroot",  
        "containerPath": "/srv/www/htdocs",  
        "readOnly": true  
      }  
    ]  
  }  
]
```

Per montare i volumi da un altro container mediante **volumesFrom**

Per le attività ospitate su EC2 istanze Amazon, puoi definire uno o più volumi su un contenitore e quindi utilizzare il `volumesFrom` parametro in una definizione di contenitore diversa all'interno della stessa attività per montare tutti i volumi dai punti di montaggio originariamente definiti. `sourceContainer` Il parametro `volumesFrom` si applica ai volumi configurati nella definizione di attività e a quelli integrati nell'immagine con un Dockerfile.

1. (Opzionale) Per condividere un volume incorporato in un'immagine, usa l'istruzione `VOLUME` nel Dockerfile. Il seguente Dockerfile di esempio utilizza un'immagine `httpd` e quindi aggiunge un volume e lo monta su `dockerfile_volume` nella radice del documento Apache. È la cartella utilizzata dal server Web `httpd`.

```
FROM httpd
VOLUME ["/usr/local/apache2/htdocs/dockerfile_volume"]
```

Puoi creare un'immagine con questo Dockerfile ed eseguirne il push a un repository, ad esempio Docker Hub, e utilizzarla nella definizione di attività. L'`my-repo/httpd-dockerfile_volume` immagine di esempio utilizzata nei passaggi seguenti è stata creata con il Dockerfile precedente.

2. Crea una definizione di attività che definisca gli altri volumi e punti di montaggio per i container. In questa sezione `volumes` di esempio, devi creare un volume vuoto denominato `empty`, gestito dal daemon Docker. Esiste anche un volume host definito che viene chiamato `host_etc`. Esporta la cartella `/etc` sull'istanza di container dell'host.

```
{
  "family": "test-volumes-from",
  "volumes": [
    {
      "name": "empty",
      "host": {}
    },
    {
      "name": "host_etc",
      "host": {
        "sourcePath": "/etc"
      }
    }
  ]
},
```

Nella sezione delle definizioni del container, crea un container che monti i volumi definiti in precedenza. In questo esempio, il container web monta i volumi empty e host_etc. Questo è il container che utilizza l'immagine creata con un volume nel Dockerfile.

```
"containerDefinitions": [  
  {  
    "name": "web",  
    "image": "my-repo/httpd_dockerfile_volume",  
    "cpu": 100,  
    "memory": 500,  
    "portMappings": [  
      {  
        "containerPort": 80,  
        "hostPort": 80  
      }  
    ],  
    "mountPoints": [  
      {  
        "sourceVolume": "empty",  
        "containerPath": "/usr/local/apache2/htdocs/empty_volume"  
      },  
      {  
        "sourceVolume": "host_etc",  
        "containerPath": "/usr/local/apache2/htdocs/host_etc"  
      }  
    ],  
    "essential": true  
  },  
],
```

Crea un altro container che utilizzi volumesFrom per montare tutti i volumi associati al container web. Tutti i volumi sul container web sono montati anche sul container busybox. È incluso il volume specificato nel Dockerfile utilizzato per creare l'immagine my-repo/httpd_dockerfile_volume.

```
{  
  "name": "busybox",  
  "image": "busybox",  
  "volumesFrom": [  
    {  
      "sourceContainer": "web"  
    }  
  ]  
}
```

```
    ],
    "cpu": 100,
    "memory": 500,
    "entryPoint": [
        "sh",
        "-c"
    ],
    "command": [
        "echo $(date) > /usr/local/apache2/htdocs/empty_volume/date && echo $(date) > /usr/local/apache2/htdocs/host_etc/date && echo $(date) > /usr/local/apache2/htdocs/dockerfile_volume/date"
    ],
    "essential": false
  }
]
```

Quando questa attività viene eseguita, i volumi vengono montati dai due container e il command nel container busybox scrive la data e l'ora su un file. Questo file è chiamato date in ciascuna cartella dei volumi, che diventano quindi visibili sul sito Web visualizzato dal container web.

Note

Il container busybox esegue un comando rapido e poi si chiude, quindi deve essere impostato come "essential": false nella definizione del container. In caso contrario, l'intera attività viene interrotta quando si chiude.

Gestione dello spazio di memoria di swap dei container su Amazon ECS

Con Amazon ECS, puoi controllare l'utilizzo dello spazio di memoria di swap sulle tue istanze EC2 Amazon basate su Linux a livello di contenitore. Utilizzando una configurazione swap per container, ogni container all'interno di una definizione di attività può avere lo swap abilitato o disabilitato. Per chi lo ha abilitato, la quantità massima di spazio di swap utilizzato può essere limitata. Ad esempio, i container critici per la latenza possono avere lo swap disabilitato. Al contrario, i contenitori con elevate richieste di memoria transitoria possono avere lo swap attivato per ridurre le possibilità di errori quando il contenitore è sotto carico. out-of-memory

La configurazione di swap per un container viene gestita dai seguenti parametri di definizione del container.

maxSwap

La quantità totale di memoria di swap (in MiB) che un container può utilizzare. Questo parametro viene tradotto nell' `--memory-swap` opzione `docker run` dove il valore è la somma della memoria del contenitore più il `maxSwap` valore.

Se viene specificato il valore `maxSwap` di 0, il container non utilizzerà lo swap. I valori accettati sono 0 o qualsiasi numero intero positivo. Se il parametro `maxSwap` viene omesso, il container utilizza la configurazione di swap per l'istanza di container su cui è in esecuzione. È necessario impostare un valore `maxSwap` per il parametro `swappiness` da utilizzare.

swappiness

In questo modo è possibile ottimizzare il comportamento `swappiness` di memoria di un container. Un valore `swappiness` di 0 fa sì che swap non si verifichi se non richiesto. Un valore `swappiness` di 100 produrrà lo swap delle pagine in modo aggressivo. I valori accettati sono numeri interi compresi tra 0 e 100. Se il parametro `swappiness` non è specificato, viene utilizzato un valore predefinito 60. Se non viene specificato un valore per `maxSwap`, questo parametro verrà ignorato. Questo parametro è mappato all' `--memory-swappiness` opzione `docker run`.

Nell'esempio seguente viene fornita la sintassi JSON.

```
"containerDefinitions": [{  
    ...  
    "linuxParameters": {  
        "maxSwap": integer,  
        "swappiness": integer  
    },  
    ...  
}]
```

Considerazioni

Quando utilizzi una configurazione swap container, considera quanto segue:

- Lo spazio di swap deve essere abilitato e allocato sull' EC2 istanza Amazon che ospita le tue attività affinché i contenitori possano essere utilizzati. Per impostazione predefinita, gli Amazon ECS ottimizzati AMIs non hanno lo swap abilitato. È necessario abilitare lo swap sull'istanza per utilizzare questa funzionalità. Per ulteriori informazioni, consulta [Instance Store Swap Volumes](#)

nella Amazon EC2 User Guide o [Come posso allocare la memoria per funzionare come spazio di swap in un'istanza Amazon?](#) EC2 .

- I parametri di definizione del contenitore dello spazio di scambio sono supportati solo per le definizioni delle attività che specificano il tipo di EC2 avvio. Questi parametri non sono supportati per le definizioni di attività destinate esclusivamente all'utilizzo di Amazon ECS su Fargate.
- Questa caratteristica è supportata solo per i container Linux. Al momento i container Windows non sono supportati.
- Se i parametri `maxSwap` e `swappiness` di definizione del container vengono omessi da una definizione di attività, ogni container avrà un valore `swappiness` predefinito di 60. Inoltre, l'utilizzo totale dello swap è limitato a due volte la memoria del contenitore.
- Se utilizzi le attività su Amazon Linux 2023, il parametro `swappiness` non è supportato.

Differenze nella definizione delle attività di Amazon ECS per il tipo di lancio di Fargate

Per utilizzare Fargate, è necessario configurare la definizione dell'attività in modo da utilizzare il tipo di avvio Fargate. Ci sono altre considerazioni sull'utilizzo di Fargate.

Parametri di definizione di attività

I processi che utilizzano il tipo di avvio Fargate non supportano tutti i parametri di definizione dei processi di Amazon ECS disponibili. Alcuni parametri non sono supportati, mentre altri si comportano diversamente con i processi Fargate.

I parametri di definizione dei processi seguenti non sono validi nei processi Fargate:

- `disableNetworking`
- `dnsSearchDomains`
- `dnsServers`
- `dockerSecurityOptions`
- `extraHosts`
- `gpu`
- `ipcMode`
- `links`
- `placementConstraints`

- `privileged`
- `maxSwap`
- `swappiness`

I seguenti parametri di definizione dei processi sono validi in Fargate, ma hanno delle limitazioni da considerare:

- `linuxParameters`: quando si indicano opzioni specifiche di Linux che vengono applicate al container, in `capabilities` l'unica capacità che puoi aggiungere è `CAP_SYS_PTRACE`. I parametri `devices`, `sharedMemorySize` e `tmpfs` non sono supportati. Per ulteriori informazioni, consulta [Parametri Linux](#).
- `volumes`: I processi Fargate supportano solo volumi host di montaggi vincolati, quindi il parametro `dockerVolumeConfiguration` non è supportato. Per ulteriori informazioni, consulta [Volumi](#).
- `cpu`- Per i contenitori Windows su AWS Fargate, il valore non può essere inferiore a 1 vCPU.
- `networkConfiguration`- Le attività Fargate utilizzano sempre la modalità di `awsvpc` rete.

Per assicurarti che la definizione di attività sia valida per l'utilizzo con Fargate, puoi specificare quanto riportato di seguito quando registri la definizione di attività:

- Nel campo Richiede compatibilità AWS Management Console, specificare. `FARGATE`
- Nel AWS CLI, specificare l'opzione. `--requires-compatibilities`
- Nell'API Amazon ECS, specifica il flag `requiresCompatibilities`.

Sistemi operativi e architetture

Quando si configura una definizione di task e container per AWS Fargate, è necessario specificare il sistema operativo utilizzato dal contenitore. I seguenti sistemi operativi sono supportati per AWS Fargate:

- Amazon Linux 2

Note

I contenitori Linux utilizzano solo il kernel e la configurazione del kernel del sistema operativo host. Ad esempio, la configurazione del kernel include i controlli di sistema `sysctl`. Un'immagine di container Linux può essere creata a partire da un'immagine di

base che contiene i file e i programmi di qualsiasi distribuzione Linux. Se l'architettura della CPU corrisponde, puoi eseguire i container da qualsiasi immagine di container Linux e su qualsiasi sistema operativo.

- Windows Server 2019 Full
- Windows Server 2019 Core
- Windows Server 2022 Full
- Windows Server 2022 Core

Quando si eseguono contenitori Windows su AWS Fargate, è necessario disporre dell'architettura della CPU X86_64.

Quando si eseguono contenitori Linux su AWS Fargate, è possibile utilizzare l'architettura della CPU X86_64 o l' ARM64 architettura per le applicazioni basate su ARM. Per ulteriori informazioni, consulta [the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”](#).

CPU e memoria del processo

Definizioni delle attività di Amazon ECS per AWS Fargate richiedono di specificare CPU e memoria a livello di attività. Sebbene per i processi Fargate sia possibile specificare CPU e memoria anche a livello di container, si tratta di un'opzione facoltativa. Per la maggior parte dei casi d'uso è necessario specificare queste risorse solo a livello di processo. La tabella seguente illustra le combinazioni valide di CPU e memoria a livello di attività. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in MiB o GB. Ad esempio, è possibile specificare un valore di memoria 3072 in MiB o 3 GB in GB. È possibile specificare i valori della CPU nel file JSON come stringa in unità CPU o in modalità virtuale CPUs (vCPUs). Ad esempio, è possibile specificare un valore di CPU come 1024 in unità CPU o 1 vCPU in vCPUs.

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
256 (0,25 vCPU)	512 MiB, 1 GB, 2 GB	Linux
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB	Linux
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB	Linux, Windows

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows
8192 (8 vCPU)	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
 Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.		
16384 (16vCPU)	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux
 Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.		

Reti di attività

Attività Amazon ECS per AWS Fargate richiedono la modalità `awsvpc` di rete, che fornisce a ogni attività un'interfaccia di rete elastica. Quando esegui un'attività o crei un servizio con questa modalità di rete, è necessario specificare una o più sottoreti per collegare l'interfaccia di rete e uno o più gruppi di sicurezza da applicare all'interfaccia di rete.

Se usi sottoreti pubbliche, decidi se fornire un indirizzo IP pubblico per l'interfaccia di rete. Affinché un processo Fargate in una sottorete pubblica estragga le immagini del container, è necessario che un indirizzo IP pubblico sia assegnato all'interfaccia di rete elastica del processo con un routing

a Internet o a un gateway NAT in grado di instradare le richieste a Internet. Affinché un processo Fargate in una sottorete privata estragga le immagini del container, è necessario un gateway NAT nella sottorete per instradare le richieste a Internet. Quando ospiti le immagini del container in Amazon ECR, puoi configurare Amazon ECR per utilizzare un endpoint VPC dell'interfaccia. In questo caso, l'IPv4 indirizzo privato dell'attività viene utilizzato per il pull dell'immagine. Per ulteriori informazioni sugli endpoint di interfaccia Amazon ECR, consulta [Endpoint VPC dell'interfaccia Amazon ECR \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.

Di seguito è riportato un esempio della `networkConfiguration` sezione relativa a un servizio Fargate:

```
"networkConfiguration": {
  "awsvpcConfiguration": {
    "assignPublicIp": "ENABLED",
    "securityGroups": [ "sg-12345678" ],
    "subnets": [ "subnet-12345678" ]
  }
}
```

Limiti delle risorse dei processi

Definizioni delle attività di Amazon ECS per contenitori Linux su AWS Fargate supporta il `ulimits` parametro per definire i limiti di risorse da impostare per un contenitore.

Definizioni delle attività di Amazon ECS per Windows su AWS Fargate non supportano il `ulimits` parametro per definire i limiti di risorse da impostare per un contenitore.

I processi di Amazon ECS ospitati su Fargate utilizzano i valori del limite di risorse predefinito impostato dal sistema operativo, ad eccezione del parametro del limite di risorse `nofile`. Il limite di risorse `nofile` imposta una restrizione sul numero di file aperti che un container può utilizzare. Su Fargate, il limite flessibile `nofile` predefinito è 65535 mentre il limite rigido è 65535. Puoi impostare i valori di entrambi i limiti fino a 1048576.

Di seguito è riportato uno snippet di definizione di attività di esempio che mostra come definire `nofile` limite che è stato raddoppiato:

```
"ulimits": [
  {
    "name": "nofile",
    "softLimit": 2048,
    "hardLimit": 8192
  }
]
```

```
}  
]
```

Per ulteriori informazioni sugli altri limiti di risorse che possono essere modificati, consulta [Limiti delle risorse](#).

Registrazione

Registrazione degli eventi

Amazon ECS registra le azioni eseguite. EventBridge Puoi utilizzare Amazon ECS Events per EventBridge ricevere notifiche quasi in tempo reale sullo stato corrente dei tuoi cluster, servizi e attività Amazon ECS. Inoltre, puoi automatizzare le operazioni per rispondere a questi eventi. Per ulteriori informazioni, consulta [Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge](#).

Registrazione del ciclo di vita delle attività

Le attività eseguite su Fargate pubblicano i timestamp per monitorare l'attività attraverso gli stati del ciclo di vita. Puoi visualizzare i timestamp nei dettagli dell'attività nel e descrivendo l'attività nella AWS Management Console e. AWS CLI SDKs Ad esempio, puoi utilizzare i timestamp per valutare il tempo impiegato dall'attività per scaricare le immagini di container e decidere se ottimizzare le dimensioni dell'immagine o utilizzare gli indici Seekable OCI. Per ulteriori informazioni sulle pratiche delle immagini di container, consulta [Best practice per le immagini dei container Amazon ECS](#).

Registrazione dell'applicazione

Definizioni delle attività di Amazon ECS per AWS Fargate supportano i driver `awslogssplunk`, e `awsfirelens log` per la configurazione del registro.

Il driver di `awslogs` registro configura le attività di Fargate per inviare informazioni di registro ad Amazon CloudWatch Logs. È riportato di seguito un frammento di una definizione di attività in cui è configurato il driver di log `awslogs`:

```
"logConfiguration": {  
  "logDriver": "awslogs",  
  "options": {  
    "awslogs-group" : "/ecs/fargate-task-definition",  
    "awslogs-region": "us-east-1",  
    "awslogs-stream-prefix": "ecs"  
  }  
}
```

```
}
```

Per ulteriori informazioni sull'utilizzo del driver di `awslogs` registro in una definizione di attività per inviare i log dei contenitori a Logs, consulta CloudWatch [Invia i log di Amazon ECS a CloudWatch](#)

Per ulteriori informazioni sul driver di log `awsfirelens`, consulta [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

Per ulteriori informazioni su come utilizzare il driver di log `splunk`, consulta [splunkdriver di registro](#).

Archiviazione dei processi

Per i processi Amazon ECS ospitati su Fargate, sono supportati i seguenti tipi di archiviazione:

- I volumi Amazon EBS forniscono storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Per ulteriori informazioni, consulta [Usa i volumi Amazon EBS con Amazon ECS](#).
- Volumi Amazon EFS per l'archiviazione persistente. Per ulteriori informazioni, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).
- Montaggi vincolati per l'archiviazione temporanea. Per ulteriori informazioni, consulta [Usa bind mount con Amazon ECS](#).

Caricamento lento delle immagini di container utilizzando Seekable OCI (SOCI)

Le attività di Amazon ECS su Fargate che utilizzano la versione della piattaforma Linux 1.4.0 possono utilizzare Seekable OCI (SOCI) per avviare le attività più velocemente. Con SOCI, i container trascorrono solo pochi secondi sul recupero dell'immagine prima di avviarsi, lasciando così il tempo necessario per la configurazione dell'ambiente e la creazione di istanze dell'applicazione mentre l'immagine viene scaricata in background. Questo processo si chiama caricamento lento. Quando Fargate avvia un'attività Amazon ECS, rileva automaticamente se esiste un indice SOCI per un'immagine nell'operazione e avvia il container senza attendere che l'intera immagine venga scaricata.

Per i container che funzionano senza indici SOCI, le immagini di container vengono scaricate completamente prima dell'avvio di quest'ultimo. Questo comportamento è lo stesso su tutte le altre versioni della piattaforma di Fargate e sull'AMI ottimizzata per Amazon ECS sulle istanze Amazon EC2

Indici Seekable OCI

Seekable OCI (SOCI) è una tecnologia open source sviluppata da Seekable OCI AWS che può avviare i container più velocemente caricando pigramente l'immagine del contenitore. SOCI funziona creando un indice (indice SOCI) dei file all'interno di un'immagine di container esistente. Questo indice aiuta ad avviare i container più velocemente, offrendo la possibilità di estrarre un singolo file da un'immagine di container prima di scaricare l'intera immagine. L'indice SOCI deve essere archiviato come artefatto nello stesso repository dell'immagine all'interno del registro dei container. Devi utilizzare solo indici SOCI provenienti da fonti attendibili, poiché l'indice è la fonte autorevole per il contenuto dell'immagine. Per ulteriori informazioni, consulta [Introduzione a Seekable OCI per il caricamento lento delle immagini di container](#).

Considerazioni

Se desideri che Fargate utilizzi un indice SOCI per caricare lentamente le immagini di container in un'attività, considera le seguenti informazioni:

- Solo le attività eseguite sulla versione della piattaforma Linux 1.4.0 possono utilizzare gli indici SOCI. Le attività che eseguono i container Windows su Fargate non sono supportate.
- Attività che vengono eseguite su X86_64 oppure ARM64 L'architettura della CPU è supportata.
- Le immagini di container nella definizione delle attività devono avere indici SOCI nello stesso registro di container dell'immagine.
- Le immagini di container nella definizione delle attività devono essere archiviate in un registro di immagini compatibile. Di seguito sono elencati i registri compatibili:
 - Registri privati Amazon ECR.
- Solo immagini di contenitori che utilizzano gzip Sono supportate la compressione o non sono compresse. Immagini del contenitore che utilizzano zstd la compressione non è supportata.
- Ti consigliamo di provare il caricamento lento con immagini del contenitore maggiori di 250 MiB di dimensioni compresse. È meno probabile che si verifichi una riduzione del tempo necessario per caricare immagini più piccole.
- Poiché il caricamento lento può modificare il tempo di avvio delle attività, potrebbe essere necessario modificare diversi timeout, ad esempio il periodo di tolleranza del controllo dell'integrità per Elastic Load Balancing.
- Se vuoi evitare che un'immagine di container venga caricata lentamente, elimina l'indice SOCI dal registro di container. Se l'immagine di container inclusa nell'operazione non soddisfa una delle considerazioni, tale immagine viene scaricata con il metodo predefinito.

Creazione di un indice Seekable OCI

Affinché un'immagine del contenitore venga caricata lentamente, è necessario un indice SOCI (un file di metadati) creato e archiviato nel repository di immagini del contenitore insieme all'immagine del contenitore. Per creare e inviare un indice SOCI puoi utilizzare lo strumento CLI [soci-snapshotter](#) open source. GitHub In alternativa, puoi implementare SOCI Index Builder. CloudFormation AWS Si tratta di una soluzione serverless che crea e invia automaticamente un indice SOCI quando un'immagine del contenitore viene inviata ad Amazon ECR. [Per ulteriori informazioni sulla soluzione e sui passaggi di installazione, consulta CloudFormation AWS SOCI Index Builder su GitHub](#) CloudFormation AWS SOCI Index Builder è un modo per automatizzare l'avvio a usare SOCI, mentre lo strumento open source soci offre una maggiore flessibilità nella generazione di indici e la capacità di integrare la generazione di indici nelle pipeline di integrazione continua e distribuzione continua (CI/CD).

Note

Affinché l'indice SOCI possa essere creato per un'immagine, l'immagine deve esistere in containerd archivio di immagini sul computer in esecuzione `soci-snapshotter`. Se l'immagine si trova in Docker archivio immagini, l'immagine non può essere trovata.

Verificare che un'attività abbia utilizzato il caricamento lento

Per verificare che sia stato utilizzato il caricamento lento per un'attività tramite SOCI, controlla l'endpoint dei metadati dell'operazione dall'interno dell'attività. Quando esegui query sulla versione 4 dell'endpoint metadati delle attività, viene visualizzato un campo `Snapshotter` nel percorso predefinito per il container da cui esegui la query. Inoltre, vengono visualizzati campi `Snapshotter` per ogni container nel percorso `/task`. Il valore predefinito per questo campo è `overlayfs`, e questo campo è impostato su `soci` se viene utilizzato SOCI.

Differenze nella definizione delle attività di Amazon ECS per le EC2 istanze che eseguono Windows

Le attività eseguite su istanze EC2 Windows non supportano tutti i parametri di definizione delle attività di Amazon ECS disponibili. Alcuni parametri non sono affatto supportati e altri si comportano in modo diverso.

I seguenti parametri di definizione delle attività non sono supportati per le definizioni delle attività di Amazon EC2 Windows:

- `containerDefinitions`

- `disableNetworking`
- `dnsServers`
- `dnsSearchDomains`
- `extraHosts`
- `links`
- `linuxParameters`
- `privileged`
- `readonlyRootFilesystem`
- `user`
- `ulimits`
- `volumes`
 - `dockerVolumeConfiguration`
- `cpu`

Consigliamo di specificare la CPU a livello di container per i container Windows.

- `memory`

Consigliamo di specificare la memoria a livello di container per i container Windows.

- `proxyConfiguration`
- `ipcMode`
- `pidMode`
- `taskRoleArn`

Le funzionalità dei ruoli IAM per le attività sulle istanze EC2 Windows richiedono una configurazione aggiuntiva, ma gran parte di questa configurazione è simile alla configurazione dei ruoli IAM per le attività sulle istanze di container Linux. Per ulteriori informazioni, consulta [the section called “Configurazione aggiuntiva dell'istanza Amazon EC2 Windows”](#).

Creazione di una definizione di attività Amazon ECS utilizzando la console

Crei una definizione di attività in modo da poter definire l'applicazione da eseguire come attività o servizio.

Quando si crea una definizione di attività per il tipo di avvio esterno, è necessario creare la definizione dell'attività utilizzando l'editor JSON e impostare il `requireCapabilities` parametro su. `EXTERNAL`

È possibile creare una definizione di attività utilizzando l'esperienza della console o specificando un file JSON.

Convalida JSON

L'editor JSON della console Amazon ECS verifica quanto segue nel file JSON:

- Il file è un file JSON valido.
- Il file non contiene chiavi estranee.
- Il file contiene il parametro. `familyName`
- C'è almeno una voce `sottocontainerDefinitions`.

AWS CloudFormation pile

Il seguente comportamento si applica alle definizioni delle attività create nella nuova console Amazon ECS prima del 12 gennaio 2023.

Quando crei una definizione di attività, la console Amazon ECS crea automaticamente uno CloudFormation stack con un nome che inizia con. `ECS-Console-V2-TaskDefinition-` Se hai utilizzato AWS CLI o un AWS SDK per annullare la registrazione della definizione dell'attività, devi eliminare manualmente lo stack di definizione dell'attività. Per ulteriori informazioni, consulta [Eliminazione di uno stack](#) nella Guida per l'utente.AWS CloudFormation

Per le definizioni delle attività create dopo il 12 gennaio 2023 non viene creato automaticamente uno CloudFormation stack per esse.

Procedura

Amazon ECS console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Nel menu Crea nuova definizione di attività, scegli Crea nuova definizione di attività.

4. Per Task definition family (Famiglia della definizione di attività) specifica un nome univoco per la definizione di attività.
5. Per Tipo di avvio, scegli l'ambiente dell'applicazione. L'impostazione predefinita della console è AWS Fargate (ovvero senza server). Amazon ECS utilizza questo valore per eseguire la convalida per garantire che i parametri di definizione delle attività siano validi per il tipo di infrastruttura.
6. In Operating system/Architecture (Sistema operativo/Architettura), scegli il sistema operativo e l'architettura della CPU per il processo.

Per eseguire l'attività su un'architettura ARM a 64 bit, scegli Linux/. ARM64 Per ulteriori informazioni, consulta [the section called “Piattaforma di runtime”](#).

Per eseguire le attività AWS Fargate sui container Windows, scegli un sistema operativo Windows supportato. Per ulteriori informazioni, consulta [the section called “Sistemi operativi e architetture”](#).

7. Per Task size (Dimensioni attività), specifica i valori di CPU e memoria da riservare per l'attività. Il valore della CPU è specificato come v CPUs e la memoria è specificata come GB.

Per i processi ospitati su Fargate, nella tabella seguente sono riportate le combinazioni valide di CPU e memoria.

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
256 (0,25 vCPU)	512 MiB, 1 GB, 2 GB	Linux
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB	Linux
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB	Linux, Windows
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
8192 (8 vCPU)  Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
16384 (16vCPU)  Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

Per le attività che utilizzano il tipo di avvio EC2 o esterno, i valori della CPU del task supportati sono compresi tra 128 unità CPU (0,125 vCPUs) e 196608 unità CPU (192 v) CPUs

Per specificare il valore della memoria in GB, inserisci GB dopo il valore. Ad esempio, per impostare il valore di memoria su 3 GB, immettere 3 GB.

 Note

I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows.

- In Network mode (Modalità di rete), scegli la modalità di rete da utilizzare. L'impostazione di default è la modalità awsvpc. Per maggiori informazioni, consulta [Networking attività Amazon ECS](#).

Se scegli bridge, in Port mappings, per Host port, inserisci il numero di porta sull'istanza del container da riservare per il container.

9. (Facoltativo) Espandi la sezione Task roles per configurare i ruoli AWS Identity and Access Management (IAM) per l'attività:
 - a. Per Task role (Ruolo attività), scegli il ruolo IAM da assegnare all'attività. Un ruolo IAM di task fornisce le autorizzazioni per i contenitori di un'attività per richiamare le operazioni AWS API.
 - b. Per Ruolo di esecuzione attività, scegli il ruolo.

Per informazioni sui casi in cui utilizzare il ruolo di esecuzione di attività, consulta [the section called “Ruolo IAM di esecuzione di attività”](#). Se non ti serve il ruolo, scegli Nessuno.

10. (Facoltativo) Espandi la sezione Posizionamento delle attività per aggiungere vincoli di posizionamento. I vincoli di posizionamento delle attività consentono di filtrare le istanze del contenitore utilizzate per il posizionamento delle attività utilizzando attributi incorporati o personalizzati.
11. (Facoltativo) Espandere la sezione Fault injection per abilitare l'iniezione dei guasti. L'iniezione di guasti consente di verificare in che modo l'applicazione risponde a determinati scenari di compromissione.
12. Per ogni container da definire nella tua definizione di attività, completa la procedura seguente.
 - a. In Name (Nome) immetti un nome per il container.
 - b. Per Image URI (URI immagine), specifica l'immagine da utilizzare per avviare un container. Le immagini nel registro Amazon ECR Public Gallery possono essere specificate utilizzando solo il nome del registro Amazon ECR Public. Ad esempio, se specificato, `public.ecr.aws/ecs/amazon-ecs-agent:latest` viene utilizzato il contenitore Amazon Linux ospitato nella Amazon ECR Public Gallery. Per tutti gli altri repository, specifica il repository utilizzando i `repository-url/image:tag` formati o `repository-url/image@digest`.
 - c. Se l'immagine si trova in un registro privato esterno ad Amazon ECR, in Registro privato, attiva Autenticazione del registro privato. Quindi, in ARN o nome di Gestione dei segreti, inserisci il nome della risorsa Amazon (ARN) del segreto.
 - d. Per Essential Container, se nella definizione dell'attività sono definiti due o più contenitori, è possibile specificare se il contenitore deve essere considerato essenziale.

Quando un contenitore è contrassegnato come Essenziale, se quel contenitore si ferma, l'attività viene interrotta. Ogni definizione di attività deve contenere almeno un container essenziale.

- e. La mappatura delle porte consente ai container di accedere alle porte sull'host per inviare o ricevere traffico. In Port mappings (Mappature di porte), esegui una delle operazioni seguenti:
- Quando utilizzi la modalità di rete awsvpc, in Container port (Porta del container) e Protocol (Protocollo), specifica la mappatura delle porte da utilizzare per il container.
 - Quando utilizzi la modalità di rete bridge, in Container port (Porta del container) e Protocol (Protocollo), specifica la mappatura delle porte da utilizzare per il container.

Scegli Add more port mappings (Aggiungi altre mappature porta) per specificare mappature aggiuntive delle porte del container.

- f. Per consentire al container l'accesso in sola lettura al file system root, per File system root di sola lettura, seleziona Sola lettura.
- g. (Facoltativo) Per definire i limiti di CPU, GPU e memoria a livello di contenitore diversi dai valori a livello di attività, in Limiti di allocazione delle risorse, procedi come segue:
- Per CPU, inserisci il numero di unità CPU che l'agente container Amazon ECS riserva per il contenitore.
 - Per GPU, inserisci il numero di unità di GPU per l'istanza di container.

Un' EC2 istanza Amazon con supporto GPU ha 1 unità GPU per ogni GPU. Per ulteriori informazioni, consulta [the section called “Definizioni delle attività per i carichi di lavoro GPU”](#).

- Per Limite rigido di memoria, inserisci la quantità di memoria, in GB, da presentare al contenitore. Se il container tenta di superare il limite rigido, si arresta.
- Il Docker Engine 20.10.0 o successivo riserva un minimo di 6 mebibyte (MiB) di memoria per un contenitore, quindi non specificare meno di 6 MiB di memoria per i tuoi contenitori.

Il Docker Engine 19.03.13-ce o precedente riserva un minimo di 4 MiB di memoria per un contenitore, quindi non specificare meno di 4 MiB di memoria per i contenitori.

- Per Limite di memoria flessibile, inserisci il limite flessibile (in GB) di memoria da prenotare per il container.

Quando la memoria di sistema è in conflitto, Docker tenta di mantenere la memoria del contenitore entro questo limite flessibile. Se non specifichi la memoria a livello di attività, devi indicare un numero intero diverso da zero per uno o entrambi i parametri Limite di memoria rigido e Limite di memoria flessibile. Se specifichi entrambi, Limite di memoria rigido deve essere maggiore di Limite di memoria flessibile.

Questa funzionalità non è supportata nei contenitori Windows.

- h. (Facoltativo) Espandi la sezione Variabili di ambiente per specificare le variabili di ambiente da inserire nel container. Puoi specificare le variabili di ambiente singolarmente utilizzando coppie chiave-valore o in blocco specificando un file di variabile di ambiente ospitato in un bucket Amazon S3. Per informazioni su come formattare un file di variabili di ambiente, consulta [Passa una singola variabile di ambiente a un contenitore Amazon ECS](#)

Quando specificate una variabile di ambiente per l'archiviazione segreta, in Key, immettete il nome segreto. Quindi ValueFrom, inserisci l'ARN completo del segreto di Systems Manager Parameter Store o del segreto di Secrets Manager

- i. (Facoltativo) Seleziona l'opzione Use log collection (Usa raccolta di log) per specificare una configurazione di log. Per ogni driver di log disponibile, sono disponibili opzioni del driver di log da specificare. L'opzione predefinita invia i log dei contenitori ad Amazon CloudWatch Logs. Le altre opzioni del driver di registro vengono configurate utilizzando AWS FireLens. Per ulteriori informazioni, vedere [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

Di seguito sono riportate descrizioni più dettagliate per ogni destinazione di log di container.

- Amazon CloudWatch: configura l'attività per inviare i log dei container a CloudWatch Logs. Vengono fornite le opzioni predefinite dei driver di registro, che creano un gruppo di CloudWatch log per tuo conto. Per specificare un nome del gruppo di log diverso, modifica i valori dell'opzione del driver.
- Esporta i log in Splunk— Configurare l'attività per inviare i log dei contenitori a Splunk driver che invia i log a un servizio remoto. Devi inserire l'URL del tuo Splunk servizio web. Il token è specificato come opzione segreta perché può essere trattato come dati sensibili.

- Esportazione dei log su Amazon Data Firehose: configura l'attività per inviare i log dei container a Firehose. Vengono fornite le opzioni predefinite del driver di registro, che inviano il log a un flusso di distribuzione Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
 - Esportazione dei log su Amazon Kinesis Data Streams: configura l'attività per inviare i log dei container a Kinesis Data Streams. Vengono fornite le opzioni predefinite del driver di registro, che inviano i log a un flusso Kinesis Data Streams. Per specificare un nome del flusso diverso, modifica i valori dell'opzione del driver.
 - Esporta i log su Amazon OpenSearch Service: configura l'attività per inviare i log dei container a un dominio di OpenSearch servizio. Devono essere fornite le opzioni del driver di log.
 - Esportazione dei log su Amazon S3: configura l'attività per inviare i log dei container a un bucket Amazon S3. Vengono fornite le opzioni di driver di registro predefinite, ma è necessario specificare un nome di bucket Amazon S3 valido.
- j. (Facoltativo) Configura i parametri del container aggiuntivi.

Per configurare questa opzione	Esegui questa operazione	
Politica di riavvio Queste opzioni definiscono una politica di riavvio per riavviare un contenitore all'uscita.	Espandi la politica di riavvio, quindi configura i seguenti elementi: • Per abilitare una politica di riavvio per il contenitore, attiva la politica Enable Restart. • Per i codici di uscita ignorati, specifica un elenco separato da virgole di codici di uscita interi del contenitore. Se il contenitore esce con uno dei codici di uscita specificati, Amazon ECS non tenterà di riavviarlo. Se non viene specificato nulla, Amazon ECS non ignorerà alcun codice di uscita. • Per il periodo di ripristino del tentativo, specifica un periodo di tempo intero, in secondi, per il quale il contenitore deve essere eseguito prima di poter tentare il riavvio in caso di	

Per configurare questa opzione	Esegui questa operazione	
	uscita. Amazon ECS può tentare di riavviare un container solo una volta ogni secondo del periodo di reimpostazione del tentativo. Se non viene specificato nulla, il contenitore deve funzionare per 300 secondi prima di poter tentare il riavvio.	

Per configurare questa opzione	Esegui questa operazione	
HealthCheck Questi sono i comandi che determinano se un contenitore è integro. Per ulteriori informazioni, consulta Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container.	HealthCheckEspandi e configura i seguenti elementi: • Per Command (Comando), inserisci un elenco di comandi separati da virgole. È possibile avviare i comandi con CMD per eseguire direttamente gli argomenti del comando oppure con CMD-SHELL per eseguire il comando con la shell (interprete di comandi) predefinita del container. Se non è specificato nessuno dei due, viene utilizzato CMD. • In Interval (Intervallo), inserisci il numero di secondi tra ogni controllo dell'integrità. I valori validi sono compresi tra 5 e 30. • For Timeout, specifica il periodo di tempo in secondi per cui attendere che un	

Per configurare questa opzione	Esegui questa operazione	
	controllo dell'integrità venga superato prima che sia considerato un errore. I valori validi sono compresi tra 2 e 60. • Per Start period (Periodo di inizio), inserisci l'intervallo di tempo (in secondi) da attendere per l'avvio di un container prima dell'esecuzione dei comandi di controllo dell'integrità. I valori validi sono compresi tra 0 e 300. • In Retries (Tentativi), inserisci il numero di volte in cui riprovare i comandi di controllo dell'integrità in caso di errore. I valori validi sono compresi tra 1 e 10.	

Per configurare questa opzione	Esegui questa operazione	
Ordinamento delle dipendenze all'avvio Questa opzione definisce le dipendenze per l'avvio e l'arresto del container. Un container può contenere più dipendenze.	Espandi Ordine delle dipendenze di avvio, quindi configura gli elementi seguenti: - Scegli Aggiungi dipendenza del container. - Per Container, seleziona il container. - Per Condizione, scegli la condizione di dipendenza di avvio. Per aggiungere un'ulteriore dipendenza, scegli Aggiungi dipendenza del container.	

Per configurare questa opzione	Esegui questa operazione	
Timeout container Queste opzioni determinano quando avviare e arrestare un container.	Espandi Timeout container, quindi configura gli elementi seguenti: • Per configurare il tempo di attesa prima di rinunciare alla risoluzione delle dipendenze per un contenitore, per Start timeout, inserisci il numero di secondi. • Per configurare il tempo di attesa prima che il contenitore si fermi se non esce normalmente da solo, per Stop timeout, inserisci il numero di secondi.	

Per configurare questa opzione	Esegui questa operazione	
Impostazioni di rete del container Queste opzioni determinano se utilizzare la rete all'interno di un container.	Espandi Impostazioni di rete del container, quindi configura gli elementi seguenti: • Per disabilitare la connessione di rete del container, seleziona Disattiva la connessione di rete. • Per configurare gli indirizzi IP del server DNS che vengono presentati al container, in Server DNS, inserisci l'indirizzo IP di ciascun server su una riga separata. • Per configurare i domini DNS per la ricerca nei nomi non-fully-qualified host presentati al contenitore, in DNS search domains, inserisci ogni dominio su una riga separata. Lo schema è <code>^[a-zA-Z0-9- .]{0,253}[a-zA-Z0-9]\$</code>. •	

Per configurare questa opzione	Esegui questa operazione	
	Per configurare il nome host del container, inserisci tale nome in Nome host. • Per aggiungere le mappature dei nomi host e degli indirizzi IP da allegare al file <code>/etc/hosts</code> nel container, scegli Aggiungi altro host, quindi in Nome host e indirizzo IP, inserisci i relativi valori.	

Per configurare questa opzione	Esegui questa operazione	
Docker configurazione Questi hanno la precedenza sui valori in Dockerfile.	Espandi Docker configurazione, quindi configura i seguenti elementi: Per Comando, inserisci un comando eseguibile per un container. Questo parametro è Cmd mappato alla sezione Crea un contenitore di Docker API remota e COMMAND opzione per <code>docker run</code>. Questo parametro sostituisce l'CMDistruzione in un Dockerfile. Per Punto di ingresso, inserire il Docker ENTRYPOINT che viene passato al contenitore. Questo parametro è Entrypoint mappato nella sezione Crea un contenitore del Docker API remota e <code>--entrypoint</code> opzione per <code>docker run</code>. Questo	

Per configurare questa opzione	Esegui questa operazione	
	parametro sostituisce l'ENTRYPOINT istruzione e in un Dockerfile. • Per Directory di lavoro, inserisci la directory in cui il container eseguirà ogni istruzione di punto di ingresso e di comando fornita. Questo parametro è WorkingDir mappato alla sezione Crea un contenuto re di Docker API remota e --workdir opzione per docker run . Questo parametro sostituisce l'WORKDIR istruzione in un Dockerfile.	

Per configurare questa opzione	Esegui questa operazione	
Limiti delle risorse (Ulimits) Questi valori sovrascrivono l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è Ulimits mappato alla sezione Crea un contenuto re dell'API Docker Remote e all'<code>--ulimit</code> opzione docker run.	Espandi i limiti delle risorse (ulimits), quindi scegli <code>Aggiungi ulimit</code>. Per Nome limite, scegli il limite. Quindi, per Limite flessibile e Limite rigido, inserisci i relativi valori. Per aggiungerne altri ulimits, scegli <code>Aggiungi ulimit</code>.	
Docker etichette Questa opzione aggiunge metadati al container. Questo parametro è Labels mappato alla sezione Crea un contenuto re dell'API Docker Remote e all'<code>--label</code> opzione docker run.	Espandi Docker etichette, scegli <code>Aggiungi coppia chiave-valore</code>, quindi inserisci la chiave e il valore. Per aggiungerne altre Docker etichette, scegli <code>Aggiungi coppia chiave-valore</code>.	

- k. (Facoltativo) Scegli `Add more containers` (Aggiungi altri container) per aggiungere altri container alla definizione del processo.

13. (Facoltativo) La sezione Archiviazione viene utilizzata per espandere la quantità di spazio di archiviazione temporaneo per le attività ospitate su Fargate. È inoltre possibile utilizzare questa sezione per aggiungere una configurazione del volume di dati per l'attività.

- Per espandere lo spazio di archiviazione temporaneo disponibile oltre il valore predefinito di 20 gibibytes (GiB) per le tue attività Fargate, in `Importo`, inserisci un valore fino a 200 GiB.

14. (Facoltativo) Per aggiungere una configurazione del volume di dati per la definizione dell'attività, scegliete Aggiungi volume, quindi seguite questi passaggi.
 - a. Per Volume name (Nome volume), immetti un nome per il volume di dati. Il nome del volume di dati viene utilizzato quando si crea un punto di montaggio del container.
 - b. Per Configurazione del volume, seleziona se desideri configurare il volume durante la creazione della definizione dell'attività o durante la distribuzione.

 Note

I volumi che possono essere configurati durante la creazione di una definizione di attività includono Bind mount, Docker, Amazon EFS e Amazon FSx per Windows File Server. I volumi che possono essere configurati al momento della distribuzione durante l'esecuzione di un'attività o durante la creazione o l'aggiornamento di un servizio includono Amazon EBS.

- c. Per Tipo di volume, seleziona un tipo di volume compatibile con il tipo di configurazione selezionato, quindi configura il tipo di volume.

Tipo di volume	Fasi	
Bind mount	a. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. b. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	

Tipo di volume	Fasi	
EFS	a. Per File system ID (ID del file system), scegli l'ID del file system Amazon EFS. b. (Facoltativo) Per Root directory (Directory root), specifica la directory all'interno del file system Amazon EFS da montare come directory root all'interno dell'host. Se questo parametro viene omissso, viene utilizzato a la radice del volume Amazon EFS. Se prevedi di utilizzare e un punto di accesso EFS, lascia vuoto questo campo. c. (Facoltativo) Per Access point (Punto di accesso), scegli l'ID del punto di accesso da utilizzare. d. (Facoltativo) Per crittografare i dati tra il file system Amazon EFS e l'host Amazon ECS o per utilizzare il ruolo di esecuzione delle attività durante il montaggio del	

Tipo di volume	Fasi	
	volume, scegli Advanced configurations (Configurazioni avanzate) e configura quanto segue: • Per crittografare i dati tra il file system Amazon EFS e l'host Amazon ECS, seleziona Transit encryption (Crittografia di transito) e quindi per Port (Porta), inserisci la porta da utilizzare e per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non si specifica una porta di crittografia di transito, verrà utilizzata la strategia di selezione della porta utilizzata dall'helper per il montaggio di Amazon EFS. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System. • Per utilizzare il ruolo IAM dell'attività	

Tipo di volume	Fasi	
	Amazon ECS riportato in una definizione di attività durante il montaggio del file system Amazon EFS, seleziona IAM authorization (Autorizzazione IAM). e. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume.	

Tipo di volume	Fasi	
	f. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
Docker	a. Per Driver, inserisci il Docker configurazione del volume. I contenitori Windows supportano solo l'uso del driver locale. Per utilizzare i montaggi vincolati, specifica un host. b. In Scope (Ambito), scegli il ciclo di vita del volume. • Per fare in modo che il ciclo di vita duri dall'inizio e all'interruzione dell'attività, scegli Task (Attività). • Per mantenere il volume dopo l'interruzione dell'attività, scegli Shared (Condiviso). c. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. •	

Tipo di volume	Fasi	
	Per Source volume (Volume sorgente) , scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. d. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
FSx per Windows File Server	- Per ID del file system, scegli FSx l'ID del file system Windows File Server. - Per la directory principale, immettete la directory , inserite FSx la directory all'interno del file system Windows File Server da montare come directory principale all'interno dell'host. - Per il parametro Credenziali, scegli il metodo di archiviazione delle credenziali. - Per utilizzarlo AWS Secrets Manager, inserisci l'Amazon Resource Name (ARN) di un segreto di Secrets Manager. - Per utilizzarlo AWS Systems Manager, inserisci l'Amazon Resource Name (ARN) di un parametro Systems Manager.	

Tipo di volume	Fasi	
	Per Dominio, inserisci il nome di dominio completo ospitato da una directory AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD) o da un Active Directory ospitato autonomamente EC2 . e. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente) , scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso	

Tipo di volume	Fasi	
	in sola lettura al volume. f. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
Amazon EBS	a. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. b. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

15. Per aggiungere un volume da un altro container, scegli Aggiungi volume da, quindi configura gli elementi seguenti:
 - Per Container, seleziona il container.
 - Per Origine, scegli il container contenente il volume da montare.
 - Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume.
16. (Facoltativo) Per configurare le impostazioni di tracciamento e raccolta delle metriche dell'applicazione utilizzando AWS Distro for OpenTelemetry integrazione, espandi Monitoraggio, quindi seleziona Usa la raccolta di metriche per raccogliere e inviare i parametri per le tue attività ad Amazon o CloudWatch Amazon Managed Service for Prometheus. Quando questa opzione è selezionata, Amazon ECS crea un AWS Distro for OpenTelemetry container sidecar preconfigurato per inviare i parametri dell'applicazione. Per ulteriori informazioni, consulta [Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni](#).
 - a. Quando Amazon CloudWatch è selezionato, i parametri delle tue applicazioni personalizzate vengono indirizzati a CloudWatch come metriche personalizzate. Per ulteriori informazioni, consulta [Esportazione dei parametri delle applicazioni su Amazon CloudWatch](#).

 Important

Quando si esportano i parametri delle applicazioni in Amazon CloudWatch, la definizione dell'attività richiede un ruolo IAM dell'attività con le autorizzazioni richieste. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon CloudWatch](#).

- b. Quando selezioni Amazon Managed Service for Prometheus (Prometheus libraries instrumentation) (Amazon Managed Service for Prometheus [strumentazione delle librerie Prometheus]), la CPU, la memoria, la rete e i parametri di archiviazione a livello di processo e i parametri delle applicazioni personalizzati vengono instradati ad Amazon Managed Service for Prometheus. Per l'endpoint di scrittura remota Workspace, inserisci l'URL dell'endpoint di scrittura remota del tuo Prometheus spazio di lavoro. Per Scraping target, inserisci l'host e porta il AWS Distro for OpenTelemetry collector può essere utilizzato per raccogliere i dati delle metriche. Per ulteriori informazioni, consulta [Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus](#).

 Important

Quando si esportano i parametri delle applicazioni in Amazon Managed Service for Prometheus, la definizione di attività richiede un ruolo IAM del processo con le autorizzazioni richieste. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon Managed Service for Prometheus](#).

- c. Quando selezioni Amazon Managed Service for Prometheus OpenTelemetry (strumentazione), i parametri di CPU, memoria, rete e storage a livello di attività e i parametri delle applicazioni personalizzate vengono indirizzati ad Amazon Managed Service for Prometheus. Per l'endpoint di scrittura remota Workspace, inserisci l'URL dell'endpoint di scrittura remota per il tuo Prometheus spazio di lavoro. Per ulteriori informazioni, consulta [Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus](#).

 Important

Quando si esportano i parametri delle applicazioni in Amazon Managed Service for Prometheus, la definizione di attività richiede un ruolo IAM del processo con le autorizzazioni richieste. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon Managed Service for Prometheus](#).

17. (Facoltativo) Espandi la sezione Tags (Tag) per aggiungere tag, come coppie chiave-valore, alla definizione di attività.
- [Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
 - [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).
18. Scegli Crea per registrare la definizione di attività.

Amazon ECS console JSON editor

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.

2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Nel menu Crea nuova definizione di attività, scegli Crea nuova definizione di attività con JSON.
4. Nella casella dell'editor JSON, modifica il tuo file JSON,

Il JSON deve superare i controlli di convalida specificati in [the section called “Convalida JSON”](#).

5. Scegli Create (Crea).

Aggiornamento di una definizione di attività Amazon ECS tramite la console

Una revisione della definizione di attività è una copia della definizione di attività corrente con i nuovi valori dei parametri che sostituiscono quelli esistenti. Tutti i parametri che non si modificano sono nella nuova revisione.

Per aggiornare una definizione di attività, devi creare una revisione della definizione di attività. Se la definizione di attività viene utilizzata all'interno di un servizio, devi aggiornare quest'ultimo per poter utilizzare la definizione di attività aggiornata.

Quando si crea una revisione, è possibile modificare le seguenti proprietà del container e le proprietà dell'ambiente.

- URI dell'immagine del container
- Mappature di porte
- Variabili di ambiente
- Dimensioni processo
- Dimensioni del container
- Ruolo del processo
- Ruolo di esecuzione di attività
- Volumi e punti di montaggio dei container
- Registro privato

Convalida JSON

L'editor JSON della console Amazon ECS verifica quanto segue nel file JSON:

- Il file è un file JSON valido
- Il file non contiene chiavi esterne
- Il file contiene il parametro `familyName`
- È presente almeno una voce in `containerDefinitions`

Procedura

Amazon ECS console

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Dalla barra di navigazione, scegli la Regione in cui si trova la definizione di attività.
3. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
4. Scegli la definizione di attività.
5. Seleziona la revisione della definizione di attività, quindi scegli Crea nuova revisione, Crea nuova revisione.
6. Nella pagina Create new task definition revision (Crea nuova revisione della definizione di attività), apporta le modifiche desiderate. Ad esempio, per modificare le definizioni del container esistenti (ad esempio l'immagine del container, i limiti di memoria o le mappature delle porte), seleziona il container e apporta le modifiche desiderate.
7. Verifica le informazioni e scegli Aggiorna.
8. Se la tua definizione di attività viene utilizzata all'interno di un servizio, devi aggiornare quest'ultimo con la definizione di attività aggiornata. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Amazon ECS console JSON editor

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new revision (Crea nuova revisione), Create new revision with JSON (Crea nuova revisione con JSON).

4. Nella casella dell'editor JSON, modifica il tuo file JSON,

Il JSON deve superare i controlli di convalida specificati in [the section called “Convalida JSON”](#).

5. Scegli Create (Crea).

Annullamento della registrazione di una revisione della definizione di attività di Amazon ECS tramite la console

È possibile annullare la registrazione della revisione della definizione dell'attività in modo che non venga più visualizzata nelle chiamate `ListTaskDefinition` API o nella console quando si desidera eseguire un'attività o aggiornare un servizio.

Quando revochi la registrazione della revisione di una definizione di attività, questa viene immediatamente contrassegnata come `INACTIVE`. Processi e servizi esistenti che fanno riferimento alla revisione di una definizione di attività `INACTIVE` continuano a essere eseguiti senza interruzioni. I servizi esistenti che fanno riferimento a una revisione di definizione di attività `INACTIVE` possono ancora essere ridotti o dimensionati orizzontalmente modificando il conteggio di servizi desiderato.

Non è possibile utilizzare una revisione di definizione di attività `INACTIVE` per eseguire nuovi processi o creare nuovi servizi. Non potrai aggiornare un servizio esistente per fare riferimento a una revisione della definizione di attività `INACTIVE` (anche se sono necessari circa 10 minuti dopo la revoca della registrazione affinché queste restrizioni vengano applicate).

Note

Quando annulli la registrazione di tutte le revisioni, la famiglia di definizioni delle attività viene spostata nell'elenco `INACTIVE`. L'aggiunta di una nuova revisione di una definizione dell'attività `INACTIVE` riporta la famiglia di definizioni delle attività nell'elenco `ACTIVE`. In questo momento, le revisioni delle definizioni di processi `INACTIVE` rimangono individuabili nel tuo account a tempo indeterminato. Tuttavia, questo comportamento è soggetto a modifiche in futuro. Pertanto, non fare affidamento a revisioni delle definizioni di processo `INACTIVE` che vanno oltre il ciclo di vita dei processi e dei servizi associati.

AWS CloudFormation pile

Il seguente comportamento si applica alle definizioni delle attività create nella nuova console Amazon ECS prima del 12 gennaio 2023.

Quando crei una definizione di attività, la console Amazon ECS crea automaticamente uno CloudFormation stack con un nome che inizia con `ECS-Console-V2-TaskDefinition-`. Se hai utilizzato AWS CLI o un AWS SDK per annullare la registrazione della definizione dell'attività, devi eliminare manualmente lo stack di definizione dell'attività. Per ulteriori informazioni, consulta [Eliminazione di uno stack](#) nella Guida per l'utente AWS CloudFormation.

Per le definizioni delle attività create dopo il 12 gennaio 2023 non viene creato automaticamente uno CloudFormation stack per esse.

Procedura

Per annullare la registrazione di una nuova definizione di attività (console Amazon ECS)

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Dalla barra di navigazione, scegli la regione in cui si trova la definizione di attività.
3. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
4. Nella pagina Task definitions (Definizioni di processi), scegli la famiglia di definizioni di attività contenente una o più revisioni per cui desideri annullare la registrazione.
5. Nella pagina Nome definizione di attività, seleziona le revisioni da eliminare, quindi seleziona Operazioni, Annulla registrazione.
6. Verifica le informazioni riportate nella finestra Deregister (Annulla registrazione), quindi scegli Deregister (Annulla registrazione) per terminare.

Eliminazione di una revisione della definizione di attività di Amazon ECS tramite la console

Quando non è più necessaria una revisione specifica della definizione dell'attività in Amazon ECS, puoi eliminare la revisione della definizione dell'attività.

Quando elimini una revisione di definizione di attività, questa passa immediatamente da `INACTIVE` a `DELETE_IN_PROGRESS`. L'esecuzione delle attività e dei servizi esistenti che fanno riferimento a una revisione di definizione di attività `DELETE_IN_PROGRESS` prosegue senza interruzioni.

Non è possibile utilizzare una revisione di definizione dell'attività `DELETE_IN_PROGRESS` per eseguire nuove attività o creare nuovi servizi. Inoltre, non è possibile aggiornare un servizio esistente per fare riferimento a una revisione di definizione di attività `DELETE_IN_PROGRESS`.

Quando elimini tutte le revisioni delle definizioni di attività nello stato `INACTIVE`, il nome della definizione di attività non viene visualizzato nella console e non viene restituito nell'API. Se `DELETE_IN_PROGRESS` lo stato di revisione della definizione dell'attività è in corso, il nome della definizione dell'attività viene visualizzato nella console e restituito nell'API. Il nome della definizione di attività viene mantenuto da Amazon ECS e la revisione viene incrementata durante la prossima creazione di una definizione di attività con tale nome.

Risorse Amazon ECS in grado di bloccare un'eliminazione

Una richiesta di eliminazione della definizione di attività non viene completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione dell'attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività autonome di Amazon ECS: la definizione dell'attività è necessaria affinché l'attività rimanga integra.
- Attività del servizio Amazon ECS: la definizione dell'attività è necessaria affinché l'attività rimanga integra.
- Distribuzioni di servizi Amazon ECS e set di attività: la definizione dell'attività è richiesta quando viene avviato un evento di scalabilità per una distribuzione o un set di attività di Amazon ECS.

Se la definizione dell'attività rimane invariata `DELETE_IN_PROGRESS`, puoi utilizzare la console o identificare e quindi interrompere AWS CLI le risorse che bloccano l'eliminazione della definizione dell'attività.

Eliminazione della definizione di attività dopo la rimozione della risorsa bloccata

Le seguenti regole si applicano dopo aver rimosso le risorse che bloccano l'eliminazione della definizione di attività:

- Attività Amazon ECS: dopo l'interruzione dell'attività, l'eliminazione della definizione di attività può richiedere fino a 1 ora.
- Distribuzioni di servizi Amazon ECS e set di attività: il completamento dell'eliminazione delle definizioni delle attività può richiedere fino a 24 ore dopo l'eliminazione della distribuzione o del set di attività.

Procedura

Per annullare le definizioni di attività (console Amazon ECS)

Prima di eliminare una revisione di definizione di attività, è necessario annullarne la registrazione. Per ulteriori informazioni, consulta [the section called “Annullamento della registrazione di una revisione di definizione dell'attività con la console”](#).

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Dalla barra di navigazione, scegli la regione in cui si trova la definizione di attività.
3. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
4. Nella pagina Definizioni di attività, seleziona la famiglia di definizioni di attività contenente una o più revisioni che desideri eliminare.
5. Nella pagina Nome di definizione dell'attività, seleziona le revisioni da eliminare, quindi scegli Azioni, Elimina.

Se l'opzione Elimina non è disponibile, è necessario annullare la registrazione della definizione dell'attività.

6. Verifica le informazioni nella casella di conferma dell'eliminazione, quindi scegli Elimina per terminare.

Casi d'uso per la definizione delle attività di Amazon ECS

Scopri di più su come scrivere definizioni di attività per vari AWS servizi e funzionalità.

A seconda del carico di lavoro, è necessario impostare determinati parametri di definizione delle attività. Inoltre, per il tipo di EC2 avvio, è necessario scegliere istanze specifiche progettate per il carico di lavoro.

Argomenti

- [Definizioni di attività Amazon ECS per carichi di lavoro GPU](#)
- [Definizioni delle attività di Amazon ECS per carichi di lavoro di transcodifica video](#)
- [Definizioni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron](#)
- [Definizioni delle attività di Amazon ECS per le istanze di deep learning](#)
- [Definizioni delle attività di Amazon ECS per carichi di lavoro ARM a 64 bit](#)

- [Invia i log di Amazon ECS a CloudWatch](#)
- [Inviare i log di Amazon ECS a un servizio o AWS Partner](#)
- [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#)
- [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#)
- [Trasferisci dati sensibili a un contenitore Amazon ECS](#)

Definizioni di attività Amazon ECS per carichi di lavoro GPU

Amazon ECS supporta i carichi di lavoro che utilizzano GPUs, quando crei cluster con istanze di container che supportano GPU. Le istanze di container EC2 basate su Amazon GPU che utilizzano i tipi di istanze p2, p3, p5, g3, g4 e g5 forniscono l'accesso a NVIDIA GPU. Per ulteriori informazioni, consulta [Linux Accelerated Computing Instances](#) nella guida Amazon EC2 Instance Types.

Amazon ECS fornisce un'AMI ottimizzata per GPU che dispone di driver del kernel NVIDIA pre-configurati e un runtime del GPU Docker. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Puoi designarne alcune nella definizione dell'attività da prendere GPU in considerazione a livello di contenitore. Amazon ECS pianifica le istanze di container disponibili che supportano GPU e collegano contenitori fisici GPU a contenitori appropriati per prestazioni ottimali.

Sono supportati i seguenti tipi di istanze EC2 basate su Amazon GPU. [Per ulteriori informazioni, consulta Istanze Amazon EC2 P2, Istanze Amazon EC2 P3, Istanze Amazon P4d, Istanze EC2 Amazon P5, Istanze EC2 Amazon G3, Istanze EC2 Amazon G4, Istanze Amazon G5 EC2, Istanze Amazon G6 e Istanze EC2 Amazon G6e. EC2 EC2](#)

Tipo di istanza	GPUs	Memoria GPU (GiB)	v CPUs	Memoria (GiB)
p3.2xlarge	1	16	8	61
p3.8xlarge	4	64	32	244
p3.16xlarge	8	128	64	488
p3dn.24xlarge	8	256	96	768
p4d.24xlarge	8	320	96	1152

Tipo di istanza	GPUs	Memoria GPU (GiB)	v CPUs	Memoria (GiB)
p5.48xlarge	8	640	192	2048
g3s.xlarge	1	8	4	30,5
g3.4xlarge	1	8	16	122
g3.8xlarge	2	16	32	244
g3.16xlarge	4	32	64	488
g4dn.xlarge	1	16	4	16
g4dn.2xlarge	1	16	8	32
g4dn.4xlarge	1	16	16	64
g4dn.8xlarge	1	16	32	128
g4dn.12xlarge	4	64	48	192
g4dn.16xlarge	1	16	64	256
g5.xlarge	1	24	4	16
g5.2xlarge	1	24	8	32
g5.4xlarge	1	24	16	64
g5.8xlarge	1	24	32	128
g5.16xlarge	1	24	64	256
g5.12xlarge	4	96	48	192
g5.24xlarge	4	96	96	384
g5.48xlarge	8	192	192	768
g6.x grande	1	24	4	16

Tipo di istanza	GPUs	Memoria GPU (GiB)	v CPUs	Memoria (GiB)
g6,2 x grande	1	24	8	32
g 6,4 x grande	1	24	16	64
g 6,8 x grande	1	24	32	128
g6,16.x grande	1	24	64	256
g6.12 x grande	4	96	48	192
g6,24 x grande	4	96	96	384
g6,48 x grande	8	192	192	768
g6. metallo	8	192	192	768
gr 6,4 x grande	1	24	16	128
g6e.xlarge	1	48	4	32
g6e.2x grande	1	48	8	64
g6e.4x grande	1	48	16	128
g6e.8x grande	1	48	32	256
g6e16.x grande	1	48	64	512
g6e12.x grande	4	192	48	384
g6e24.x grande	4	192	96	768
g6e48.x grande	8	384	192	1536
gr 6,8 x grande	1	24	32	256

Puoi recuperare l'ID Amazon Machine Image (AMI) per Amazon ECS ottimizzato per Amazon ECS AMIs interrogando l' AWS Systems Manager API Parameter Store. Utilizzando questo parametro,

non è necessario cercare manualmente le AMI ottimizzate per Amazon ECS. IDs Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere [GetParameter](#). Il principale IAM che utilizzi deve disporre dell'autorizzazione IAM `ssm:GetParameter` per recuperare i metadati dell'AMI ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended --region us-east-1
```

Considerazioni

Note

Il supporto per il tipo di famiglia di istanze g2 è obsoleto.

La famiglia di istanze p2 è supportata solo nelle versioni precedenti a 20230912 dell'AMI Amazon ECS ottimizzata per GPU. Per continuare a utilizzare le istanze p2, consulta [Operazioni da eseguire per utilizzare un'istanza P2](#).

Gli aggiornamenti locali dei driver NVIDIA/CUDA su entrambi i tipi di famiglie di istanze causano potenziali errori nel carico di lavoro della GPU.

Ti consigliamo di considerare quanto segue prima di iniziare a lavorare GPUs su Amazon ECS.

- I cluster possono contenere una combinazione di istanze di container GPU e non GPU.
- Puoi eseguire carichi di lavoro GPU su istanze esterne. Durante la registrazione di un'istanza esterna nel tuo cluster, assicurati che il flag `--enable-gpu` sia incluso nello script di installazione. Per ulteriori informazioni, consulta [Registrazione di un'istanza esterna in un cluster Amazon ECS](#).
- È necessario impostare `ECS_ENABLE_GPU_SUPPORT` su `true` nel file di configurazione dell'agente. Per ulteriori informazioni, consulta [the section called "Configurazione dell'agente del container"](#).
- Durante l'esecuzione di un'attività o la creazione di un servizio, puoi utilizzare gli attributi del tipo di istanza quando configuri i vincoli di posizionamento delle attività per garantire su quali istanze di container verrà avviata l'attività. In questo modo, puoi utilizzare in modo più efficiente le risorse. Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

L'esempio seguente avvia un'attività su un'istanza di container `g4dn.xlarge` nel cluster predefinito.

```
aws ecs run-task --cluster default --task-definition ecs-gpu-task-def \
  --placement-constraints type=memberOf,expression="attribute:ecs.instance-type ==
  g4dn.xlarge" --region us-east-2
```

- Per ogni container che ha un requisito di risorsa GPU specificato nella definizione di container, Amazon ECS imposta il runtime del container in modo che sia il runtime del container NVIDIA.
- Il runtime del container NVIDIA richiede per poter funzionare l'impostazione di alcune variabili di ambiente nel container. Per un elenco di queste variabili di ambiente, consulta [Configurazioni specializzate con Docker](#). Amazon ECS imposta il valore della variabile di NVIDIA_VISIBLE_DEVICES ambiente in modo che sia un elenco dei dispositivi GPU che IDs Amazon ECS assegna al contenitore. Le altre variabili di ambiente richieste non vengono impostate da, Amazon ECS. Quindi, assicurati che siano impostate dall'immagine del container o nella definizione di quest'ultimo.
- La famiglia del tipo di istanza p5 è supportata nella versione 20230929 e successive dell'AMI Amazon ECS ottimizzata per GPU.
- La famiglia del tipo di istanza g4 è supportata nella versione 20230913 e successive dell'AMI Amazon ECS ottimizzata per GPU. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#). Non è supportata nel flusso di lavoro Crea cluster nella console Amazon ECS. Per utilizzare questi tipi di istanze, devi utilizzare la EC2 console Amazon o l'API e registrare manualmente le istanze nel tuo cluster. AWS CLI
- Il tipo di istanza p4d.24xlarge funziona solo con CUDA 11 o versioni successive.
- L'AMI Amazon ECS ottimizzata per GPU IPv6 è abilitata, il che causa problemi durante l'utilizzo. yum Questo problema può essere risolto configurandolo per l'uso IPv4 con yum il seguente comando.

```
echo "ip_resolve=4" >> /etc/yum.conf
```

- Quando si crea un'immagine di container che non utilizza le immagini di base NVIDIA/CUDA, è necessario impostare la variabile di runtime del container NVIDIA_DRIVER_CAPABILITIES su uno dei seguenti valori:
 - utility,compute
 - all

Per informazioni su come impostare la variabile, consulta [Controllo del runtime del container NVIDIA](#) sul sito Web di NVIDIA.

- GPUs non sono supportati nei contenitori Windows.

Avvia un'istanza di contenitore GPU per Amazon ECS

Per utilizzare un'istanza GPU su Amazon ECS, devi creare un modello di avvio, un file di dati utente e avviare l'istanza.

È quindi possibile eseguire un'attività che utilizza una definizione di attività configurata per la GPU.

Utilizzo di un modello di avvio

È possibile creare un modello di lancio.

- Crea un modello di lancio che utilizzi l'ID AMI GPU ottimizzato per Amazon ECS per l'AMI. Per informazioni su come creare un modello di lancio, consulta [Creare un nuovo modello di lancio utilizzando i parametri definiti](#) nella Amazon EC2 User Guide.

Usa l'ID AMI del passaggio precedente per l'immagine Amazon Machine. Per informazioni su come specificare l'ID AMI con il parametro Systems Manager, consulta [Specificare un parametro Systems Manager in un modello di avvio](#) nella Amazon EC2 User Guide.

Aggiungi quanto segue ai dati utente nel modello di avvio. Sostituisci *cluster-name* con il nome del cluster.

```
#!/bin/bash
echo ECS_CLUSTER=cluster-name >> /etc/ecs/ecs.config;
echo ECS_ENABLE_GPU_SUPPORT=true >> /etc/ecs/ecs.config
```

Usa il AWS CLI

È possibile utilizzare il AWS CLI per avviare l'istanza del contenitore.

1. Crea un file denominato `userdata.toml`. Questo file viene utilizzato per i dati utente dell'istanza. Sostituisci *cluster-name* con il nome del cluster.

```
#!/bin/bash
echo ECS_CLUSTER=cluster-name >> /etc/ecs/ecs.config;
echo ECS_ENABLE_GPU_SUPPORT=true >> /etc/ecs/ecs.config
```

2. Esegui il seguente comando per ottenere l'ID AMI della GPU. Ti servirà per la fase successiva.


```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended --region us-east-1
```

3. Esegui il comando seguente per avviare l'istanza GPU. Ricordati di sostituire i seguenti parametri:

- Sostituiscilo *subnet* con l'ID della sottorete privata o pubblica in cui verrà avviata l'istanza.
- Sostituisci *gpu_ami* con l'ID AMI del passaggio precedente.
- Sostituisci *t3.large* con il tipo di istanza che desideri utilizzare.
- Sostituisci *region* con il codice regionale.

```
aws ec2 run-instances --key-name ecs-gpu-example \
  --subnet-id subnet \
  --image-id gpu_ami \
  --instance-type t3.large \
  --region region \
  --tag-specifications 'ResourceType=instance,Tags=[{Key=GPU,Value=example}]' \
  --user-data file:///userdata.toml \
  --iam-instance-profile Name=ecsInstanceRole
```

4. Esegui il comando seguente per verificare che l'istanza di container sia registrata nel cluster. Quando esegui questo comando, ricordati di sostituire i parametri seguenti:

- Sostituisci *cluster* con il nome del cluster.
- Sostituisci *region* con il tuo codice regionale.

```
aws ecs list-container-instances --cluster cluster-name --region region
```

Specificazione GPUs in una definizione di attività Amazon ECS

Per utilizzare l'istanza GPUs on a container e il runtime della GPU Docker, assicurati di indicare il numero di elementi richiesti dal contenitore GPUs nella definizione dell'attività. Man mano che GPUs vengono posizionati i contenitori che supportano il supporto, l'agente container di Amazon ECS aggiunge il numero desiderato di elementi fisici GPUs al contenitore appropriato. Il numero di contenitori GPUs riservati per tutti i contenitori di un'attività non può superare il numero di quelli

disponibili GPUs sull'istanza del contenitore su cui viene avviata l'attività. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

⚠ Important

Se i requisiti della GPU non sono specificati nella definizione di attività, il processo utilizzerà il runtime del Docker di default.

Di seguito viene visualizzato il formato JSON per i requisiti GPU in una definizione di attività:

```
{
  "containerDefinitions": [
    {
      ...
      "resourceRequirements" : [
        {
          "type" : "GPU",
          "value" : "2"
        }
      ],
    },
    ...
  ]
}
```

L'esempio seguente mostra la sintassi per un container Docker che specifica un requisito GPU. Questo contenitore ne utilizza due GPUs, esegue l'`nvidia-smi` e quindi esce.

```
{
  "containerDefinitions": [
    {
      "memory": 80,
      "essential": true,
      "name": "gpu",
      "image": "nvidia/cuda:11.0.3-base",
      "resourceRequirements": [
        {
          "type": "GPU",
          "value": "2"
        }
      ],
    },
  ],
}
```

```

    "command": [
      "sh",
      "-c",
      "nvidia-smi"
    ],
    "cpu": 100
  }
],
"family": "example-ecs-gpu"
}

```

Condividi GPUs

Quando vuoi condividere GPUs, devi configurare quanto segue.

1. Rimuovi i requisiti di risorse GPU dalle definizioni delle tue attività in modo che Amazon ECS non riservi quelle GPUs che devono essere condivise.
2. Aggiungi i seguenti dati utente alle tue istanze quando desideri condividerli. GPUs Ciò renderà nvidia il runtime del contenitore Docker predefinito sull'istanza del contenitore in modo che tutti i contenitori Amazon ECS possano utilizzare il. GPUs Per ulteriori informazioni, consulta [Esegui comandi all'avvio di un' EC2 istanza con immissione di dati utente](#) nella Amazon EC2 User Guide.

```

const userData = ec2.UserData.forLinux();
userData.addCommands(
  'sudo rm /etc/sysconfig/docker',
  'echo DAEMON_MAXFILES=1048576 | sudo tee -a /etc/sysconfig/docker',
  'echo OPTIONS="--default-ulimit nofile=32768:65536 --default-runtime nvidia" |'
  'sudo tee -a /etc/sysconfig/docker',
  'echo DAEMON_PIDFILE_TIMEOUT=10 | sudo tee -a /etc/sysconfig/docker',
  'sudo systemctl restart docker',
);

```

3. Imposta la variabile di NVIDIA_VISIBLE_DEVICES ambiente sul tuo contenitore. Puoi farlo specificando la variabile di ambiente nella definizione dell'attività. Per informazioni sui valori validi, consulta [GPU Enumeration nel sito di documentazione](#) di NVIDIA.

Operazioni da eseguire per utilizzare un'istanza P2

Per continuare a lavorare con le istanze P2, puoi utilizzare una delle seguenti opzioni.

È necessario modificare i dati utente dell'istanza per entrambe le opzioni. Per ulteriori informazioni, consulta [Esegui comandi all'avvio di un' EC2 istanza con immissione di dati utente](#) nella Amazon EC2 User Guide.

Usa l'ultima AMI ottimizzata per GPU supportata

Puoi utilizzare la versione 20230906 dell'AMI ottimizzata per GPU e aggiungere quanto segue ai dati utente dell'istanza.

Sostituire cluster-name con il nome del cluster.

```
#!/bin/bash
echo "exclude=*nvidia* *cuda*" >> /etc/yum.conf
echo "ECS_CLUSTER=cluster-name" >> /etc/ecs/ecs.config
```

Usa l'AMI ottimizzata per GPU più recente e aggiorna i dati utente

Puoi aggiungere i comandi seguenti ai dati utente dell'istanza. Questa operazione consente di disinstallare i driver Nvidia 535/Cuda12.2 e installare i driver Nvidia 470/Cuda11.4, correggendo la versione.

```
#!/bin/bash
yum remove -y cuda-toolkit* nvidia-driver-latest-dkms*
tmpfile=$(mktemp)
cat >$tmpfile <<EOF
[amzn2-nvidia]
name=Amazon Linux 2 Nvidia repository
mirrorlist=\$awsproto://\$amazonlinux.\$awsregion.\$awsdomain/\$releasever/amzn2-
nvidia/latest/\$basearch/mirror.list
priority=20
gpgcheck=1
gpgkey=https://developer.download.nvidia.com/compute/cuda/repos/rhel7/
x86_64/7fa2af80.pub
enabled=1
exclude=libglvnd-*
EOF

mv $tmpfile /etc/yum.repos.d/amzn2-nvidia-tmp.repo
yum install -y system-release-nvidia cuda-toolkit-11-4 nvidia-driver-latest-
dkms-470.182.03
yum install -y libnvidia-container-1.4.0 libnvidia-container-tools-1.4.0 nvidia-
container-runtime-hook-1.4.0 docker-runtime-nvidia-1
```

```
echo "exclude=*nvidia* *cuda*" >> /etc/yum.conf
nvidia-smi
```

Crea la tua AMI ottimizzata per GPU compatibile con P2

Puoi creare un'AMI Amazon ECS ottimizzata per GPU personalizzata e compatibile con le istanze P2, quindi avviare le istanze P2 utilizzando l'AMI.

1. Esegui il comando seguente per clonare `amazon-ecs-ami` repo.

```
git clone https://github.com/aws/amazon-ecs-ami
```

2. Imposta l'agente Amazon ECS richiesto e le versioni dell'AMI Amazon Linux di origine in `release.auto.pkrrvars.hcl` o `overrides.auto.pkrrvars.hcl`.
3. Esegui il comando seguente per creare un EC2 AMI privato compatibile con P2.

Sostituisci la regione con la regione dell'istanza.

```
REGION=region make al2keplergpu
```

4. Utilizza l'AMI con i seguenti dati utente dell'istanza per connetterti al cluster Amazon ECS.

Sostituire `cluster-name` con il nome del cluster.

```
#!/bin/bash
echo "ECS_CLUSTER=cluster-name" >> /etc/ecs/ecs.config
```

Definizioni delle attività di Amazon ECS per carichi di lavoro di transcodifica video

[Per utilizzare i carichi di lavoro di transcodifica video su Amazon ECS, registra le istanze Amazon.](#)

EC2 VT1 Dopo aver registrato queste istanze, puoi eseguire carichi di lavoro di transcodifica video live e pre-renderizzati come attività su Amazon ECS. EC2 VT1 Le istanze Amazon utilizzano schede di transcodifica multimediale Xilinx U30 per accelerare i carichi di lavoro di transcodifica video live e pre-renderizzati.

 Note

Per istruzioni su come eseguire carichi di lavoro di transcodifica video in container diversi da Amazon ECS, consulta la [documentazione di Xilinx](#).

Considerazioni

Prima di iniziare la distribuzione VT1 su Amazon ECS, considera quanto segue:

- I tuoi cluster possono contenere un mix di VT1 e non istanze. VT1
- Hai bisogno di un'applicazione Linux che utilizzi schede di transcodifica multimediale Xilinx U30 con codec accelerati AVC (H.264) e HEVC (H.265).

 Important

Le applicazioni che utilizzano altri codec potrebbero non avere prestazioni migliorate sulle istanze. VT1

- Solo un'attività di transcodifica può essere eseguita su una scheda U30. Ogni scheda ha due dispositivi ad essa associati. È possibile eseguire tante attività di transcodifica quante sono le schede disponibili per ciascuna istanza. VT1
- Durante l'esecuzione di un servizio o di un'attività autonoma, puoi utilizzare gli attributi del tipo di istanza quando configuri i vincoli di posizionamento delle attività. Ciò garantisce che l'attività venga avviata sull'istanza di container specificata. In questo modo è possibile utilizzare le risorse in modo efficace e che le attività relative ai carichi di lavoro di transcodifica video siano affidate alle istanze. VT1 Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

Nell'esempio seguente viene eseguita un'attività su una istanza vt1.3xlarge sul cluster default.

```
aws ecs run-task \  
  --cluster default \  
  --task-definition vt1-3xlarge-ffmpeg-processor \  
  --placement-constraints type=memberOf,expression="attribute:ecs.instance-type == vt1.3xlarge"
```

- Puoi configurare un container per utilizzare la scheda U30 specifica disponibile nell'istanza di container host. Puoi farlo usando il parametro `linuxParameters` e specificando i dettagli del dispositivo. Per ulteriori informazioni, consulta [Requisiti di definizione di attività](#).

Utilizzo di un VT1 AMI

Sono disponibili due opzioni per eseguire un'AMI su Amazon EC2 per le istanze di container Amazon ECS. La prima opzione è quella di utilizzare l'AMI ufficiale Xilinx su Marketplace AWS. La seconda opzione è quella di creare la propria AMI dal repository di esempio.

- [Offerte Xilinx](#) su AMIs Marketplace AWS
- Amazon ECS fornisce un repository di esempio che è possibile utilizzare per creare un'AMI per carichi di lavoro di transcodifica video. Questa AMI è dotata di driver Xilinx U30. Puoi trovare il repository che contiene gli script di Packer su [GitHub](#). Per ulteriori informazioni su Packer, consulta la [documentazione di Packer](#).

Requisiti di definizione di attività

Per eseguire container di transcodifica video su Amazon ECS, la definizione di attività deve contenere un'applicazione di transcodifica video che utilizza i codec accelerati H.264/AVC e H.265/HEVC. [È possibile creare un'immagine del contenitore seguendo i passaggi su Xilinx. GitHub](#)

La definizione dell'attività deve essere specifica per il tipo di istanza. I tipi di istanza sono 3xlarge, 6xlarge e 24xlarge. Per utilizzare i dispositivi Xilinx U30 specifici disponibili nell'istanza di container host, è necessario configurare un container. Puoi farlo usando il parametro `linuxParameters`. La tabella seguente descrive in dettaglio le schede e i dispositivi SoCs specifici per ogni tipo di istanza.

Tipo di istanza	v CPUs	RAM (GiB)	Schede acceleratore U30	Dispositivi XCU3 0 SoC indirizzabili	Percorsi dispositivi
vt1.3xlarge	12	24	1	2	/dev/dri/renderD128 ,/dev/dri/renderD129

Tipo di istanza	v CPUs	RAM (GiB)	Schede acceleratore U30	Dispositivi XCU3 0 SoC indirizzabili	Percorsi dispositivi
vt1.6xlarge	24	48	2	4	/dev/dri/ renderD12 8 ,/dev/ dri/ renderD12 9 ,/dev/ dri/ renderD13 0 ,/dev/ dri/ renderD13 1

Tipo di istanza	v CPUs	RAM (GiB)	Schede acceleratore U30	Dispositivi XCU3 0 SoC indirizzabili	Percorsi dispositivi
vt1.24xlarge	96	182	8	16	/dev/dri/ renderD12 8 ,/dev/ dri/ renderD12 9 ,/dev/ dri/ renderD13 0 ,/dev/ dri/ renderD13 1 ,/dev/ dri/ renderD13 2 ,/dev/ dri/ renderD13 3 ,/dev/ dri/ renderD13 4 ,/dev/ dri/ renderD13 5 ,/dev/ dri/ renderD13 6 ,/dev/ dri/ renderD13 7 ,/dev/ dri/ renderD13

Tipo di istanza	v CPUs	RAM (GiB)	Schede acceleratore U30	Dispositivi XCU3 0 SoC indirizzabili	Percorsi dispositivi
					8 ,/dev/dri/renderD13
					9 ,/dev/dri/renderD14
					0 ,/dev/dri/renderD14
					1 ,/dev/dri/renderD14
					2 ,/dev/dri/renderD14
					3

Important

Se la definizione dell'attività elenca i dispositivi che l' EC2 istanza non dispone, l'operazione non viene eseguita. Quando l'attività ha esito negativo, viene visualizzato il seguente messaggio di errore in `stoppedReason`: `CannotStartContainerError: Error response from daemon: error gathering device information while adding custom device "/dev/dri/renderD130": no such file or directory.`

Specificare la transcodifica video in una definizione di attività Amazon ECS

Nell'esempio seguente, viene fornita la sintassi utilizzata per la definizione di un'attività di un contenitore Linux su Amazon EC2 . Questa definizione di attività viene utilizzata per le immagini del container create seguendo la procedura fornita nella [documentazione di Xilinx](#). Se utilizzi questo

esempio, sostituisci `image` con la tua immagine e copia i tuoi file video nell'istanza della directory `/home/ec2-user`.

vt1.3xlarge

1. Crea un file di testo denominato `vt1-3xlarge-ffmpeg-linux.json` con il seguente contenuto.

```
{
  "family": "vt1-3xlarge-ffmpeg-processor",
  "requiresCompatibilities": ["EC2"],
  "placementConstraints": [
    {
      "type": "memberOf",
      "expression": "attribute:ecs.os-type == linux"
    },
    {
      "type": "memberOf",
      "expression": "attribute:ecs.instance-type == vt1.3xlarge"
    }
  ],
  "containerDefinitions": [
    {
      "entryPoint": [
        "/bin/bash",
        "-c"
      ],
      "command": ["/video/ecs_ffmpeg_wrapper.sh"],
      "linuxParameters": {
        "devices": [
          {
            "containerPath": "/dev/dri/renderD128",
            "hostPath": "/dev/dri/renderD128",
            "permissions": [
              "read",
              "write"
            ]
          },
          {
            "containerPath": "/dev/dri/renderD129",
            "hostPath": "/dev/dri/renderD129",
            "permissions": [
              "read",
```

```

        "write"
      ]
    }
  ],
  },
  "mountPoints": [
    {
      "containerPath": "/video",
      "sourceVolume": "video_file"
    }
  ],
  "cpu": 0,
  "memory": 12000,
  "image": "0123456789012.dkr.ecr.us-west-2.amazonaws.com/aws/xilinx-
xffmpeg",
  "essential": true,
  "name": "xilinx-ffmpeg"
}
],
"volumes": [
  {
    "name": "video_file",
    "host": {"sourcePath": "/home/ec2-user"}
  }
]
}

```

2. Registra la definizione dell'attività.

```
aws ecs register-task-definition --family vt1-3xlarge-ffmpeg-processor --cli-
input-json file://vt1-3xlarge-ffmpeg-linux.json --region us-east-1
```

vt1.6xlarge

1. Crea un file di testo denominato `vt1-6xlarge-ffmpeg-linux.json` con il seguente contenuto.

```

{
  "family": "vt1-6xlarge-ffmpeg-processor",
  "requiresCompatibilities": ["EC2"],
  "placementConstraints": [
    {

```

```

        "type": "memberOf",
        "expression": "attribute:ecs.os-type == linux"
    },
    {
        "type": "memberOf",
        "expression": "attribute:ecs.instance-type == vt1.6xlarge"
    }
],
"containerDefinitions": [
    {
        "entryPoint": [
            "/bin/bash",
            "-c"
        ],
        "command": ["/video/ecs_ffmpeg_wrapper.sh"],
        "linuxParameters": {
            "devices": [
                {
                    "containerPath": "/dev/dri/renderD128",
                    "hostPath": "/dev/dri/renderD128",
                    "permissions": [
                        "read",
                        "write"
                    ]
                },
                {
                    "containerPath": "/dev/dri/renderD129",
                    "hostPath": "/dev/dri/renderD129",
                    "permissions": [
                        "read",
                        "write"
                    ]
                },
                {
                    "containerPath": "/dev/dri/renderD130",
                    "hostPath": "/dev/dri/renderD130",
                    "permissions": [
                        "read",
                        "write"
                    ]
                },
                {
                    "containerPath": "/dev/dri/renderD131",
                    "hostPath": "/dev/dri/renderD131",

```

```

        "permissions": [
            "read",
            "write"
        ]
    },
    "mountPoints": [
        {
            "containerPath": "/video",
            "sourceVolume": "video_file"
        }
    ],
    "cpu": 0,
    "memory": 12000,
    "image": "0123456789012.dkr.ecr.us-west-2.amazonaws.com/aws/xilinx-
xffmpeg",
    "essential": true,
    "name": "xilinx-xffmpeg"
  }
],
"volumes": [
  {
    "name": "video_file",
    "host": {"sourcePath": "/home/ec2-user"}
  }
]
}

```

2. Registra la definizione dell'attività.

```
aws ecs register-task-definition --family vt1-6xlarge-xffmpeg-processor --cli-
input-json file://vt1-6xlarge-xffmpeg-linux.json --region us-east-1
```

vt1.24xlarge

1. Crea un file di testo denominato `vt1-24xlarge-ffmpeg-linux.json` con il seguente contenuto.

```

{
  "family": "vt1-24xlarge-xffmpeg-processor",
  "requiresCompatibilities": ["EC2"],

```

```

"placementConstraints": [
  {
    "type": "memberOf",
    "expression": "attribute:ecs.os-type == linux"
  },
  {
    "type": "memberOf",
    "expression": "attribute:ecs.instance-type == vt1.24xlarge"
  }
],
"containerDefinitions": [
  {
    "entryPoint": [
      "/bin/bash",
      "-c"
    ],
    "command": ["/video/ecs_ffmpeg_wrapper.sh"],
    "linuxParameters": {
      "devices": [
        {
          "containerPath": "/dev/dri/renderD128",
          "hostPath": "/dev/dri/renderD128",
          "permissions": [
            "read",
            "write"
          ]
        },
        {
          "containerPath": "/dev/dri/renderD129",
          "hostPath": "/dev/dri/renderD129",
          "permissions": [
            "read",
            "write"
          ]
        },
        {
          "containerPath": "/dev/dri/renderD130",
          "hostPath": "/dev/dri/renderD130",
          "permissions": [
            "read",
            "write"
          ]
        }
      ]
    }
  }
]

```

```
        "containerPath": "/dev/dri/renderD131",
        "hostPath": "/dev/dri/renderD131",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD132",
        "hostPath": "/dev/dri/renderD132",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD133",
        "hostPath": "/dev/dri/renderD133",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD134",
        "hostPath": "/dev/dri/renderD134",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD135",
        "hostPath": "/dev/dri/renderD135",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD136",
        "hostPath": "/dev/dri/renderD136",
        "permissions": [
            "read",
```



```
        "write"
      ]
    },
    {
      "containerPath": "/dev/dri/renderD137",
      "hostPath": "/dev/dri/renderD137",
      "permissions": [
        "read",
        "write"
      ]
    },
    {
      "containerPath": "/dev/dri/renderD138",
      "hostPath": "/dev/dri/renderD138",
      "permissions": [
        "read",
        "write"
      ]
    },
    {
      "containerPath": "/dev/dri/renderD139",
      "hostPath": "/dev/dri/renderD139",
      "permissions": [
        "read",
        "write"
      ]
    },
    {
      "containerPath": "/dev/dri/renderD140",
      "hostPath": "/dev/dri/renderD140",
      "permissions": [
        "read",
        "write"
      ]
    },
    {
      "containerPath": "/dev/dri/renderD141",
      "hostPath": "/dev/dri/renderD141",
      "permissions": [
        "read",
        "write"
      ]
    }
  ],
  {
```

```

        "containerPath": "/dev/dri/renderD142",
        "hostPath": "/dev/dri/renderD142",
        "permissions": [
            "read",
            "write"
        ]
    },
    {
        "containerPath": "/dev/dri/renderD143",
        "hostPath": "/dev/dri/renderD143",
        "permissions": [
            "read",
            "write"
        ]
    }
],
"mountPoints": [
    {
        "containerPath": "/video",
        "sourceVolume": "video_file"
    }
],
"cpu": 0,
"memory": 12000,
"image": "0123456789012.dkr.ecr.us-west-2.amazonaws.com/aws/xilinx-
xffmpeg",
"essential": true,
"name": "xilinx-ffmpeg"
}
],
"volumes": [
    {
        "name": "video_file",
        "host": {"sourcePath": "/home/ec2-user"}
    }
]
}

```

2. Registra la definizione dell'attività.

```
aws ecs register-task-definition --family vt1-24xlarge-ffmpeg-processor --cli-
input-json file://vt1-24xlarge-ffmpeg-linux.json --region us-east-1
```

Definizioni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron

Puoi registrare istanze [Amazon EC2 Trn1](#), [Amazon EC2 Inf1](#) e [Amazon EC2 Inf2](#) nei tuoi cluster per carichi di lavoro di machine learning.

[Le istanze Amazon EC2 Trn1 sono alimentate da AWS chip Trainium](#). Queste istanze offrono addestramento ad alte prestazioni e a basso costo per il machine learning nel cloud. Puoi addestrare un modello di inferenza di machine learning utilizzando un framework di machine learning con AWS Neuron su un'istanza Trn1. Quindi, puoi eseguire il modello su un'istanza Inf1 o un'istanza Inf2 per utilizzare l'accelerazione dei chip Inferentia. AWS

Le istanze Amazon EC2 Inf1 e Inf2 sono alimentate da chip [AWS Inferentia e forniscono inferenze](#) ad alte prestazioni e al minor costo nel cloud.

I modelli di machine learning vengono implementati in container utilizzando [AWS Neuron](#), ovvero un Software Developer Kit (SDK) specializzato. L'SDK è composto da un compilatore, un runtime e strumenti di profilazione che ottimizzano le prestazioni di apprendimento automatico dei chip di apprendimento automatico. AWS AWS Neuron supporta i più diffusi framework di machine learning come, e Apache. TensorFlow PyTorch MXNet

Considerazioni

Prima di iniziare a implementare Neuron su Amazon ECS, tieni presente quanto segue:

- I cluster possono contenere una combinazione di istanze Trn1, Inf1, Inf2 e altre.
- È necessaria un'applicazione Linux in un contenitore che utilizza un framework di apprendimento automatico che supporti Neuron. AWS

Important

Le applicazioni che utilizzano altri framework potrebbero non avere prestazioni migliorate sulle istanze Trn1, Inf1 e Inf2.

- Puoi eseguire soltanto un'attività di inferenza o di addestramento all'inferenza su ogni chip [AWS Trainium](#) o [AWS Inferentia](#). Per Inf1, ogni chip ne ha 4. NeuronCores Per Trn1 e Inf2 ogni chip ne ha 2. NeuronCores Puoi eseguire tante attività quanti sono i chip per ciascuna istanza Trn1, Inf1 e Inf2.

- Durante l'esecuzione di un servizio o di un'attività autonoma, puoi utilizzare gli attributi del tipo di istanza quando configuri i vincoli di posizionamento dell'attività. Ciò garantisce che l'attività venga avviata sull'istanza di container specificata. In questo modo puoi ottimizzare l'utilizzo complessivo delle risorse e garantire che le attività per i carichi di lavoro di inferenza si trovino nelle istanze Trn1, Inf1 e Inf2. Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

Nell'esempio seguente viene eseguita un'attività su una istanza Inf1.xlarge sul cluster default.

```
aws ecs run-task \
  --cluster default \
  --task-definition ecs-inference-task-def \
  --placement-constraints type=memberOf,expression="attribute:ecs.instance-type == Inf1.xlarge"
```

- I requisiti di risorse Neuron non possono essere definiti in una definizione di attività. Invece, configuri un contenitore per utilizzare chip AWS Trainium o AWS Inferentia specifici disponibili sull'istanza del contenitore host. Puoi farlo usando il parametro `linuxParameters` e specificando i dettagli del dispositivo. Per ulteriori informazioni, consulta [Requisiti di definizione di attività](#).

Usa l'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS

Amazon ECS fornisce un'AMI ottimizzata per Amazon ECS basata su Amazon Linux 2023 per carichi di lavoro AWS Trainium e AWS Inferentia. Viene fornito con i driver AWS Neuron e il runtime per Docker. Questa AMI semplifica l'esecuzione dei carichi di lavoro di inferenza di machine learning su Amazon ECS.

Ti consigliamo di utilizzare l'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS all'avvio delle istanze EC2 Amazon Trn1, Inf1 e Inf2.

Puoi recuperare l'attuale AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS AWS CLI utilizzando il comando seguente.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/neuron/recommended
```

L'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS è supportata nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)

- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)
- Europa (Stoccolma)
- Sud America (San Paolo)

Usa l'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS

Amazon ECS fornisce un'AMI ottimizzata per Amazon ECS basata su Amazon Linux 2 per carichi di lavoro AWS Trainium e AWS Inferentia. Viene fornito con i driver AWS Neuron e il runtime per Docker. Questa AMI semplifica l'esecuzione dei carichi di lavoro di inferenza di machine learning su Amazon ECS.

Ti consigliamo di utilizzare l'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS all'avvio delle istanze Amazon EC2 Trn1, Inf1 e Inf2.

Puoi recuperare l'attuale AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS utilizzando il comando AWS CLI seguente.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/inf/  
recommended
```

L'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS è supportata nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)
- Europa (Stoccolma)
- Sud America (San Paolo)

Requisiti di definizione di attività

Per distribuire Neuron su Amazon ECS, la definizione dell'attività deve contenere la definizione del contenitore per un contenitore predefinito che serve il modello di inferenza per TensorFlow. È fornito da AWS Deep Learning Containers. Questo contenitore contiene il runtime AWS Neuron e l'applicazione TensorFlow Serving. All'avvio, questo contenitore recupera il modello da Amazon S3, avvia TensorFlow Neuron Serving con il modello salvato e attende le richieste di previsione. Nell'esempio seguente, l'immagine del contenitore ha 1.15 e Ubuntu 18.04. TensorFlow È disponibile un elenco completo di Deep Learning Containers predefiniti ottimizzati per Neuron. GitHub Per ulteriori informazioni, consulta [Using AWS TensorFlow Neuron Serving](#).

```
763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference-neuron:1.15.4-neuron-py37-ubuntu18.04
```

In alternativa, puoi creare la tua immagine di container sidecar di Neuron. Per ulteriori informazioni, consulta [Tutorial: Neuron TensorFlow Serving](#) nella Guida per gli AWS Deep Learning AMIs sviluppatori.

La definizione di attività deve essere specifica per il tipo di istanza. È necessario configurare un contenitore per utilizzare dispositivi AWS Trainium o AWS Inferentia specifici disponibili sull'istanza del contenitore host. Puoi farlo usando il parametro `linuxParameters`. Nella tabella seguente vengono descritti in dettaglio i chip specifici per ogni tipo di istanza.

Tipo di istanza	v CPUs	RAM (GiB)	AWS chip acceleratori ML	Percorsi dispositivi
trn1.2xlarge	8	32	1	/dev/neuron0
trn1.32xlarge	128	512	16	/dev/neuron0 , /dev/neuron1 , /dev/neuron2 , /dev/neuron3 , /dev/neuron4 , /dev/neuron5 , /dev/neuron6 , /dev/neuron7 , /dev/neuron8 , /dev/neuron9 , /dev/neuron10 , /dev/neuron11 , /dev/neuron12 , /dev/neuron13 , /dev/neuron14 ,

Tipo di istanza	v CPUs	RAM (GiB)	AWS chip acceleratori ML	Percorsi disponibili
				/dev/neuron15
inf1.xlarge	4	8	1	/dev/neuron0
inf1.2xlarge	8	16	1	/dev/neuron0
inf1.6xlarge	24	48	4	/dev/neuron0 , /dev/neuron1 , /dev/neuron2 , /dev/neuron3

Tipo di istanza	v CPUs	RAM (GiB)	AWS chip acceleratori ML	Percorsi disponibili
inf1.24xlarge	96	192	16	/dev/neuron0 , /dev/neuron1 , /dev/neuron2 , /dev/neuron3 , /dev/neuron4 , /dev/neuron5 , /dev/neuron6 , /dev/neuron7 , /dev/neuron8 , /dev/neuron9 , /dev/neuron10 , /dev/neuron11 , /dev/neuron12 , /dev/neuron13 , /dev/neuron14 , /dev/neuron15
inf2.xlarge	8	16	1	/dev/neuron0
inf2.8xlarge	32	64	1	/dev/neuron0

Tipo di istanza	v CPUs	RAM (GiB)	AWS chip acceleratori ML	Percorsi disponibili
inf2.24xlarge	96	384	6	/dev/neuron0 , /dev/neuron1 , /dev/neuron2 , /dev/neuron3 , /dev/neuron4 , /dev/neuron5 ,
inf2.48xlarge	192	768	12	/dev/neuron0 , /dev/neuron1 , /dev/neuron2 , /dev/neuron3 , /dev/neuron4 , /dev/neuron5 , /dev/neuron6 , /dev/neuron7 , /dev/neuron8 , /dev/neuron9 , /dev/neuron10 , /dev/neuron11

Specificare l'apprendimento automatico di AWS Neuron in una definizione di attività Amazon ECS

Di seguito è riportato un esempio di definizione di attività Linux per `inf1.xlarge` che riporta la sintassi da utilizzare.

```

{
  "family": "ecs-neuron",
  "requiresCompatibilities": ["EC2"],
  "placementConstraints": [
    {
      "type": "memberOf",
      "expression": "attribute:ecs.os-type == linux"
    },
    {
      "type": "memberOf",
      "expression": "attribute:ecs.instance-type == inf1.xlarge"
    }
  ],
  "executionRoleArn": "${YOUR_EXECUTION_ROLE}",
  "containerDefinitions": [
    {
      "entryPoint": [
        "/usr/local/bin/entrypoint.sh",
        "--port=8500",
        "--rest_api_port=9000",
        "--model_name=resnet50_neuron",
        "--model_base_path=s3://amzn-s3-demo-bucket/resnet50_neuron"
      ],
      "portMappings": [
        {
          "hostPort": 8500,
          "protocol": "tcp",
          "containerPort": 8500
        },
        {
          "hostPort": 8501,
          "protocol": "tcp",
          "containerPort": 8501
        },
        {
          "hostPort": 0,
          "protocol": "tcp",
          "containerPort": 80
        }
      ],
      "linuxParameters": {
        "devices": [
          {

```

```

        "containerPath": "/dev/neuron0",
        "hostPath": "/dev/neuron0",
        "permissions": [
            "read",
            "write"
        ]
    },
    "capabilities": {
        "add": [
            "IPC_LOCK"
        ]
    },
    "cpu": 0,
    "memoryReservation": 1000,
    "image": "763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-  
inference-neuron:1.15.4-neuron-py37-ubuntu18.04",
    "essential": true,
    "name": "resnet50"
}
]
}

```

Definizioni delle attività di Amazon ECS per le istanze di deep learning

Per utilizzare i carichi di lavoro di deep learning su Amazon ECS, registra EC2 DL1 le istanze [Amazon](#) nei tuoi cluster. Le EC2 DL1 istanze Amazon sono alimentate dagli acceleratori Gaudi di Habana Labs (una società Intel). Usa l'SDK Habana SynapseAI per connetterti agli acceleratori Habana Gaudi. L'SDK supporta i più diffusi framework di machine learning e. TensorFlow PyTorch

Considerazioni

Prima di iniziare la distribuzione DL1 su Amazon ECS, considera quanto segue:

- I tuoi cluster possono contenere un mix di DL1 e non istanze. DL1
- Durante la creazione di un servizio o l'esecuzione di un'attività autonoma, puoi utilizzare gli attributi del tipo di istanza quando configuri i vincoli di posizionamento delle attività per assicurarti che l'attività venga avviata sull'istanza di container specificata. In questo modo si garantisce che le risorse vengano utilizzate in modo efficace e che le attività per i carichi di lavoro di deep learning si

svolgano sulle istanze. DL1 Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

Nell'esempio seguente viene eseguito un processo su una istanza `d11.24xlarge` sul cluster `default`.

```
aws ecs run-task \
  --cluster default \
  --task-definition ecs-dl1-task-def \
  --placement-constraints type=memberOf,expression="attribute:ecs.instance-type == d11.24xlarge"
```

Utilizzo di un DL1 AMI

Sono disponibili tre opzioni per eseguire un'AMI su EC2 DL1 istanze Amazon per Amazon ECS:

- Marketplace AWS AMIs [fornite da Habana qui](#).
- Habana Deep Learning AMIs fornito da Amazon Web Services. Poiché non è incluso, è necessario installare l'agente container Amazon ECS separatamente.
- Usa Packer per creare un'AMI personalizzata fornita dal [GitHub repository](#). Per ulteriori informazioni, consulta [la documentazione di Packer](#).

Specificare il deep learning in una definizione di attività Amazon ECS

Per eseguire i contenitori di deep learning accelerato di Habana Gaudi su Amazon ECS, la definizione dell'attività deve contenere la definizione del contenitore per un contenitore predefinito che serve il modello di deep learning per TensorFlow o che PyTorch utilizza Habana SynapseAI fornito da Deep Learning Containers. AWS

L'immagine seguente del contenitore ha 2.7.0 e Ubuntu 20.04. TensorFlow Viene mantenuto un elenco completo di Deep Learning Containers predefiniti ottimizzati per gli acceleratori Habana Gaudi. GitHub Per ulteriori informazioni, consulta [Habana Training Containers](#) (Container di training Habana).

```
763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-training-habana:2.7.0-hpu-py38-synapseai1.2.0-ubuntu20.04
```

Di seguito è riportato un esempio di definizione di attività per contenitori Linux su Amazon EC2, che mostra la sintassi da utilizzare. In questo esempio viene utilizzata un'immagine contenente l'Habana Labs System Management Interface Tool (HL-SMI) che trovi qui: vault.habana.ai/gaudi-docker/1.1.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.6.0:1.1.0-614

```
{
  "family": "dl-test",
  "requiresCompatibilities": ["EC2"],
  "placementConstraints": [
    {
      "type": "memberOf",
      "expression": "attribute:ecs.os-type == linux"
    },
    {
      "type": "memberOf",
      "expression": "attribute:ecs.instance-type == dl1.24xlarge"
    }
  ],
  "networkMode": "host",
  "cpu": "10240",
  "memory": "1024",
  "containerDefinitions": [
    {
      "entryPoint": [
        "sh",
        "-c"
      ],
      "command": ["hl-smi"],
      "cpu": 8192,
      "environment": [
        {
          "name": "HABANA_VISIBLE_DEVICES",
          "value": "all"
        }
      ],
      "image": "vault.habana.ai/gaudi-docker/1.1.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.6.0:1.1.0-614",
      "essential": true,
      "name": "tensorflow-installer-tf-hpu"
    }
  ]
}
```

```
}
```

Definizioni delle attività di Amazon ECS per carichi di lavoro ARM a 64 bit

Amazon ECS supporta l'utilizzo di applicazioni ARM a 64 bit. [Puoi eseguire le tue applicazioni sulla piattaforma basata su processori Graviton.AWS](#) adatti per un'ampia varietà di carichi di lavoro, tra cui server applicativi, micro-servizi, elaborazione ad alte prestazioni, inferenza di machine learning basata su CPU, codifica video, automazione della progettazione elettronica, giochi, database open-source e cache in memoria.

Considerazioni

Prima di iniziare a implementare le definizioni di attività che utilizzano l'architettura ARM a 64 bit, tieni in considerazione quanto segue:

- Le applicazioni possono utilizzare i tipi Fargate o EC2 Launch.
- Le applicazioni possono utilizzare solo il sistema operativo Linux.
- Per il tipo Fargate, le applicazioni devono utilizzare la versione 1.4.0 o versione successiva della piattaforma Fargate.
- Le applicazioni possono utilizzare Fluent Bit o CloudWatch per il monitoraggio.
- Per il tipo di lancio Fargate, quanto segue Regioni AWS non supporta i carichi di lavoro ARM a 64 bit:
 - Stati Uniti orientali (Virginia settentrionale), la zona di disponibilità use1-az3
- Per il tipo di EC2 avvio di Amazon, consulta quanto segue per verificare che la regione in cui ti trovi supporti il tipo di istanza che desideri utilizzare:
 - [Istanze Amazon EC2 M6g](#)
 - [Istanze Amazon EC2 T4g](#)
 - [Istanze Amazon EC2 C6g](#)
 - [Istanze Amazon EC2 R6gd](#)
 - [Istanze Amazon EC2 X2gd](#)

Puoi anche utilizzare il `EC2 describe-instance-type-offerings` comando Amazon con un filtro per visualizzare l'offerta di istanze per la tua regione.

```
aws ec2 describe-instance-type-offerings --filters Name=instance-  
type,Values=instance-type --region region
```

Nell'esempio seguente viene verificata la disponibilità del tipo di istanza M6 nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1).

```
aws ec2 describe-instance-type-offerings --filters "Name=instance-type,Values=m6*" --region us-east-1
```

Per ulteriori informazioni, consulta [describe-instance-type-offerings](#) Amazon EC2 Command Line Reference.

Specificazione dell'architettura ARM in una definizione di attività Amazon ECS

Per utilizzare l'architettura ARM, specifica ARM64 per il parametro di definizione di attività `cpuArchitecture`.

Nell'esempio seguente, l'architettura ARM viene specificata in una definizione di attività. È in formato JSON.

```
{
  "runtimePlatform": {
    "operatingSystemFamily": "LINUX",
    "cpuArchitecture": "ARM64"
  },
  ...
}
```

Nell'esempio seguente, viene riportata una definizione di attività per l'architettura ARM che visualizza "hello world".

```
{
  "family": "arm64-testapp",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "arm-container",
      "image": "public.ecr.aws/docker/library/busybox:latest",
      "cpu": 100,
      "memory": 100,
      "essential": true,
      "command": [ "echo hello world" ],
      "entryPoint": [ "sh", "-c" ]
    }
  ]
}
```



```
    }  
  ],  
  "requiresCompatibilities": [ "EC2" ],  
  "cpu": "256",  
  "memory": "512",  
  "runtimePlatform": {  
    "operatingSystemFamily": "LINUX",  
    "cpuArchitecture": "ARM64"  
  },  
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"  
}
```

Invia i log di Amazon ECS a CloudWatch

È possibile configurare i contenitori delle attività per inviare informazioni di registro ai CloudWatch registri. Se utilizzi il tipo di avvio Fargate per le tue attività, puoi visualizzare i log provenienti dai tuoi container. Se utilizzi il tipo di EC2 avvio, puoi visualizzare diversi log dei tuoi contenitori in un'unica comoda posizione ed evitare che i log dei contenitori occupino spazio su disco sulle istanze del contenitore.

Note

Il tipo di informazioni registrate dai container nell'attività dipende per lo più dal relativo comando ENTRYPOINT. Per impostazione predefinita, i log acquisiti mostrano l'output di comando che normalmente viene visualizzato in un terminale interattivo se si esegue il container in locale, ovvero i flussi I/O STDOUT e STDERR. Il driver di awslogs registro passa semplicemente questi log da Docker a Logs. CloudWatch Per ulteriori informazioni su come vengono elaborati i log Docker, inclusi metodi alternativi per acquisire diversi flussi o dati di file, consulta l'articolo relativo alla [visualizzazione di log per un container o servizio](#) nella documentazione di Docker.

Per inviare i log di sistema dalle istanze di container Amazon ECS a CloudWatch Logs, consulta [Monitoring Log Files](#) e [CloudWatch Logs quote nella Amazon Logs](#) User Guide. CloudWatch

Tipo di avvio di Fargate

Se utilizzi il tipo di avvio Fargate per le tue attività, per attivare il driver di log logConfiguration devi aggiungere i parametri awslogs necessari alla definizione di attività. Per ulteriori informazioni, consulta [Esempio di definizione di attività Amazon ECS: indirizza i log verso CloudWatch](#).

Per il contenitore Windows su Fargate, eseguite una delle seguenti opzioni quando uno dei parametri di definizione dell'attività contiene caratteri speciali come: & \ < > ^ |

- Aggiungi un escape (\) con virgolette doppie attorno all'intera stringa di parametri

Esempio

```
"awslogs-multiline-pattern": "\"^[|DEBUG|INFO|WARNING|ERROR\"",
```

- Aggiungi un carattere escape (^) attorno a ogni carattere speciale

Esempio

```
"awslogs-multiline-pattern": "^[^|DEBUG^|INFO^|WARNING^|ERROR",
```

EC2 tipo di lancio

Se utilizzi il tipo di EC2 avvio per le tue attività e desideri attivare il driver di awslogs registro, le istanze del contenitore Amazon ECS richiedono almeno la versione 1.9.0 dell'agente contenitore. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Note

È necessario utilizzare un'AMI ottimizzata per Amazon ECS o un'AMI personalizzata con almeno la versione 1.9.0-1 del ecs-init pacchetto. Quando utilizzi un'AMI personalizzata, devi specificare che il driver di awslogs registrazione è disponibile sull' EC2 istanza Amazon all'avvio dell'agente utilizzando la seguente variabile di ambiente nell'docker runistruzione o nel file della variabile di ambiente.

```
ECS_AVAILABLE_LOGGING_DRIVERS=["json-file","awslogs"]
```

Le istanze di container di Amazon ECS richiedono inoltre l'autorizzazione logs:CreateLogStream e logs:PutLogEvents nel ruolo IAM con cui avvii le istanze di container. Se hai creato il ruolo dell'istanza di container di Amazon ECS prima che fosse abilitato il supporto per il driver di log awslogs in Amazon ECS, potrebbe essere necessario aggiungere questa autorizzazione. ecsTaskExecutionRole viene utilizzato quando viene assegnato all'attività e deve contenere le

autorizzazioni corrette. Per informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#). Se nelle istanze di container viene utilizzata l'apposita policy IAM gestita, le tue istanze di container dispongono delle autorizzazioni corrette. Per informazioni sulla politica IAM gestita per le istanze di container, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

Esempio di definizione di attività Amazon ECS: indirizza i log verso CloudWatch

Prima che i contenitori possano inviare i log a CloudWatch, è necessario specificare il driver di `awslogs` registro per i contenitori nella definizione dell'attività. Per ulteriori informazioni sui parametri di registro, vedere [Archiviazione e registrazione](#)

La definizione di attività JSON che segue ha un oggetto `logConfiguration` specificato per ogni container. Uno è per il WordPress contenitore che invia i log a un gruppo di log chiamato `awslogs-wordpress`. L'altro è per un container MySQL che invia i log a un gruppo di log chiamato `awslogs-mysql`. Entrambi i container utilizzano il prefisso `awslogs-example` per il flusso di log.

```
{
  "containerDefinitions": [
    {
      "name": "wordpress",
      "links": [
        "mysql"
      ],
      "image": "public.ecr.aws/docker/library/wordpress:latest",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80
        }
      ],
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-create-group": "true",
          "awslogs-group": "awslogs-wordpress",
          "awslogs-region": "us-west-2",
          "awslogs-stream-prefix": "awslogs-example"
        }
      },
      "memory": 500,
    }
  ]
}
```

```

        "cpu": 10
    },
    {
        "environment": [
            {
                "name": "MYSQL_ROOT_PASSWORD",
                "value": "password"
            }
        ],
        "name": "mysql",
        "image": "public.ecr.aws/docker/library/mysql:latest",
        "cpu": 10,
        "memory": 500,
        "essential": true,
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-create-group": "true",
                "awslogs-group": "awslogs-mysql",
                "awslogs-region": "us-west-2",
                "awslogs-stream-prefix": "awslogs-example",
                "mode": "non-blocking",
                "max-buffer-size": "25m"
            }
        }
    }
],
"family": "awslogs-example"
}

```

Passaggi successivi

- Facoltativamente, puoi impostare una politica di conservazione per il gruppo di log utilizzando l'API CloudWatch AWS CLI o. Per ulteriori informazioni, consulta [put-retention-policy](#) nella documentazione di riferimento AWS Command Line Interface .
- Dopo aver registrato una definizione di attività con il driver di awslogs registro in una configurazione di registro delle definizioni dei contenitori, è possibile eseguire un'attività o creare un servizio con tale definizione di attività per iniziare a inviare i log ai CloudWatch registri. Per ulteriori informazioni, consultare [Esecuzione di un'applicazione come attività Amazon ECS](#) e [Creazione di un servizio Amazon ECS utilizzando la console](#).

Inviare i log di Amazon ECS a un servizio o AWS Partner

Puoi utilizzare Amazon ECS FireLens per utilizzare i parametri di definizione delle attività per indirizzare i log verso un AWS servizio o una destinazione AWS Partner Network (APN) per l'archiviazione e l'analisi dei log. AWS Partner Network È una comunità globale di partner che sfrutta programmi, competenze e risorse per creare, commercializzare e vendere offerte ai clienti. Per ulteriori informazioni, consulta [AWS Partner](#). FireLens funziona con [Fluentd](#) e [Fluent Bit](#). Forniamo AWS il Fluent Bit immagine o puoi usare la tua Fluentd oppure Fluent Bit immagine.

Per impostazione predefinita, Amazon ECS configura la dipendenza del contenitore in modo che il contenitore FireLens si avvii prima di qualsiasi contenitore che lo utilizza. Il contenitore FireLens si ferma anche dopo che tutti i contenitori che lo utilizzano si fermano.

Considera quanto segue quando utilizzi FireLens Amazon ECS:

- Ti consigliamo di `my_service_` aggiungere il nome del contenitore di log in modo da poter distinguere facilmente i nomi dei contenitori nella console.
- Amazon ECS aggiunge una dipendenza dall'ordine iniziale dei container tra i contenitori delle applicazioni e il FireLens contenitore per impostazione predefinita. Quando si specifica un ordine dei contenitori tra i contenitori dell'applicazione e il FireLens container, quindi l'ordine iniziale del contenitore predefinito viene sovrascritto.
- FireLens per Amazon ECS è supportato per le attività ospitate sia su Linux che AWS Fargate su Amazon EC2 su Linux. I contenitori Windows non supportano FireLens.

Per informazioni su come configurare la registrazione centralizzata per i container di Windows, consulta [Registrazione centralizzata per container di Windows su Amazon ECS tramite Fluent Bit](#).

- È possibile utilizzare AWS CloudFormation modelli per configurare FireLens per Amazon ECS. Per ulteriori informazioni, consulta la [AWS::ECS::TaskDefinition FireLensConfiguration](#) Guida per l'AWS CloudFormation utente
- FireLens ascolta sulla porta 24224, quindi per garantire che il FireLens log router non sia raggiungibile al di fuori dell'operazione, non devi consentire il traffico 24224 in entrata sulla porta del gruppo di sicurezza utilizzato dall'attività. Per attività che utilizzano la modalità di rete `awsvpc`, questo è il gruppo di sicurezza associato all'attività. Per le attività che utilizzano la modalità di `host` rete, questo è il gruppo di sicurezza associato all' EC2 istanza Amazon che ospita l'attività. Per attività che utilizzano la modalità di rete `bridge`, non creare mappature di porte che utilizzano la porta 24224.

- Per le attività che utilizzano la modalità di `bridge` rete, il contenitore con la FireLens configurazione deve avviarsi prima dell'avvio di qualsiasi contenitore di applicazioni che si basa su di essa. Per controllare l'ordine di avvio dei container, utilizza le condizioni di dipendenza nella definizione di attività. Per ulteriori informazioni, consulta [Dipendenze per i container](#).

Note

Se utilizzi i parametri delle condizioni di dipendenza nelle definizioni dei contenitori con una FireLens configurazione, assicurati che ogni contenitore abbia un requisito di `HEALTHY` condizione `START` or.

- Per impostazione predefinita, FireLens aggiunge il nome di definizione del cluster e dell'attività e l'Amazon Resource Name (ARN) del cluster come chiavi di metadati ai log del contenitore `stdout`/ `stderr`. Di seguito è riportato un esempio del formato dei metadati.

```
"ecs_cluster": "cluster-name",  
"ecs_task_arn": "arn:aws:ecs:region:111122223333:task/cluster-  
name/f2ad7dba413f45ddb4EXAMPLE",  
"ecs_task_definition": "task-def-name:revision",
```

Se non desideri i metadati nei tuoi log, imposta `enable-ecs-log-metadata` su `false` nella sezione `firelensConfiguration` della definizione di attività.

```
"firelensConfiguration":{  
  "type":"fluentbit",  
  "options":{  
    "enable-ecs-log-metadata":"false",  
    "config-file-type":"file",  
    "config-file-value":"/extra.conf"  
  }  
}
```

Per utilizzare questa funzionalità, devi creare un ruolo IAM per le tue attività che fornisca le autorizzazioni necessarie per utilizzare tutti i servizi richiesti dalle attività. AWS Ad esempio, se un contenitore sta instradando i log verso Firehose, l'operazione richiede l'autorizzazione per chiamare l'API. `firehose:PutRecordBatch` Per ulteriori informazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente di IAM.

La tua attività potrebbe anche richiedere il ruolo di esecuzione delle attività di Amazon ECS nelle seguenti condizioni. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

- Se la tua attività è ospitata su Fargate e stai estraendo immagini di container da Amazon ECR o stai facendo riferimento AWS Secrets Manager a dati sensibili dalla tua configurazione di registro, devi includere il ruolo IAM di esecuzione dell'attività.
- Quando utilizzi un file di configurazione personalizzato ospitato in Amazon S3, il ruolo IAM per l'esecuzione delle attività deve includere l'`s3:GetObject` autorizzazione.

Per informazioni su come utilizzare più file di configurazione con Amazon ECS, inclusi i file ospitati o i file in Amazon S3, [consulta Processo di avvio per Fluent Bit on ECS](#), supporto multi-config.

Per informazioni sulle configurazioni di esempio, consulta [Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens](#)

Configurazione dei log di Amazon ECS per un throughput elevato

Quando si crea una definizione di attività, è possibile specificare il numero di righe di registro memorizzate nel buffer di memoria specificando il valore in `log-driver-buffer-limit`. Per ulteriori informazioni, consulta la pagina [Driver di registro di Fluentd](#) nella documentazione Docker.

Utilizza questa opzione quando la velocità effettiva è elevata, poiché Docker potrebbe esaurire la memoria buffer e scartare i messaggi del buffer, in modo da poter aggiungere nuovi messaggi.

Considera quanto segue quando utilizzi FireLens Amazon ECS con l'opzione di limite di buffer:

- Questa opzione è supportata nel tipo di EC2 lancio Amazon e nel tipo di lancio Fargate con versione della piattaforma 1.4.0 o successiva.
- L'opzione è valida solo quando `logDriver` è impostato su `awsfirelens`.
- Il limite di buffer predefinito è costituito dalle righe di 1048576 registro.
- Il limite del buffer deve essere maggiore o uguale 0 e minore delle righe di 536870912 registro.
- La quantità massima di memoria utilizzata per questo buffer è il prodotto della dimensione di ogni riga di registro e della dimensione del buffer. Ad esempio, se le righe di registro dell'applicazione sono in media 2 KB, un limite di buffer di 4096 utilizzerebbe al massimo 8 MiB. La quantità totale di memoria allocata a livello di attività deve essere maggiore della quantità di memoria allocata per tutti i contenitori oltre al buffer di memoria del driver di registro.

Quando il driver di log `awsfirelens` è specificato in una definizione di attività, l'agente Amazon ECS inserisce le seguenti variabili di ambiente nel container:

FLUENT_HOST

L'indirizzo IP assegnato al contenitore. FireLens

Note

Se utilizzate il tipo di EC2 avvio con la modalità di bridge rete, la variabile di `FLUENT_HOST` ambiente nel contenitore dell'applicazione può diventare imprecisa dopo il riavvio del contenitore del FireLens log router (il contenitore con `firelensConfiguration` oggetto nella definizione del contenitore). Questo perché `FLUENT_HOST` è un indirizzo IP dinamico e può cambiare dopo un riavvio. La registrazione diretta dal contenitore dell'applicazione all'indirizzo `FLUENT_HOST` IP può iniziare a fallire dopo la modifica dell'indirizzo. Per ulteriori informazioni sul riavvio di singoli contenitori, vedere [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#)

FLUENT_PORT

La porta su cui il protocollo Fluent Forward è in ascolto.

Le variabili di ambiente `FLUENT_HOST` e `FLUENT_PORT` consentono di accedere direttamente al router di log dal codice anziché passare attraverso `stdout`. Per ulteriori informazioni, vedere [fluent-logger-golang](#) on GitHub

Di seguito viene illustrata la sintassi per specificare il `log-driver-buffer-limit`. Sostituisci `my_service_` con il nome del tuo servizio:

```
{
  "containerDefinitions": [
    {
      "name": "my_service_log_router",
      "image": "public.ecr.aws/aws-observability/aws-for-fluent-bit:stable",
      "cpu": 0,
      "memoryReservation": 51,
      "portMappings": [],
      "essential": true,
```



```

    "environment": [],
    "mountPoints": [],
    "volumesFrom": [],
    "user": "0",
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-group": "/ecs/ecs-aws-firelens-sidecar-container",
        "mode": "non-blocking",
        "awslogs-create-group": "true",
        "max-buffer-size": "25m",
        "awslogs-region": "us-east-1",
        "awslogs-stream-prefix": "firelens"
      },
      "secretOptions": []
    },
    "systemControls": [],
    "firelensConfiguration": {
      "type": "fluentbit"
    }
  },
  {
    "essential": true,
    "image": "public.ecr.aws/docker/library/httpd:latest",
    "name": "app",
    "logConfiguration": {
      "logDriver": "awsfirelens",
      "options": {
        "Name": "firehose",
        "region": "us-west-2",
        "delivery_stream": "my-stream",
        "log-driver-buffer-limit": "51200"
      }
    },
    "dependsOn": [
      {
        "containerName": "log_router",
        "condition": "START"
      }
    ],
    "memoryReservation": 100
  }
]

```

```
}
```

AWS per Fluent Bit repository di immagini per Amazon ECS

AWS fornisce un Fluent Bit immagine con plugin sia per CloudWatch Logs che per Firehose. Si consiglia di utilizzare Fluent Bit come router di registro perché ha un tasso di utilizzo delle risorse inferiore a Fluentd. Per ulteriori informazioni, consulta [CloudWatch Logs for Fluent Bit](#) e [Amazon Kinesis Firehose](#) for Fluent Bit.

L'immagine AWS per Fluent Bit è disponibile su Amazon ECR sia nella Amazon ECR Public Gallery che in un repository Amazon ECR nella maggior parte dei casi per l'alta disponibilità. Regioni AWS

Galleria pubblica di Amazon ECR

Il per AWS Fluent Bit l'immagine è disponibile nella Amazon ECR Public Gallery. Questa è la posizione consigliata per scaricare il modulo AWS Fluent Bit immagine perché è un archivio pubblico e può essere utilizzato da tutti Regioni AWS. Per ulteriori informazioni, consulta la [aws-for-fluent-bit](#) Amazon ECR Public Gallery.

Linux

Il per AWS Fluent Bit l'immagine nella Amazon ECR Public Gallery supporta il sistema operativo Amazon Linux con l'x86-64architetturaARM 64, or.

Puoi tirare la forchetta AWS Fluent Bit immagine dalla Amazon ECR Public Gallery specificando l'URL del repository con il tag immagine desiderato. I tag immagine disponibili sono disponibili nella scheda Tag immagine nella galleria pubblica di Amazon ECR.

Di seguito è mostrata la sintassi da utilizzare per la CLI di Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:tag
```

Ad esempio, puoi scaricare l'ultima versione stabile per AWS Fluent Bit immagine usando questo comando CLI Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:stable
```

Note

I pull non autenticati sono consentiti, ma hanno un limite di velocità inferiore rispetto ai pull autenticati. Per autenticarti utilizzando il tuo AWS account prima di estrarlo, usa il seguente comando.

```
aws ecr-public get-login-password --region us-east-1 | docker login --username  
AWS --password-stdin public.ecr.aws
```

Windows

Il per AWS Fluent Bit l'immagine nella Amazon ECR Public Gallery supporta l'AMD64architettura con i seguenti sistemi operativi:

- Windows Server 2022 Full
- Windows Server 2022 Core
- Windows Server 2019 Full
- Windows Server 2019 Core

I contenitori Windows che si trovano su AWS Fargate non supportano. FireLens

Puoi tirare la forchetta AWS Fluent Bit immagine dalla Amazon ECR Public Gallery specificando l'URL del repository con il tag immagine desiderato. I tag immagine disponibili sono disponibili nella scheda Tag immagine nella galleria pubblica di Amazon ECR.

Di seguito è mostrata la sintassi da utilizzare per la CLI di Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:tag
```

Ad esempio, puoi scaricare la stable più recente per AWS Fluent Bit immagine usando questo comando CLI Docker.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:windowsservercore-  
stable
```

Note

I pull non autenticati sono consentiti, ma hanno un limite di velocità inferiore rispetto ai pull autenticati. Per autenticarti utilizzando il tuo AWS account prima di estrarlo, usa il seguente comando.

```
aws ecr-public get-login-password --region us-east-1 | docker login --username  
AWS --password-stdin public.ecr.aws
```

Amazon ECR

L'immagine AWS for Fluent Bit è disponibile su Amazon ECR per un'elevata disponibilità. Queste immagini sono disponibili nella maggior parte dei casi Regioni AWS, tra cui. AWS GovCloud (US)

Linux

L'URI dell'immagine stabile più recente AWS per Fluent Bit può essere recuperata utilizzando il seguente comando.

```
aws ssm get-parameters \  
  --names /aws/service/aws-for-fluent-bit/stable \  
  --region us-east-1
```

Tutte le versioni dell'immagine AWS for Fluent Bit possono essere elencate utilizzando il comando seguente per interrogare il parametro Systems Manager Parameter Store.

```
aws ssm get-parameters-by-path \  
  --path /aws/service/aws-for-fluent-bit \  
  --region us-east-1
```

È possibile fare riferimento all'immagine stabile più recente AWS per Fluent Bit in un AWS CloudFormation modello facendo riferimento al nome dell'archivio dei parametri di Systems Manager. Di seguito è riportato un esempio:

```
Parameters:  
  FireLensImage:  
    Description: Fluent Bit image for the FireLens Container  
    Type: AWS::SSM::Parameter::Value<String>
```

```
Default: /aws/service/aws-for-fluent-bit/stable
```

Windows

L'ultima versione stabile dell'URI AWS di immagine Fluent Bit può essere recuperata utilizzando il seguente comando.

```
aws ssm get-parameters \  
  --names /aws/service/aws-for-fluent-bit/windowsservercore-stable \  
  --region us-east-1
```

Tutte le versioni dell'immagine AWS for Fluent Bit possono essere elencate utilizzando il comando seguente per interrogare il parametro Systems Manager Parameter Store.

```
aws ssm get-parameters-by-path \  
  --path /aws/service/aws-for-fluent-bit/windowsservercore \  
  --region us-east-1
```

È possibile fare riferimento all'ultima immagine stabile AWS per Fluent Bit in un AWS CloudFormation modello facendo riferimento al nome dell'archivio dei parametri di Systems Manager. Di seguito è riportato un esempio:

```
Parameters:  
  FireLensImage:  
    Description: Fluent Bit image for the FireLens Container  
    Type: AWS::SSM::Parameter::Value<String>  
    Default: /aws/service/aws-for-fluent-bit/windowsservercore-stable
```

Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens

Per utilizzare il routing personalizzato dei log con FireLens, è necessario specificare quanto segue nella definizione dell'attività:

- Un contenitore di log router che contiene FireLens configurazione. Si consiglia di contrassegnare il container come `essential`.
- Uno o più container dell'applicazione contenenti una configurazione del log che specifica il driver di log `awsfirelens`.
- Un nome della risorsa Amazon (ARN) del ruolo IAM dell'attività contenente le autorizzazioni richieste dall'attività per instradare i log.

Quando si crea una nuova definizione di attività utilizzando AWS Management Console, è disponibile una sezione di FireLens integrazione che semplifica l'aggiunta di un contenitore di log router. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Amazon ECS converte la configurazione del log e genera la configurazione di output Fluentd o Fluent Bit. La configurazione di output è montata nel container di routing dei log in corrispondenza di `/fluent-bit/etc/fluent-bit.conf` for Fluent Bit e `/fluentd/etc/fluent.conf` for Fluentd.

 Important

FireLens ascolta sulla porta 24224. Pertanto, per garantire che il FireLens log router non sia raggiungibile al di fuori dell'attività, non è necessario consentire il traffico 24224 in ingresso sulla porta del gruppo di sicurezza utilizzato dall'attività. Per attività che utilizzano la modalità di rete `awsvpc`, questo è il gruppo di sicurezza associato all'attività. Per le attività che utilizzano la modalità di rete `host`, questo è il gruppo di sicurezza associato all'EC2 istanza Amazon che ospita l'attività. Per attività che utilizzano la modalità di rete `bridge`, non creare mappature di porte che utilizzano la porta 24224.

Di default, Amazon ECS aggiunge ulteriori campi nelle voci di log che consentono di identificare l'origine dei log.

- `ecs_cluster`: il nome del cluster di cui fa parte il processo.
- `ecs_task_arn`: il nome della risorsa Amazon (ARN) completo dell'attività a cui appartiene il container.
- `ecs_task_definition`: il nome della definizione di attività e la revisione che il processo sta utilizzando.
- `ec2_instance_id`— L'ID dell'EC2 istanza Amazon su cui è ospitato il contenitore. Questo campo è valido solo per le attività che utilizzano il tipo di EC2 avvio.

Puoi `enable-ecs-log-metadata` impostarlo su `false` se non desideri i metadati.

Il seguente esempio di definizione di attività definisce un contenitore di log router che utilizza Fluent Bit per indirizzare i log verso Logs. CloudWatch. Definisce inoltre un contenitore di applicazioni che utilizza una configurazione di log per indirizzare i log verso Amazon Data Firehose e imposta la memoria utilizzata per bufferizzare gli eventi su 2 MiB.

Note

Per ulteriori esempi di definizioni di attività, consulta gli [FireLensesempi di Amazon ECS](#) su GitHub.

```
{
  "family": "firelens-example-firehose",
  "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
  "containerDefinitions": [
    {
      "name": "log_router",
      "image": "public.ecr.aws/aws-observability/aws-for-fluent-bit:stable",
      "cpu": 0,
      "memoryReservation": 51,
      "portMappings": [],
      "essential": true,
      "environment": [],
      "mountPoints": [],
      "volumesFrom": [],
      "user": "0",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/ecs-aws-firelens-sidecar-container",
          "mode": "non-blocking",
          "awslogs-create-group": "true",
          "max-buffer-size": "25m",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "firelens"
        },
        "secretOptions": []
      },
      "systemControls": [],
      "firelensConfiguration": {
        "type": "fluentbit"
      }
    },
    {
      "essential": true,
      "image": "public.ecr.aws/docker/library/httpd:latest",
      "name": "app",
```

```
"logConfiguration": {
  "logDriver": "awsfirelens",
  "options": {
    "Name": "firehose",
    "region": "us-west-2",
    "delivery_stream": "my-stream",
    "log-driver-buffer-limit": "2097152"
  }
},
"memoryReservation": 100
}
```

Le coppie chiave-valore specificate come opzioni nell'`logConfiguration` oggetto vengono utilizzate per generare Fluentd oppure Fluent Bit configurazione di uscita. Di seguito è riportato un esempio di codice tratto da un Fluent Bit definizione dell'output.

[OUTPUT]

```
Name    firehose
Match   app-firelens*
region  us-west-2
delivery_stream my-stream
```

Note

FireLens gestisce la match configurazione. Non si specifica la match configurazione nella definizione dell'attività.

Utilizza un file di configurazione personalizzato

È possibile specificare un file di configurazione personalizzato. Il formato del file di configurazione è il formato nativo per il router di log in uso. [Per ulteriori informazioni, vedere Sintassi del file di configurazione Fluentd e configurazione YAML.](#)

Nel file di configurazione personalizzato, per le attività che utilizzano la modalità di awsvpc rete bridge o, non impostate un input di inoltro Fluentd o Fluent Bit su TCP perché FireLens lo aggiunge alla configurazione di input.

Il tuo FireLens la configurazione deve contenere le seguenti opzioni per specificare un file di configurazione personalizzato:

`config-file-type`

Il percorso di origine del file di configurazione personalizzato. Le opzioni disponibili sono `s3` o `file`.

Note

Le attività ospitate su supportano AWS Fargate solo il tipo `file` di file di configurazione.

`config-file-value`

L'origine del file di configurazione personalizzato. Se viene utilizzato il tipo di file di configurazione `s3`, il valore del file di configurazione è l'ARN completo del file e del bucket Amazon S3. Se viene utilizzato il tipo di file di configurazione `file`, il valore del file di configurazione è il percorso completo del file di configurazione presente nell'immagine del container o in un volume montato nel container.

Important

Quando si utilizza un file di configurazione personalizzato, è necessario specificare un percorso diverso da quello FireLens usi. Amazon ECS riserva il percorso file `/fluent-bit/etc/fluent-bit.conf` for Fluent Bit e `/fluentd/etc/fluent.conf` for Fluentd.

L'esempio seguente mostra la sintassi richiesta quando si specifica una configurazione personalizzata.

Important

Per specificare un file di configurazione personalizzato ospitato in Amazon S3, assicurati di aver creato un ruolo IAM di esecuzione dell'attività con le autorizzazioni appropriate.

Di seguito viene illustrata la sintassi richiesta quando si specifica una configurazione personalizzata.

```
{
  "containerDefinitions": [
    {
      "essential": true,
      "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-bit:stable",
      "name": "log_router",
      "firelensConfiguration": {
        "type": "fluentbit",
        "options": {
          "config-file-type": "s3 | file",
          "config-file-value": "arn:aws:s3:::amzn-s3-demo-bucket/fluent.conf
| filepath"
        }
      }
    }
  ]
}
```

Note

Le attività ospitate su supportano AWS Fargate solo il tipo `file` di file di configurazione.

Utilizzo di immagini non AWS containerizzate in Amazon ECS

Utilizzate il registro privato per memorizzare le vostre credenziali e AWS Secrets Manager quindi fate riferimento ad esse nella definizione dell'attività. In questo modo è possibile fare riferimento alle immagini dei contenitori presenti in registri privati AWS che non richiedono l'autenticazione nelle definizioni delle attività. Questa funzionalità è supportata da attività ospitate su Fargate, istanze Amazon e EC2 istanze esterne che utilizzano Amazon ECS Anywhere.

Important

Le informazioni di questo argomento non sono applicabili se la definizione di attività fa riferimento a un'immagine memorizzata in Amazon ECR. Per ulteriori informazioni, consulta [Utilizzo di immagini Amazon ECR con Amazon ECS](#) nella Guida per l'utente di Amazon Elastic Container Registry.

Per le attività ospitate su EC2 istanze Amazon, questa funzionalità richiede una versione 1.19.0 o successiva dell'agente container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Per i processi ospitati su Fargate, questa funzione richiede la versione 1.2.0 o successiva della piattaforma. Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).

All'interno della definizione del container, specifica l'oggetto `repositoryCredentials` con i dettagli del segreto che hai creato. Il segreto a cui si fa riferimento può provenire da un account diverso Regione AWS o diverso da quello dell'attività che lo utilizza.

Note

Quando utilizzi l'API o l'AWS SDK di Amazon ECS, se il segreto esiste nella Regione AWS stessa attività che stai avviando, puoi utilizzare l'ARN completo o il nome del segreto. AWS CLI Se il segreto esiste in un altro account, occorre specificare l'ARN completo del segreto. Quando si utilizza AWS Management Console, è necessario specificare sempre l'ARN completo del segreto.

Di seguito viene riportato un frammento di una definizione di attività che mostra i parametri obbligatori:

Sostituisci i seguenti parametri:

- *private-repo* con il nome host del repository privato
- *private-image* con il nome dell'immagine
- *arn:aws:secretsmanager:region:aws_account_id:secret:secret_name* con il nome segreto Amazon Resource Name (ARN)

```
"containerDefinitions": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {  
      "credentialsParameter":  
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"  
    }  
  }  
]
```

```
}  
]
```

Note

Un altro metodo per abilitare l'autenticazione di registri privati utilizza le variabili di ambiente dell'agente del container di Amazon ECS per effettuare l'autenticazione a registri privati. Questo metodo è supportato solo per le attività ospitate su EC2 istanze Amazon. Per ulteriori informazioni, consulta [Configurazione delle istanze di container Amazon ECS per immagini Docker private](#).

Per utilizzare il registro privato

1. La definizione dell'attività deve avere un ruolo di esecuzione dell'attività. In questo modo l'agente container può recuperare l'immagine del container. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Le autorizzazioni seguenti devono essere aggiunte manualmente come policy inline al ruolo per l'esecuzione di attività, per fornire l'accesso ai segreti che crei. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: obbligatorio solo se la chiave utilizza una chiave KMS personalizzata e non quella di default. Il nome della risorsa Amazon (ARN) per la chiave personalizzata deve essere aggiunto come risorsa.

Di seguito viene riportata una policy inline che aggiunge le autorizzazioni.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:Decrypt",  
        "secretsmanager:GetSecretValue"  
      ],  
      "Resource": [  

```

```

    "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
    "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
  ]
}
]
}
```

- Viene utilizzato AWS Secrets Manager per creare un segreto per le credenziali del registro privato. Per informazioni su come creare un segreto, consulta [Create an AWS Secrets Manager secret](#) nella Guida per l'AWS Secrets Manager utente.

Immettete le credenziali del registro privato utilizzando il seguente formato:

```

{
  "username" : "privateRegistryUsername",
  "password" : "privateRegistryPassword"
}
```

- Registra una definizione di attività. Per ulteriori informazioni, consulta [the section called "Creazione di una definizione di attività attraverso la nuova console"](#).

Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori

È possibile abilitare una politica di riavvio per ogni contenitore essenziale e non essenziale definito nella definizione dell'attività, per superare più rapidamente gli errori transitori e mantenere la disponibilità delle attività. Quando abiliti una politica di riavvio per un contenitore, Amazon ECS può riavviare il contenitore se esce, senza dover sostituire l'attività.

Le politiche di riavvio non sono abilitate per i contenitori per impostazione predefinita. Quando abiliti una politica di riavvio per un contenitore, puoi specificare i codici di uscita in base ai quali il contenitore non verrà riavviato. Questi possono essere codici di uscita che indicano l'esito positivo, ad esempio codici di uscita 0, che non richiedono un riavvio. È inoltre possibile specificare per quanto tempo un contenitore deve funzionare correttamente prima di tentare il riavvio. Per ulteriori informazioni su questi parametri, consultare [Politica di riavvio](#). Per un esempio di definizione dell'attività che specifica questi valori, vedere [Specificazione di una politica di riavvio dei container in una definizione di attività Amazon ECS](#).

Puoi utilizzare l'endpoint dei metadati delle attività Amazon ECS o CloudWatch Container Insights per monitorare il numero di volte in cui un container è stato riavviato. Per ulteriori informazioni sull'endpoint dei metadati delle attività, consulta e. [Endpoint di metadati delle attività Amazon ECS versione 4](#) [Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate](#) Per ulteriori informazioni sui parametri di Container Insights per Amazon ECS, consulta i parametri di [Amazon ECS Container Insights nella](#) Amazon User Guide. CloudWatch

Le politiche di riavvio dei container sono supportate da attività ospitate su Fargate, istanze Amazon e EC2 istanze esterne che utilizzano Amazon ECS Anywhere.

Considerazioni

Considerate quanto segue prima di abilitare una politica di riavvio per il contenitore:

- I criteri di riavvio non sono supportati per i contenitori Windows su Fargate.
- Per le attività ospitate su EC2 istanze Amazon, questa funzionalità richiede una versione 1.86.0 o successiva dell'agente container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).
- Per i processi ospitati su Fargate, questa funzione richiede la versione 1.4.0 o successiva della piattaforma. Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).
- Se utilizzi il tipo di EC2 avvio con la modalità di bridge rete, la variabile di FLUENT_HOST ambiente nel contenitore dell'applicazione può diventare imprecisa dopo il riavvio del contenitore del FireLens log router (il contenitore con l'`firelensConfiguration` oggetto nella definizione del contenitore). Questo perché FLUENT_HOST è un indirizzo IP dinamico e può cambiare dopo un riavvio. La registrazione diretta dal contenitore dell'applicazione all'indirizzo FLUENT_HOST IP può iniziare a fallire dopo la modifica dell'indirizzo. Per ulteriori informazioni su FLUENT_HOST, consulta [Configurazione dei log di Amazon ECS per un throughput elevato](#).
- L'agente Amazon ECS gestisce le politiche di riavvio dei container. Se per qualche motivo imprevisto l'agente Amazon ECS si guasta o non è più in esecuzione, il contenitore non verrà riavviato.
- Il periodo di tentativo di riavvio definito nella tua politica determina il periodo di tempo (in secondi) per cui il contenitore deve essere eseguito prima che Amazon ECS riavvii un contenitore.

Specificazione di una politica di riavvio dei container in una definizione di attività Amazon ECS

Per specificare una politica di riavvio per un contenitore in una definizione di attività, all'interno della definizione del contenitore, specificare l'`restartPolicy` oggetto. Per ulteriori informazioni sull'`restartPolicy` oggetto, vedere [Politica di riavvio](#).

Di seguito è riportata una definizione di attività che utilizza i contenitori Linux sul tipo di avvio Fargate che configura un server Web. La definizione del contenitore include l'`restartPolicy` oggetto, `enabled` impostata su `true` per abilitare una politica di riavvio per il contenitore. Il contenitore deve funzionare per 180 secondi prima di poter essere riavviato e non verrà riavviato se esce con il codice di uscita, il che indica l'esito positivo. 0

```
{
  "containerDefinitions": [
    {
      "command": [
        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title>
<style>body {margin-top: 40px; background-color: #333;} </style> </head><body>
<div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
<h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-
foreground\""]
      ],
      "entryPoint": ["sh", "-c"],
      "essential": true,
      "image": "public.ecr.aws/docker/library/httpd:2.4",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/fargate-task-definition",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "ecs"
        }
      },
      "name": "sample-fargate-app",
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp"
        }
      ]
    }
  ]
}
```

```
    ],  
    "restartPolicy": {  
      "enabled": true,  
      "ignoredExitCodes": [0],  
      "restartAttemptPeriod": 180  
    }  
  }  
],  
"cpu": "256",  
"executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",  
"family": "fargate-task-definition",  
"memory": "512",  
"networkMode": "awsvpc",  
"runtimePlatform": {  
  "operatingSystemFamily": "LINUX"  
},  
"requiresCompatibilities": ["FARGATE"]  
}
```

Dopo aver registrato una definizione di attività con l'`restartPolicy` oggetto in una definizione di contenitore, è possibile eseguire un'attività o creare un servizio con tale definizione di attività. Per ulteriori informazioni, consultare [Esecuzione di un'applicazione come attività Amazon ECS](#) e [Creazione di un servizio Amazon ECS utilizzando la console](#).

Trasferisci dati sensibili a un contenitore Amazon ECS

Puoi inviare in sicurezza dati sensibili, come le credenziali di un database, al tuo container.

I segreti, come ad esempio chiavi API e credenziali di database, vengono spesso utilizzati dalle applicazioni per accedere ad altri sistemi. In molti casi sono costituiti da un nome utente e una password, un certificato o una chiave API. L'accesso a tali segreti dovrebbe essere limitato a principali IAM specifici che utilizzano IAM e inseriti nei container durante il runtime.

I segreti possono essere inseriti senza problemi nei contenitori da AWS Secrets Manager Amazon EC2 Systems Manager Parameter Store. È possibile fare riferimento a questi segreti nell'attività in qualsiasi dei modi seguenti.

1. Come variabili di ambiente che utilizzano il parametro di definizione di container `secrets`.
2. Come `secretOptions` se la piattaforma per la registrazione di log richiede l'autenticazione. Per ulteriori informazioni, consulta le [opzioni di configurazione della registrazione di log](#).

3. Come segreti estratti da immagini che utilizzano il parametro di definizione di container `repositoryCredentials` se il registro da cui viene estratto il container richiede l'autenticazione. Utilizza questo metodo quando estrai immagini da Amazon ECR Public Gallery. Per ulteriori informazioni, consulta [Autenticazione del registro privato per le attività](#).

Consigliamo di completare la procedura seguente quando configuri la gestione dei segreti.

Usa il AWS Secrets Manager nostro AWS Systems Manager Parameter Store per archiviare materiali segreti

È necessario archiviare in modo sicuro le chiavi API, le credenziali del database e altri materiali segreti in Secrets Manager o come parametro crittografato in Systems Manager Parameter Store. Questi servizi sono simili perché sono entrambi archivi chiave-valore gestiti che vengono utilizzati AWS KMS per crittografare i dati sensibili. Secrets Manager, tuttavia, include anche la possibilità di ruotare automaticamente i segreti, generare segreti casuali e condividere segreti tra gli account. Per utilizzare queste funzionalità, usa Secrets Manager. In caso contrario, utilizzare i parametri crittografati in Systems Manager Parameter Store.

Important

Se il segreto cambia, devi forzare una nuova implementazione o avviare una nuova attività per recuperare l'ultimo valore segreto. Per ulteriori informazioni, consulta i seguenti argomenti:

- Attività: interrompi l'attività, quindi avviala. Per ulteriori informazioni, consultare [Interruzione di un'attività Amazon ECS](#) e [Esecuzione di un'applicazione come attività Amazon ECS](#).
- Servizio: aggiorna il servizio e utilizza l'opzione Force New Deployment. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Recupera dati da un bucket Amazon S3 crittografato

È necessario archiviare i segreti in un bucket Amazon S3 crittografato e utilizzare i ruoli delle attività per limitare l'accesso a tali segreti. In questo modo si evita che i valori delle variabili di ambiente vengano inavvertitamente pubblicati nei log e vengano rivelati durante l'esecuzione. `docker inspect` Quando esegui questa operazione, l'applicazione deve essere scritta in modo tale da leggere il segreto dal bucket Amazon S3. Per ricevere istruzioni, consulta [Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3](#).

Montaggio del segreto in un volume con un container sidecar

Poiché esiste un rischio elevato di perdita di dati a causa delle variabili di ambiente, è consigliabile utilizzare un contenitore secondario che legga i segreti AWS Secrets Manager e li scriva su un volume condiviso. Questo container può essere eseguito e terminato prima del container dell'applicazione utilizzando l'[ordinamento di container Amazon di ECS](#). Quando esegui questa operazione, il container dell'applicazione monta successivamente il volume in cui il segreto è stato scritto. Analogamente al metodo bucket Amazon S3, l'applicazione deve essere scritta per leggere il segreto dal volume condiviso. Poiché il suo ambito è limitato all'attività, il volume viene eliminato in automatico dopo l'interruzione dell'attività. [Per un esempio, consulta il progetto task-def.json.](#)

Su Amazon EC2, il volume su cui è scritto il segreto può essere crittografato con una chiave gestita AWS KMS dal cliente. SÌ AWS Fargate, lo storage di volumi viene crittografato automaticamente utilizzando una chiave gestita dal servizio.

Passa una singola variabile di ambiente a un contenitore Amazon ECS

Important

Ti consigliamo di archiviare i dati sensibili nei parametri AWS Secrets Manager segreti o AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

Le variabili di ambiente specificate nella definizione di attività sono leggibili da tutti gli utenti IAM e i ruoli per i quali è consentita l'operazione `DescribeTaskDefinition` per la definizione di attività.

Puoi passare le variabili di ambiente ai container nei seguenti modi:

- Individualmente utilizzando il parametro di definizione del container `environment`. Questo viene mappato all'opzione `--env` su [docker container run](#).
- In blocco, utilizzando il parametro di definizione del container `environmentFiles` per elencare uno o più file contenenti le variabili di ambiente. Il file deve essere ospitato in Amazon S3. Questo viene mappato all'opzione `--env-file` su [docker run](#).

Di seguito è riportato uno snippet di una definizione di attività che mostra come specificare singole variabili di ambiente.

```
{
  "family": "",
  "containerDefinitions": [
    {
      "name": "",
      "image": "",
      ...
      "environment": [
        {
          "name": "variable",
          "value": "value"
        }
      ],
      ...
    }
  ],
  ...
}
```

Passa le variabili di ambiente a un contenitore Amazon ECS

Important

Ti consigliamo di archiviare i dati sensibili nei parametri AWS Secrets Manager segreti o AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

I file variabili di ambiente sono oggetti in Amazon S3 e si applicano tutte le considerazioni sulla sicurezza di Amazon S3.

Non è possibile utilizzare il `environmentFiles` parametro sui contenitori Windows e sui contenitori Windows su Fargate.

Puoi creare un file di variabili di ambiente e archivarlo in Amazon S3 per passare le variabili di ambiente al tuo contenitore.

Specificando le variabili di ambiente in un file, puoi inserire in blocco le variabili di ambiente. All'interno della definizione del container, specifica l'oggetto `environmentFiles` con un elenco di bucket Amazon S3 contenenti i file delle variabili di ambiente.

Amazon ECS non impone un limite di dimensione per le variabili di ambiente, ma un file di variabili di ambiente di grandi dimensioni potrebbe occupare lo spazio su disco. Ogni processo che utilizza un file di variabili di ambiente fa sì che una copia del file venga scaricata sul disco. Amazon ECS rimuove il file come parte della pulizia delle attività.

Per informazioni sulle variabili di ambiente supportate, consulta [Parametri avanzati di definizione del container - Ambiente](#).

Quando si specifica un file di variabili di ambiente in una definizione di container, considera quanto segue.

- Per le attività di Amazon ECS su Amazon EC2, le istanze di container richiedono che l'agente container sia una versione 1.39.0 o successiva per utilizzare questa funzionalità. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).
- Per le attività di Amazon ECS su AWS Fargate, le attività devono utilizzare la 1.4.0 versione della piattaforma o successiva (Linux) per utilizzare questa funzionalità. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Verifica che la variabile sia supportata per la piattaforma del sistema operativo. Per ulteriori informazioni, consultare [the section called “Definizioni del container”](#) e [the section called “Altri parametri di definizione di attività”](#).

- Il file deve utilizzare l'estensione del file `.env` e la codifica UTF-8.
- Il ruolo di esecuzione delle attività è necessario per utilizzare questa funzionalità con le autorizzazioni aggiuntive per Amazon S3. Ciò consente all'agente del container di estrarre il file della variabile di ambiente da Amazon S3. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
- È previsto un limite di 10 file per definizione di attività.
- Ogni riga di un file di ambiente deve contenere una variabile di ambiente nel formato `VARIABLE=VALUE`. Spazi o virgolette sono inclusi come parte dei valori per i file Amazon ECS. Le righe che iniziano con `#` vengono trattate come commenti e vengono ignorate. Per ulteriori informazioni sulla sintassi del file delle variabili di ambiente, vedete [Impostare le variabili di ambiente \(-e, --env, --env-file\)](#) nella Docker documentazione.

Di seguito è riportata la sintassi appropriata.

```
#This is a comment and will be ignored
```

```
VARIABLE=VALUE
ENVIRONMENT=PRODUCTION
```

- Se sono specificate variabili di ambiente utilizzando il parametro `environment` in una definizione di container, queste hanno la precedenza sulle variabili contenute in un file di ambiente.
- Se vengono specificati più file di ambiente che contengono la stessa variabile, vengono elaborati in ordine di inserimento. Ciò significa che viene utilizzato il primo valore della variabile e i valori successivi delle variabili duplicate vengono ignorati. Consigliamo di utilizzare nomi di variabili univoci.
- Se un file di ambiente viene specificato come override di container, viene utilizzato. Inoltre, tutti gli altri file di ambiente specificati nella definizione del container vengono ignorati.
- Le seguenti regole si applicano al tipo di avvio Fargate:
 - Il file viene gestito in modo simile a un file `env` Docker nativo.
 - Le definizioni dei container che fanno riferimento a variabili di ambiente vuote e archiviate in Amazon S3 non vengono visualizzate nel contenitore.
 - Non è disponibile alcun supporto per la gestione dell'escape della shell (interprete di comandi).
 - Il punto di ingresso del container interpreta i valori `VARIABLE`.

Esempio

Di seguito è riportato uno snippet di una definizione di attività che mostra come specificare un file di variabile di ambiente.

```
{
  "family": "",
  "containerDefinitions": [
    {
      "name": "",
      "image": "",
      ...
      "environmentFiles": [
        {
          "value": "arn:aws:s3:::amzn-s3-demo-  
bucket/envfile_object_name.env",
          "type": "s3"
        }
      ],
      ...
    }
  ]
}
```

```
],  
...  
}
```

Trasmetti i segreti di Secrets Manager in modo programmatico in Amazon ECS

Invece di codificare le informazioni sensibili in testo normale nell'applicazione, è possibile utilizzare Secrets Manager per archiviare i dati sensibili.

Consigliamo questo metodo di recupero dei dati sensibili perché, se il segreto di Gestione dei segreti verrà aggiornato in futuro, l'applicazione recupererà automaticamente l'ultima versione del segreto.

Creazione di un segreto in Secrets Manager. Dopo aver creato un segreto di Gestione dei segreti, aggiorna il codice dell'applicazione per recuperarlo.

Prima di proteggere i dati sensibili in Gestione dei segreti, considera i seguenti aspetti.

- Sono supportati solo i segreti che memorizzano dati di testo, ovvero segreti creati con il `SecretString` parametro dell'[CreateSecret](#) API. I segreti che memorizzano dati binari, ovvero segreti creati con il `SecretBinary` parametro dell'[CreateSecret](#) API, non sono supportati.
- Utilizza gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza. È necessario creare gli endpoint VPC di interfaccia per Gestione dei segreti. Per informazioni sull'endpoint VPC, consulta [Creazione di endpoint VPC](#) nella Guida per l'utente di AWS Secrets Manager .
- Il VPC utilizzato dalla tua attività deve utilizzare la risoluzione DNS.
- La definizione dell'attività deve utilizzare un ruolo di attività con autorizzazioni aggiuntive per Secrets Manager. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Creazione del segreto di Gestione dei segreti

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni sulla creazione dei segreti, consulta [Creazione di un segreto AWS Secrets Manager](#) nella Guida per l'utente di AWS Secrets Manager .

Aggiornamento dell'applicazione per il recupero programmatico dei segreti di Gestione dei segreti

È possibile recuperare i segreti con una chiamata al Secrets Manager APIs direttamente dall'applicazione. Per informazioni, consulta [Recupera segreti dalla Guida AWS Secrets Manager](#) per l'AWS Secrets Manager utente.

Per recuperare i dati sensibili memorizzati in AWS Secrets Manager, consulta [Esempi di codice da AWS Secrets Manager utilizzare AWS SDKs nella libreria di](#) codici AWS SDK Code Examples.

Trasmetti i segreti di Systems Manager Parameter Store in modo programmatico in Amazon ECS

Archivio dei parametri Systems Manager fornisce l'archiviazione e la gestione sicure dei segreti. È possibile archiviare dati come password, stringhe di database, EC2 istanze e IDs AMI IDs e codici di licenza come valori dei parametri, anziché codificare queste informazioni nell'applicazione. È possibile memorizzare i valori in testo semplice o crittografati.

Consigliamo questo metodo di recupero dei dati sensibili perché se il parametro Systems Manager Parameter Store viene successivamente aggiornato, l'applicazione recupera automaticamente la versione più recente.

Prima di proteggere i dati sensibili in Archivio dei parametri Systems Manager, considera i seguenti aspetti.

- Sono supportati solo i segreti che archiviano dati di testo. I segreti che memorizzano dati binari non sono supportati.
- Utilizza gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza.
- Il VPC utilizzato dalla tua attività deve utilizzare la risoluzione DNS.
- Per le attività che utilizzano il tipo di EC2 avvio, è necessario utilizzare la variabile di configurazione dell'agente Amazon ECS `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` per utilizzare questa funzionalità. Puoi aggiungerlo al file `/etc/ecs/ecs.config` durante la creazione dell'istanza di container oppure aggiungerlo a un'istanza esistente e quindi riavviare l'agente ECS. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).
- La definizione dell'attività deve utilizzare un ruolo di attività con autorizzazioni aggiuntive per Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Creazione del parametro di

Puoi utilizzare la console Systems Manager per creare un parametro di Archivio dei parametri Systems Manager per i dati sensibili. Per ulteriori informazioni, consulta [Creazione di un parametro Systems Manager \(console\)](#) o [Creare un parametro Systems Manager \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager .

Aggiornamento automatico dell'applicazione per recuperare in modo programmatico i segreti di Archivio dei parametri Systems Manager

Per recuperare i dati sensibili memorizzati nel parametro Systems Manager Parameter Store, consulta [Esempi di codice per l'utilizzo di Systems Manager AWS SDKs](#) nella libreria di codici AWS SDK Code Examples.

Trasmetti i segreti di Secrets Manager tramite le variabili di ambiente Amazon ECS

Quando si inserisce un segreto come variabile di ambiente, è possibile specificare l'intero contenuto di un segreto, una chiave JSON specifica all'interno di un segreto. Questo processo consente di controllare i dati sensibili esposti al container. Per ulteriori informazioni sul controllo delle versioni segrete, vedi [Cosa c'è in un segreto di Secrets Manager?](#) nella Guida per l'AWS Secrets Manager utente.

Quando si utilizza una variabile di ambiente per iniettare un segreto di Secrets Manager in un contenitore, è necessario considerare quanto segue.

- I dati sensibili vengono inseriti nel container quando viene inizialmente avviato. Se il segreto o parametro viene in seguito aggiornato o ruotato, il container non riceverà automaticamente il valore aggiornato. È necessario avviare una nuova attività o se l'attività è parte di un servizio, è possibile aggiornare il servizio e utilizzare la nuova opzione Force new deployment (Forza nuova implementazione) per forzare il servizio ad avviare una nuova attività.
- Le applicazioni eseguite sul contenitore, i registri del contenitore e gli strumenti di debug hanno accesso alle variabili di ambiente.
- Per le attività di Amazon ECS su AWS Fargate, considera quanto segue:
 - Per inserire il contenuto completo di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma 11.3.0 o successiva. Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).
 - Per inserire una chiave JSON specifica o la versione di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma 1.4.0 o successiva (Linux) o 1.0.0 (Windows). Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).
- Per le attività di Amazon ECS su EC2, è necessario considerare quanto segue:
 - Per inserire un segreto utilizzando una chiave JSON specifica o una versione di un segreto, l'istanza di container deve avere la versione 1.37.0 o successiva dell'agente del container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per

informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Per inserire il contenuto completo di un segreto come variabile di ambiente o per inserire un segreto in una configurazione di log, l'istanza del container deve avere la versione 1.22.0 o successiva dell'agente del container.

- Usa gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza e connessi a Secrets Manager tramite una sottorete privata. È necessario creare gli endpoint VPC di interfaccia per Gestione dei segreti. Per informazioni sull'endpoint VPC, consulta [Creazione di endpoint VPC](#) nella Guida per l'utente di AWS Secrets Manager. Per ulteriori informazioni sull'utilizzo di Secrets Manager e Amazon VPC, consulta [Come connettersi al servizio Secrets Manager all'interno di un Amazon VPC](#).
- Per le attività di Windows configurate per utilizzare il driver di registrazione awslogs, è necessario impostare anche la variabile di ambiente ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE nell'istanza di container. Utilizzare la seguente sintassi:

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE",
$TRUE, "Machine")
Initialize-ECSAgent -Cluster <cluster name> -EnableTaskIAMRole -LoggingDrivers
'["json-file","awslogs"]'
</powershell>
```

- La definizione dell'attività deve utilizzare un ruolo di esecuzione dell'attività con autorizzazioni aggiuntive per Secrets Manager. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Creazione del segreto di AWS Secrets Manager

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni, consulta [Creare un AWS Secrets Manager segreto](#) nella Guida per l'AWS Secrets Manager utente.

Aggiunta della variabile di ambiente alla definizione di container

All'interno della definizione del container, è possibile specificare quanto segue:

- L'oggetto `secrets` contenente il nome della variabile di ambiente da impostare nel container
- L'Amazon Resource Name (ARN) del segreto di Secrets Manager

- Parametri aggiuntivi che contengono i dati sensibili da presentare al container

Nell'esempio seguente viene illustrata la sintassi completa che deve essere specificata per il segreto di Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

Nella sezione seguente vengono descritti i parametri aggiuntivi. Questi parametri sono facoltativi, ma se non li si utilizza, è necessario includere i due punti : per utilizzare i valori predefiniti. Esempi sono forniti di seguito per un maggiore contesto.

json-key

Specificare il nome della chiave in una coppia chiave-valore con il valore che si desidera impostare come valore della variabile di ambiente. Sono supportati solo i valori in formato JSON. Se non si specifica una chiave JSON, viene utilizzato il contenuto completo del segreto.

version-stage

Specificare l'etichetta di gestione temporanea della versione di un segreto che si desidera utilizzare. Se viene specificata un'etichetta di gestione temporanea della versione, non è possibile specificare un ID versione. Se non viene specificata alcuna fase di versione, il comportamento predefinito consiste nel recuperare il segreto con l'etichetta AWSCURRENT di gestione temporanea.

Le etichette di gestione temporanea vengono utilizzate per tenere traccia di diverse versioni di un segreto quando vengono aggiornate o ruotate. Ogni versione di un segreto ha una o più etichette di gestione temporanea e un ID. Per ulteriori informazioni, consulta [Termini e concetti chiave di AWS Secrets Manager](#) nella Guida per l'utente di AWS Secrets Manager .

version-id

Specifica l'identificatore univoco della versione del segreto che intendi utilizzare. Se viene specificato un ID versione, non è possibile specificare un'etichetta di gestione temporanea della versione. Se non viene specificato alcun ID versione, il comportamento predefinito consiste nel recuperare il segreto con l'etichetta AWSCURRENT di gestione temporanea.

IDs Le versioni vengono utilizzate per tenere traccia delle diverse versioni di un segreto quando vengono aggiornate o ruotate. Ogni versione di un segreto ha un ID. Per ulteriori informazioni, consulta [Termini e concetti chiave di AWS Secrets Manager](#) nella Guida per l'utente di AWS Secrets Manager .

Esempio di definizioni del container

Negli esempi seguenti vengono illustrati i modi in cui è possibile fare riferimento ai segreti di Secrets Manager nelle definizioni del container.

Example riferimento a un segreto completo

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un segreto di Secrets Manager.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
    }]
  }]
}
```

Per accedere al valore di questo segreto dall'interno del container, è necessario effettuare la chiamata a `$environment_variable_name`.

Example fare riferimento a una chiave specifica all'interno di un segreto

Di seguito viene illustrato un esempio di output di un [get-secret-value](#) comando che visualizza il contenuto di un segreto insieme all'etichetta temporanea della versione e all'ID di versione ad esso associati.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "VersionId": "871d9eca-18aa-46a9-8785-981ddEXAMPLE",
  "SecretString": "{\"username1\":\"password1\",\"username2\":\"password2\", \"username3\":\"password3\"}",
  "VersionStages": [
    "AWSCURRENT"
  ],
  "CreateDate": 1581968848.921
}
```

Fare riferimento a una chiave specifica dell'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf:username1::"
    }]
  }]
}
```

Example riferimento a una versione segreta specifica

Di seguito viene illustrato un output di esempio da un comando [describe-secret](#) che visualizza il contenuto non crittografato di un segreto insieme ai metadati per tutte le versioni del segreto.

```
{
  "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
  "Name": "appauthexample",
  "Description": "Example of a secret containing application authorization data.",
  "RotationEnabled": false,
  "LastChangedDate": 1581968848.926,
  "LastAccessedDate": 1581897600.0,
  "Tags": [],
  "VersionIdsToStages": {
    "871d9eca-18aa-46a9-8785-981ddEXAMPLE": [
      "AWSCURRENT"
    ],
    "9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE": [
      "AWSPREVIOUS"
    ]
  }
}
```

Fare riferimento a un'etichetta di gestione temporanea della versione specifica dall'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```
{
  "containerDefinitions": [{
    "secrets": [{
```

```

    "name": "environment_variable_name",
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::AWSPREVIOUS:"
  }}
}}
}

```

Fare riferimento a un ID di versione specifico dall'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```

{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE"
    }]
  }]
}

```

Example riferimento a una chiave specifica e un'etichetta di gestione temporanea della versione di un segreto

Di seguito viene illustrato come fare riferimento sia a una chiave specifica all'interno di un segreto che a una specifica etichetta di gestione temporanea della versione.

```

{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}

```

Per specificare una chiave e un ID di versione specifici, utilizzare la seguente sintassi.

```

{
  "containerDefinitions": [{
    "secrets": [{

```

```
"name": "environment_variable_name",
"valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE"
}]
}]
}
```

Per informazioni su come creare una definizione di attività con il segreto specificato in una variabile di ambiente, vedere [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Passa i parametri di Systems Manager tramite le variabili di ambiente Amazon ECS

Amazon ECS ti consente di iniettare dati sensibili nei tuoi contenitori memorizzando i dati sensibili nei AWS Systems Manager parametri di Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore.

Considerate quanto segue quando utilizzate una variabile di ambiente per iniettare un segreto di Systems Manager in un contenitore.

- I dati sensibili vengono inseriti nel container quando viene inizialmente avviato. Se il segreto o parametro viene in seguito aggiornato o ruotato, il container non riceverà automaticamente il valore aggiornato. È necessario avviare una nuova attività o se l'attività è parte di un servizio, è possibile aggiornare il servizio e utilizzare la nuova opzione Force new deployment (Forza nuova implementazione) per forzare il servizio ad avviare una nuova attività.
- Per le attività di Amazon ECS su AWS Fargate, è necessario considerare quanto segue:
 - Per inserire il contenuto completo di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma 11.3.0 o successiva. Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).
 - Per inserire una chiave JSON specifica o la versione di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma 1.4.0 o successiva (Linux) o 1.0.0 (Windows). Per informazioni, consultare [Versioni della piattaforma Fargate per Amazon ECS](#).
- Per le attività di Amazon ECS su EC2, è necessario considerare quanto segue:
 - Per inserire un segreto utilizzando una chiave JSON specifica o una versione di un segreto, l'istanza di container deve avere la versione 1.37.0 o successiva dell'agente del container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Per inserire il contenuto completo di un segreto come variabile di ambiente o per inserire un segreto in una configurazione di log, l'istanza del container deve avere la versione 1.22.0 o successiva dell'agente del container.

- Utilizza gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza. È necessario creare gli endpoint VPC di interfaccia per Systems Manager. Per informazioni sull'endpoint VPC, consulta [Migliorare la sicurezza delle EC2 istanze utilizzando gli endpoint VPC per Systems Manager nella Guida per l'utente](#).AWS Systems Manager
- La definizione dell'attività deve utilizzare un ruolo di esecuzione dell'attività con autorizzazioni aggiuntive per Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
- Per le attività di Windows configurate per utilizzare il driver di registrazione awslogs, è necessario impostare anche la variabile di ambiente ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE nell'istanza di container. Utilizzare la seguente sintassi:

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE",
    $TRUE, "Machine")
Initialize-ECSAgent -Cluster <cluster name> -EnableTaskIAMRole -LoggingDrivers
    ["json-file","awslogs"]'
</powershell>
```

Creare il parametro Systems Manager

Puoi utilizzare la console Systems Manager per creare un parametro di Archivio dei parametri Systems Manager per i dati sensibili. Per ulteriori informazioni, consulta [Creazione di un parametro Systems Manager \(console\)](#) o [Creare un parametro Systems Manager \(AWS CLI\)](#) nella Guida per l'utente di AWS Systems Manager .

Aggiunta della variabile di ambiente alla definizione di container

All'interno della definizione del contenitore nella definizione dell'attività, specificare `secrets` il nome della variabile di ambiente da impostare nel contenitore e l'ARN completo del parametro Systems Manager Parameter Store contenente i dati sensibili da presentare al contenitore. Per ulteriori informazioni, consulta [secrets](#).

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un parametro dell'archivio parametri di Systems Manager. Se il parametro dell'archivio

parametri di Systems Manager esiste nella stessa regione del processo in fase di avvio, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro si trova in una regione diversa, specifica l'ARN completo.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Per informazioni su come creare una definizione di attività con il segreto specificato in una variabile di ambiente, vedere [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Aggiornamento automatico dell'applicazione per recuperare in modo programmatico i segreti di Archivio dei parametri Systems Manager

Per recuperare i dati sensibili memorizzati nel parametro Systems Manager Parameter Store, consulta [Esempi di codice per l'utilizzo di Systems Manager AWS SDKs](#) nella libreria di codici AWS SDK Code Examples.

Pass secret per la configurazione dei log di Amazon ECS

È possibile utilizzare il `secretOptions` parametro in `logConfiguration` per trasmettere dati sensibili utilizzati per la registrazione.

È possibile memorizzare il segreto in Secrets Manager o Systems Manager.

Usa Secrets Manager

Nella definizione del container, quando specifichi `logConfiguration` è possibile specificare `secretOptions` con il nome dell'opzione del driver di log per impostare il container e l'ARN completo del segreto di Secrets Manager contenente i dati riservati da presentare al container.

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un segreto Secrets Manager.

```
{
  "containerDefinitions": [{
```



```
"logConfiguration": [{
  "logDriver": "splunk",
  "options": {
    "splunk-url": "https://your_splunk_instance:8088"
  },
  "secretOptions": [{
    "name": "splunk-token",
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
  }]
}]
}]
}
```

Aggiunta della variabile di ambiente alla definizione di container

Nella definizione del container specifica `secrets` con il nome della variabile di ambiente per impostare il container e l'ARN del parametro dell'archivio parametri di Systems Manager contenente i dati sensibili da presentare al container. Per ulteriori informazioni, consulta [secrets](#).

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un parametro dell'archivio parametri di Systems Manager. Se il parametro dell'archivio parametri di Systems Manager esiste nella stessa regione del processo in fase di avvio, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro si trova in una regione diversa, specifica l'ARN completo.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Per informazioni su come creare una definizione di attività con il segreto specificato in una variabile di ambiente, vedere [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Utilizzo di Systems Manager

Puoi inserire dati sensibili in una configurazione di log. Nella definizione del container, quando specifichi un `logConfiguration` è possibile specificare `secretOptions` con il nome dell'opzione

del driver di log per impostare il container e l'ARN completo del parametro dell'archivio parametri di Systems Manager contenente i dati sensibili da presentare al container.

Important

Se il parametro dell'archivio parametri di Systems Manager esiste nella stessa regione del processo in fase di avvio, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro si trova in una regione diversa, specifica l'ARN completo.

Di seguito è riportato un frammento di una definizione di attività che mostra il formato quando si fa riferimento a un parametro dell'archivio parametri di Systems Manager.

```
{
  "containerDefinitions": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter:/parameter_name"
      }]
    }]
  }]
}
```

Specificazione di dati sensibili utilizzando i segreti di Secrets Manager in Amazon ECS

Amazon ECS ti consente di iniettare dati sensibili nei tuoi contenitori archiviando i dati sensibili in modo AWS Secrets Manager segreto e quindi facendone riferimento nella definizione del contenitore. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

Scopri come creare un segreto di Secrets Manager, fare riferimento al segreto in una definizione di attività Amazon ECS e quindi verificarne il funzionamento interrogando la variabile di ambiente all'interno di un contenitore che mostra il contenuto del segreto.

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente dispone delle autorizzazioni IAM necessarie per creare le risorse Secrets Manager e Amazon ECS.

Fase 1: Creazione di un segreto di Secrets Manager

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. In questo tutorial creeremo un segreto di base per l'archiviazione di un nome utente e una password cui fare riferimento in seguito in un container. Per ulteriori informazioni, consulta [Creare un AWS Secrets Manager segreto nella Guida](#) per l'AWS Secrets Manager utente.

Le coppie chiave/valore da memorizzare in questo segreto rappresentano il valore della variabile di ambiente nel container alla fine del tutorial.

Salva l'ARN del segreto a cui fare riferimento nella policy IAM di esecuzione dell'attività e nella definizione dell'attività nelle fasi successive.

Passaggio 2: aggiungere le autorizzazioni segrete al ruolo di esecuzione dell'attività

Affinché Amazon ECS possa recuperare i dati sensibili dal tuo segreto di Secrets Manager, devi disporre delle autorizzazioni segrete per il ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta [Autorizzazioni Secrets Manager o Systems Manager](#).

Fase 3: creazione di una definizione di attività

Puoi utilizzare la console Amazon ECS per creare una definizione di attività che fa riferimento a un segreto di Secrets Manager.

Per creare una definizione di attività che specifichi un segreto

Utilizza la console IAM per aggiornare il ruolo di esecuzione del processo con le autorizzazioni richieste.

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).
4. Nella casella dell'editor JSON, inserisci il seguente testo JSON di definizione dell'attività, assicurandoti di specificare l'ARN completo del segreto di Secrets Manager che hai creato nel

passaggio 1 e il ruolo di esecuzione dell'attività che hai aggiornato nel passaggio 2. Scegli Save (Salva).

5.

```
{
  "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "entryPoint": [
        "sh",
        "-c"
      ],
      "portMappings": [
        {
          "hostPort": 80,
          "protocol": "tcp",
          "containerPort": 80
        }
      ],
      "command": [
        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-foreground\""
      ],
      "cpu": 10,
      "secrets": [
        {
          "valueFrom":
            "arn:aws:secretsmanager:region:aws_account_id:secret:username_value",
          "name": "username_value"
        }
      ],
      "memory": 300,
      "image": "public.ecr.aws/docker/library/httpd:2.4",
      "essential": true,
      "name": "ecs-secrets-container"
    }
  ],
  "family": "ecs-secrets-tutorial"
}
```

6. Scegli Create (Crea).

Fase 4: Creare un cluster

Puoi utilizzare la console Amazon ECS per creare un cluster che contiene un'istanza di container su cui eseguire il processo. Se disponi di un cluster esistente con almeno un'istanza di container registrata nello stesso con le risorse disponibili per eseguire un'istanza della definizione di attività creata per questo tutorial, puoi passare alla fase successiva.

Per questo tutorial creeremo un cluster con una istanza di container `t2.micro` utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS.

Per informazioni su come creare un cluster per il tipo di EC2 avvio, consulta [the section called “Creazione di un cluster per il tipo di EC2 lancio Amazon”](#).

Fase 5: Eseguire un'attività

Puoi utilizzare la console Amazon ECS per eseguire un processo utilizzando la definizione di attività creata. In questo tutorial eseguiremo un'attività utilizzando il tipo di EC2 avvio, utilizzando il cluster creato nel passaggio precedente.

Per informazioni sulla modalità di esecuzione del comando, consulta [the section called “Esecuzione di un'applicazione come attività”](#).

Fase 6: verifica

Puoi verificare che tutte le fasi sono state completate correttamente e che la variabile di ambiente è stata creata nel container utilizzando le fasi seguenti.

Per verificare che la variabile di ambiente è stata creata

1. Trova l'indirizzo DNS o IP pubblico per l'istanza di container.
 - a. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
 - b. Nel riquadro di navigazione, seleziona Cluster e quindi il cluster che hai creato.
 - c. Seleziona Infrastruttura e quindi l'istanza di container.
 - d. Registra l'IP pubblico o il DNS pubblico per l'istanza.
2. Se utilizzi un computer macOS o Linux, connettiti all'istanza con il comando seguente, sostituisci il percorso con la chiave privata e l'indirizzo pubblico dell'istanza:

```
$ ssh -i /path/to/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
```

Per ulteriori informazioni sull'uso di un computer Windows, consulta [Connect alla tua istanza Linux usando PuTTY](#) nella Amazon EC2 User Guide.

 Important

Per ulteriori informazioni su eventuali problemi di connessione alla tua istanza, consulta [Risoluzione dei problemi di connessione alla tua istanza](#) nella Amazon EC2 User Guide.

3. Elencare i contenitori in esecuzione sull'istanza. Prendere nota dell'ID container per il container `ecs-secrets-tutorial`.

```
docker ps
```

4. Eseguire il collegamento al container `ecs-secrets-tutorial` utilizzando l'ID container dell'output della fase precedente.

```
docker exec -it container_ID /bin/bash
```

5. Utilizzare il comando `echo` per stampare il valore della variabile di ambiente.

```
echo $username_value
```

Se il tutorial è stato completato, viene visualizzato l'output seguente:

```
password_value
```

 Note

In alternativa, è possibile elencare tutte le variabili di ambiente nel container utilizzando il comando `env` (o `printenv`).

Fase 7: pulire

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Come ripulire le risorse

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Scegli Elimina Cluster.
5. Nella casella di conferma, inserisci delete **cluster name**, quindi scegli Elimina.
6. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
7. Nel riquadro di navigazione, seleziona Ruoli.
8. Cercare l'elenco di ruoli per ecsTaskExecutionRole e selezionarlo.
9. Scegli Autorizzazioni, quindi scegli la X accanto a ECSSecretsTutorial. Scegli Rimuovi.
10. Apri la console Secrets Manager all'indirizzo <https://console.aws.amazon.com/secretsmanager/>.
11. Selezionare il segreto username_value creato e scegliere Actions (Operazioni), Delete secret (Elimina segreto).

Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate

Le definizioni delle attività sono suddivise in parti separate: la famiglia di attività, il ruolo dell'attività AWS Identity and Access Management (IAM), la modalità di rete, le definizioni dei contenitori, i volumi e i tipi di avvio. Le definizioni della famiglia e del container sono richieste in una definizione di attività. Al contrario, il ruolo dell'attività, la modalità di rete, i volumi e il tipo di avvio sono facoltativi.

Puoi utilizzare questi parametri in un file JSON per configurare la definizione del processo.

Di seguito sono riportate descrizioni più dettagliate per ogni parametro di definizione delle attività per il tipo di avvio Fargate.

Family

family

Tipo: stringa

Campo obbligatorio: sì

Quando registri una definizione di attività, le assegna una famiglia, ovvero una sorta di nome per più versioni della definizione di attività, specificando un numero di revisione. Alla prima definizione di attività registrata in una determinata famiglia viene assegnato il numero di revisione 1 e a qualsiasi definizione di attività registrata successivamente viene assegnato un numero di revisione sequenziale.

Tipi di avvio

Quando registri una definizione di attività, puoi specificare un tipo di avvio che Amazon ECS deve convalidare per la definizione di attività. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client. Per ulteriori informazioni, consulta [Tipi di avvio di Amazon ECS](#).

Il parametro seguente è permesso in una definizione di attività.

`requiresCompatibilities`

Tipo: array di stringhe

Campo obbligatorio: no

Valori validi: FARGATE

Il tipo di avvio per il quale è stata convalidata la definizione di attività. In questo modo viene avviato un controllo per garantire che tutti i parametri utilizzati nella definizione di attività soddisfino i requisiti del tipo di avvio.

Ruolo del processo

`taskRoleArn`

Tipo: string

Campo obbligatorio: no

Quando registrate una definizione di task, potete fornire un ruolo di task per un ruolo IAM che consenta ai contenitori inclusi nell'operazione di richiamare per vostro conto i AWS APIs dati specificati nelle policy associate. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Ruolo per l'esecuzione del processo

`executionRoleArn`

Tipo: stringa

Obbligatorio: condizionale

L'Amazon Resource Name (ARN) del ruolo di esecuzione dell'attività che concede all'agente container Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto.

Note

Il ruolo IAM di esecuzione del processo è richiesto in base ai requisiti del processo. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Modalità di rete

`networkMode`

Tipo: stringa

Campo obbligatorio: sì

La modalità di rete Docker da utilizzare per i container nel processo. Per le attività di Amazon ECS ospitate su Fargate, è richiesta `awsvpc` la modalità di rete.

Se la modalità di rete è impostata su `none`, i container dell'attività non dispongono di connettività esterna e non è possibile specificare le mappature delle porte nella definizione del container.

Quando la modalità di rete è `awsvpc`, al processo viene assegnata un'interfaccia di rete elastica e, quando crei un servizio o esegui un processo con la definizione di attività, devi specificare `NetworkConfiguration`. Per ulteriori informazioni, consulta [Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#). La modalità `awsvpc` di rete offre le massime prestazioni di rete per i container perché utilizzano lo stack EC2 di rete Amazon. Le porte dei container esposte vengono mappate direttamente alla porta dell'interfaccia di rete elastica collegata. Pertanto, non puoi utilizzare le mappature delle porte host dinamiche.

La modalità `awsvpc` di rete offre le massime prestazioni di rete per i container perché utilizzano lo stack EC2 di rete Amazon. Con le modalità di `awsvpc` rete, le porte container esposte vengono mappate direttamente sulla porta elastic network interface collegata.

Pertanto, non puoi utilizzare le mappature delle porte host dinamiche.

È richiesta la modalità di `awsvpc` rete.

Piattaforma di runtime

`operatingSystemFamily`

Tipo: stringa

Obbligatorio: condizionale

Di default: LINUX

Questo parametro è richiesto per le attività Amazon ECS ospitate su Fargate.

Quando registri una definizione di attività, devi specificare la famiglia del sistema operativo.

I valori validi sono `LINUX`, `WINDOWS_SERVER_2019_FULL`, `WINDOWS_SERVER_2019_CORE`, `WINDOWS_SERVER_2022_FULL` e `WINDOWS_SERVER_2022_CORE`.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Quando una definizione di attività fa parte di un servizio, questo valore deve corrispondere al valore `platformFamily` del servizio.

`cpuArchitecture`

Tipo: stringa

Obbligatorio: condizionale

Valore di default: `X86_64`

Se il parametro viene lasciato come `null`, il valore predefinito viene assegnato automaticamente all'avvio di un'attività ospitata su Fargate.

Quando registri una definizione di attività, devi specificare l'architettura della CPU. I valori validi sono X86_64 e ARM64.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Quando si eseguono attività su Linux, è possibile impostare il valore su ARM64. Per ulteriori informazioni, consulta [the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”](#).

Dimensioni processo

Quando registri una definizione di attività, puoi specificare la quantità totale di CPU e memoria utilizzata per l'attività. Questo valore è separato dai valori `cpu` e `memory` a livello di definizione del container. Per le attività ospitate su Fargate (sia Linux sia Windows), questi campi sono obbligatori e sono supportati valori specifici per `cpu` e `memory`.

Note

I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows. Ti consigliamo di specificare risorse a livello di container per i container Windows.

Il parametro seguente è permesso in una definizione di attività:

`cpu`

Tipo: stringa

Obbligatorio: condizionale

Note

Questo parametro non è supportato per i container Windows.

Il limite rigido di unità CPU da presentare per il processo. È possibile specificare i valori della CPU nel file JSON come stringa in unità CPU o in modalità virtuale CPUs (vCPUs). Ad esempio, è

possibile specificare un valore di CPU come 1024 in unità CPU o 1 vCPU in vCPUs. Quando la definizione dell'attività viene registrata, un valore vCPU viene convertito in un numero intero che indica le unità CPU.

Questo campo è obbligatorio ed è necessario utilizzare uno dei seguenti valori, che determina l'intervallo di valori supportati per il memory parametro. La tabella seguente illustra le combinazioni valide di CPU e memoria a livello di attività.

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
256 (0,25 vCPU)	512 MiB, 1 GB, 2 GB	Linux
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB	Linux
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB	Linux, Windows
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows
8192 (8 vCPU)	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
 Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.		
16384 (16vCPU)	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux
 Note Questa opzione richiede la piattaforma		

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
ma Linux 1.4.0 o successiva.		

memory

Tipo: stringa

Obbligatorio: condizionale

 Note

Questo parametro non è supportato per i container Windows.

Il limite rigido di memoria da presentare all'attività. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibyte (MiB) o gigabyte (GB). Ad esempio, è possibile specificare un valore di memoria 3072 in MiB o 3 GB in GB. Quando la definizione di attività è registrata, un valore GB viene convertito in un numero intero che indica il MiB.

Questo campo è obbligatorio ed è necessario utilizzare uno dei seguenti valori, che determina l'intervallo di valori supportati per il cpu parametro:

Valore di memoria (in MiB, con valore equivalente approssimativo in GB)	Valore CPU	Sistemi operativi supportati per Fargate
512 (0,5 GB), 1024 (1 GB), 2048 (2 GB)	256 (0,25 vCPU)	Linux
1024 (1 GB), 2048 (2 GB), 3072 (3 GB), 4096 (4 GB)	512 (0,5 vCPU)	Linux
2048 (2 GB), 3072 (3 GB), 4096 (4 GB), 5120 (5 GB),	1024 (1 vCPU)	Linux, Windows

Valore di memoria (in MiB, con valore equivalente approssimativo in GB)	Valore CPU	Sistemi operativi supportati per Fargate
6144 (6 GB), 7168 (7 GB), 8192 (8 GB)		
Tra 4096 (4 GB) e 16384 (16 GB) in incrementi di 1024 (1 GB)	2048 (2 vCPU)	Linux, Windows
Tra 8192 (8 GB) e 30720 (30 GB) in incrementi di 1024 (1 GB)	4096 (4 vCPU)	Linux, Windows
Tra 16 GB e 60 GB in incrementi di 4 GB	8192 (8 vCPU)	Linux
 Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.		
Tra 32 GB e 120 GB in incrementi di 8 GB	16384 (16vCPU)	Linux
 Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.		

Definizioni del container

Quando si registra una definizione di attività, è necessario specificare un elenco di definizioni di contenitore che vengono passate a Docker demone su un'istanza del contenitore. I seguenti parametri sono consentiti in una definizione del container.

Argomenti

- [Parametri standard di definizione del container](#)
- [Parametri avanzati di definizione del container](#)
- [Altri parametri di definizione del container](#)

Parametri standard di definizione del container

I seguenti parametri di definizione di attività sono obbligatori o utilizzati nella maggior parte delle definizioni del container.

Argomenti

- [Nome](#)
- [Immagine](#)
- [Memoria](#)
- [Mappature di porte](#)
- [Credenziali del repository privato](#)

Nome

name

Tipo: stringa

Campo obbligatorio: sì

Il nome di un container. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. Se colleghi più container in una definizione di attività, il parametro name di uno dei container può essere inserito nel parametro links di un altro container. Questo per collegare i container.

Immagine

image

Tipo: stringa

Campo obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene passata direttamente al Docker demone. Per impostazione predefinita, le immagini in Docker Hub il registro è disponibile. Puoi anche specificare altri repository con *repository-url/image:tag* o *repository-url/image@digest*. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole); sono consentiti numeri, trattini, caratteri di sottolineatura, due punti, punti, barre e cancelletti. Questo parametro è Image mappato al comando docker create-container e al parametro IMAGE del comando docker run.

- Quando viene avviato un nuovo processo, l'agente del container Amazon ECS esegue il pull della versione più recente dell'immagine e del tag specificati per il container da utilizzare. Tuttavia, gli aggiornamenti successivi a un'immagine del repository non vengono propagate alle attività già in esecuzione.
- Quando non specifichi un tag o un digest nel percorso dell'immagine nella definizione dell'attività, l'agente container Amazon ECS recupera la versione più recente dell'immagine specificata.
- Tuttavia, gli aggiornamenti successivi a un'immagine del repository non vengono propagate alle attività già in esecuzione.
- Le immagini nei registri privati sono supportate. Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).
- Le immagini nei repository Amazon ECR possono essere specificate utilizzando la convenzione di denominazione registry/repository:tag o registry/repository@digest, ad esempio, *aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest* o *aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app@sha256:94afd1f2e64d908bc90dbca0035a5b567EXAMPLE*.
- Immagini nei repository ufficiali su Docker Hub usa un solo nome (ad esempio, ubuntu omongo).
- Immagini in altri repository su Docker Hub sono qualificati con il nome di un'organizzazione (ad esempio, amazon/amazon-ecs-agent).
- Le immagini in altri archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, quay.io/assemblyline/ubuntu).

versionConsistency

Tipo: stringa

Valori validi: enabled|disabled

Campo obbligatorio: no

Specifica se Amazon ECS risolverà il tag dell'immagine del contenitore fornito nella definizione del contenitore in un digest di immagini. Per impostazione predefinita, questo comportamento è `enabled`. Se imposti il valore per un contenitore come `disabled`, Amazon ECS non risolverà il tag dell'immagine del contenitore in un digest e utilizzerà l'URI dell'immagine originale specificato nella definizione del contenitore per la distribuzione. Per ulteriori informazioni sulla risoluzione dell'immagine del contenitore, consulta [Risoluzione dell'immagine del contenitore](#)

Memoria

memory

Tipo: integer

Campo obbligatorio: no

La quantità (in MiB) della memoria da presentare al container. Se il container tenta di superare la memoria specificata qui, viene terminato. La quantità totale di memoria prenotata per tutti i container all'interno di un processo deve essere inferiore al valore `memory` del processo, se specificato. Questo parametro è mappato `Memory` al comando `docker create-container` e `--memory` all'opzione `docker run`.

Il daemon Docker 20.10.0 o versione successiva prenota un minimo di 6 MiB di memoria per un container. Pertanto, non specificare meno di 6 MiB di memoria per i container.

Il daemon Docker 19.03.13-ce o versione precedente prenota un minimo di 4 MiB di memoria per un container. Pertanto, non specificare meno di 4 MiB di memoria per i container.

Note

Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

memoryReservation

Tipo: integer

Campo obbligatorio: no

Il limite flessibile (in MiB) della memoria da prenotare per il container. Quando la memoria di sistema è in conflitto, Docker tenta di mantenere la memoria del contenitore entro questo limite flessibile. Tuttavia, il container può utilizzare una quantità maggiore di memoria, se necessario. Il container può utilizzare memoria fino al limite rigido specificato con il parametro `memory` (se applicabile) o tutta la memoria disponibile sull'istanza di container, a seconda di quale evento si verifica prima. Questo parametro è mappato `MemoryReservation` al comando `docker create-container` e `--memory-reservation` all'opzione `docker run`.

Se non viene specificato un valore di memoria a livello di attività, è necessario specificare un numero intero diverso da zero per uno o entrambi i codici `memory` o `memoryReservation` in una definizione del container. Se specifichi entrambe, `memory` deve essere superiore a `memoryReservation`. Se specifichi `memoryReservation`, tale valore viene sottratto dalle risorse di memoria disponibili per l'istanza di container in cui il container viene posizionato. In caso contrario, viene utilizzato il valore `memory`.

Ad esempio, supponiamo che il container normalmente utilizza 128 MiB di memoria, ma con picchi occasionali di 256 MiB di memoria per brevi periodi di tempo. Puoi impostare un valore di `memoryReservation` di 128 MiB e un limite rigido `memory` di 300 MiB. Questa configurazione consente al container di riservare 128 MiB di memoria solo dalle risorse restanti nell'istanza di container. Allo stesso tempo, questa configurazione consente al container di utilizzare più risorse di memoria quando necessario.

Note

Questo parametro non è supportato per i container Windows.

Il daemon Docker 20.10.0 o versione successiva prenota un minimo di 6 MiB di memoria per un container. Pertanto, non specificare meno di 6 MiB di memoria per i container.

Il daemon Docker 19.03.13-ce o versione precedente prenota un minimo di 4 MiB di memoria per un container. Pertanto, non specificare meno di 4 MiB di memoria per i container.

 Note

Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

Mappature di porte

portMappings

Tipo: array di oggetti

Campo obbligatorio: no

Le mappature delle porte espongono le porte di rete del contenitore al mondo esterno. Ciò consente ai client di accedere all'applicazione. Viene anche utilizzato per la comunicazione tra container nell'ambito della stessa attività.

Per le definizioni di attività che utilizzano la modalità di rete `awsvpc`, specifica solo il parametro `containerPort`. `hostPort` viene sempre ignorato e la porta del contenitore viene automaticamente mappata su una porta casuale con numero alto sull'host.

La mappatura delle porte su Windows usa l'indirizzo gateway `NetNAT` anziché `localhost`. Non vi è alcun loopback per le mappature delle porte su Windows, perciò non è possibile accedere alla porta mappata di un container dall'host stesso.

La maggior parte dei campi di questo parametro (incluso `containerPort`, `hostPort`, `protocol`) è mappata `PortBindings` al comando `docker create-container` e `--publish` all'opzione `docker run`. Se la modalità di rete di una definizione di attività è impostata su `host`, le porte `host` devono essere non definite o devono corrispondere alla porta del container nella mappatura della porta.

 Note

Dopo che un processo raggiunge lo stato `RUNNING`, gli incarichi manuali e automatici relativi alle porte del container e dell'host sono visibili nelle posizioni seguenti:

- Console: sezione `Binding di rete` della descrizione di un container per un processo selezionato.
- AWS CLI: la sezione `networkBindings` dell'output del comando `describe-tasks`.

- API: risposta `DescribeTasks`.
- Metadati: l'endpoint dei metadati dell'attività.

`appProtocol`

Tipo: `string`

Campo obbligatorio: no

Il protocollo dell'applicazione utilizzato per la mappatura delle porte. Questo parametro si applica solo a Service Connect. Ti consigliamo di impostare questo parametro in maniera coerente con il protocollo utilizzato dall'applicazione. Se imposti questo parametro, Amazon ECS aggiunge la gestione delle connessioni specifica del protocollo al proxy Service Connect. Se imposti questo parametro, Amazon ECS aggiunge la telemetria specifica del protocollo nella console Amazon ECS e. CloudWatch

Se non imposti un valore per questo parametro, viene utilizzato TCP. Amazon ECS, tuttavia, non aggiunge la telemetria specifica del protocollo TCP.

Per ulteriori informazioni, consulta [the section called “ Service Connect”](#).

Valori di protocollo validi: `"http"` | `"http2"` | `"grpc"`

`containerPort`

Tipo: `integer`

Obbligatorio: sì, quando si utilizzano `portMappings`

Il numero di porta nel container associato alla porta dell'host definito dall'utente o assegnata automaticamente.

Per le attività che utilizzano la modalità di `awsvpc` rete, è necessario specificare le porte esposte. `containerPort`

Per i container Windows su Fargate, non è possibile utilizzare la porta 3150 per `containerPort`. Questo perché la porta è riservata.

`containerPortRange`

Tipo: `string`

Campo obbligatorio: no

L'intervallo dei numeri di porta nel container associato all'intervallo di porte host mappato in maniera dinamica.

È possibile impostare questo parametro solo utilizzando l'API `register-task-definition`. L'opzione è disponibile nel parametro `portMappings`. Per ulteriori informazioni, consulta [register-task-definition](#) nella documentazione di riferimento AWS Command Line Interface .

Quando specifichi un `containerPortRange`, si applicano le seguenti regole:

- È necessario utilizzare la modalità `awsvpc` di rete.
- Questo parametro è disponibile per sistemi operativi sia Linux che Windows.
- L'istanza di container deve avere almeno la versione 1.67.0 dell'agente del container e almeno la versione 1.67.0-1 del pacchetto `ecs-init`.
- Puoi specificare fino a 100 intervalli di porte per container.
- Non specificare un `hostPortRange`. Il valore dell'`hostPortRange` è impostato come indicato di seguito:
 - Per i container in un'attività con la modalità di rete `awsvpc`, la `hostPort` è impostata sullo stesso valore della `containerPort`. Questa è una strategia di mappatura statica.
- I valori validi di `containerPortRange` sono compresi tra 1 e 65535.
- Una porta può essere inclusa solo in una sola mappatura delle porte per ogni container.
- Non puoi specificare intervalli di porte sovrapposti.
- La prima porta nell'intervallo deve essere minore dell'ultima porta nell'intervallo.
- Docker consiglia di disattivare il `docker-proxy` nel Docker file di configurazione del demone quando si dispone di un numero elevato di porte.

Per ulteriori informazioni, consultate il [numero #11185 su](#). GitHub

Per informazioni su come disattivare il `docker-proxy` nel Docker file di configurazione del demone, vedere [Docker daemon](#) nella Amazon ECS Developer Guide.

Puoi effettuare la chiamata a [DescribeTasks](#) per visualizzare l'`hostPortRange`, cioè le porte dell'host associate alle porte del container.

Gli intervalli di porte non sono inclusi negli eventi delle attività di Amazon ECS, a EventBridge cui vengono inviati. Per ulteriori informazioni, consulta [the section called “Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge”](#).

hostPortRange

Tipo: string

Campo obbligatorio: no

L'intervallo di numeri di porta sull'host utilizzato con il collegamento di rete. Questo viene assegnato da Docker e fornito dall'agente Amazon ECS.

hostPort

Tipo: integer

Campo obbligatorio: no

Il numero di porta nell'istanza di container per prenotare per il container.

hostPortPuò essere lasciato vuoto o avere lo stesso valore di containerPort.

L'intervallo di porte temporanee predefinito Docker la versione 1.6.0 e successive è elencata nell'istanza sotto. `/proc/sys/net/ipv4/ip_local_port_range` Se questo parametro kernel non è disponibile, viene utilizzato l'intervallo delle porte temporaneo di default da 49153–65535. Non tentare di specificare una porta dell'host nell'intervallo di porte effimere. Questo perché sono riservate per l'assegnazione automatica. In generale, le porte al di sotto di 32768 non rientrano nell'intervallo delle porte temporanee.

Le porte riservate predefinite sono 22 per SSH, le Docker porte 2375 e 2376 porte Amazon ECS Container Agent. 51678–51680 Qualsiasi porta dell'host precedentemente specificata dall'utente per un'attività in esecuzione viene prenotata anche mentre tale attività è in esecuzione. Dopo l'arresto di un'attività, la porta dell'host viene rilasciata. Le porte prenotate correnti vengono visualizzate nel parametro `remainingResources` dell'output `describe-container-instances`. Un'istanza di container può contenere fino a 100 porte prenotate alla volta, incluse quelle predefinite. Le porte assegnate automaticamente non vengono conteggiate ai fini della quota di 100 porte prenotate.

name

Tipo: stringa

Obbligatorio: No, richiesto per la configurazione di Service Connect e VPC Lattice in un servizio

Il nome utilizzato per la mappatura delle porte. Questo parametro si applica solo a Service Connect e VPC Lattice. Questo parametro è il nome utilizzato nella configurazione Service Connect e VPC Lattice di un servizio.

Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

Nell'esempio seguente, vengono utilizzati entrambi i campi obbligatori per Service Connect e VPC Lattice.

```
"portMappings": [  
  {  
    "name": string,  
    "containerPort": integer  
  }  
]
```

protocol

Tipo: string

Campo obbligatorio: no

Il protocollo utilizzato per la mappatura delle porte. I valori validi sono tcp e udp. Il valore predefinito è tcp.

Important

È supportato solo tcp per Service Connect. Ricorda che tcp è implicito se questo campo non è impostato.

Se specifichi una porta dell'host, utilizza la seguente sintassi.

```
"portMappings": [  
  {  
    "containerPort": integer,  
    "hostPort": integer  
  }  
]
```

```
    ...  
  ]
```

Se desideri una porta dell'host assegnata automaticamente, utilizza la seguente sintassi.

```
"portMappings": [  
  {  
    "containerPort": integer  
  }  
  ...  
]
```

Credenziali del repository privato

`repositoryCredentials`

Tipo: oggetto [RepositoryCredentials](#)

Campo obbligatorio: no

Le credenziali dell'archivio per l'autenticazione di un registro privato.

Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

`credentialsParameter`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `repositoryCredentials`

L'Amazon Resource Name (ARN) del segreto contenente le credenziali dell'archivio privato.

Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

Note

Quando utilizzi l'API Amazon ECS o AWS SDKs se il segreto esiste nella stessa regione dell'attività che stai avviando, puoi utilizzare l'ARN completo o il nome del segreto. AWS CLI Quando si utilizza il AWS Management Console, è necessario specificare l'ARN completo del segreto.

Di seguito viene riportato un frammento di una definizione di attività che mostra i parametri obbligatori:

```
"containerDefinitions": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {  
      "credentialsParameter":  
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"  
    }  
  }  
]
```

Parametri avanzati di definizione del container

I seguenti parametri avanzati di definizione dei contenitori forniscono funzionalità estese al comando docker run utilizzato per avviare contenitori sulle istanze di container Amazon ECS.

Argomenti

- [Politica di riavvio](#)
- [Controllo dello stato](#)
- [Ambiente](#)
- [Impostazioni di rete](#)
- [Archiviazione e registrazione](#)
- [Sicurezza](#)
- [Limiti delle risorse](#)
- [Etichette Docker](#)

Politica di riavvio

restartPolicy

La politica di riavvio del contenitore e i parametri di configurazione associati. Quando configuri una politica di riavvio per un contenitore, Amazon ECS può riavviare il contenitore senza dover

sostituire l'attività. Per ulteriori informazioni, consulta [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#).

`enabled`

Tipo: Booleano

Campo obbligatorio: sì

Specifica se è abilitata una politica di riavvio per il contenitore.

`ignoredExitCodes`

Tipo: array di numeri interi

Campo obbligatorio: no

Un elenco di codici di uscita che Amazon ECS ignorerà e non tenterà di riavviare. Puoi specificare un massimo di 50 codici di uscita del contenitore. Per impostazione predefinita, Amazon ECS non ignora alcun codice di uscita.

`restartAttemptPeriod`

Tipo: integer

Campo obbligatorio: no

Un periodo di tempo (in secondi) durante il quale il contenitore deve funzionare prima di poter tentare un riavvio. Un contenitore può essere riavviato solo una volta ogni secondo. `restartAttemptPeriod` Se un contenitore non è in grado di funzionare per questo periodo di tempo e esce in anticipo, non verrà riavviato. È possibile impostare un minimo `restartAttemptPeriod` di 60 secondi e un massimo `restartAttemptPeriod` di 1800 secondi. Per impostazione predefinita, un contenitore deve funzionare per 300 secondi prima di poter essere riavviato.

Controllo dello stato

`healthCheck`

Comando di controllo dell'integrità del container e parametri di configurazione associati per il container. Per ulteriori informazioni, consulta [Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container](#).

command

Matrice di stringhe che rappresenta il comando eseguito dal container per determinare l'integrità. La matrice di stringhe può iniziare con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell predefinita del container. Se non è specificato nessuno dei due, viene utilizzato `CMD`.

Quando si registra una definizione di attività in AWS Management Console, utilizzare un elenco di comandi separati da virgole. Questi comandi vengono convertiti in stringa dopo la creazione della definizione delle attività. Di seguito è riportato un esempio di input per il controllo dell'integrità.

```
CMD-SHELL, curl -f http://localhost/ || exit 1
```

Quando si registra una definizione di attività utilizzando il pannello AWS Management Console JSON, o il AWS CLI APIs, racchiude l'elenco dei comandi tra parentesi. Di seguito è riportato un esempio di input per il controllo dell'integrità.

```
[ "CMD-SHELL", "curl -f http://localhost/ || exit 1" ]
```

Un codice di uscita 0, senza output `stderr`, indica l'esito positivo, mentre un codice di uscita diverso da zero indica un errore.

interval

Intervallo di tempo (in secondi) tra ogni controllo dell'integrità. Puoi specificare un valore compreso tra 5 e 300 secondi. Il valore di predefinito è 30 secondi.

timeout

Periodo di tempo in secondi per cui attendere che un controllo dell'integrità venga superato prima di considerarlo un errore. Puoi specificare un valore compreso tra 2 e 60 secondi. Il valore di default è 5 secondi.

retries

Numero di tentativi per cui riprovare un controllo dello stato non riuscito prima che il container venga considerato non integro. Puoi specificare un valore compreso tra 1 e 10 tentativi. Il valore predefinito è tre tentativi.

startPeriod

Periodo di tolleranza facoltativo entro il quale concedere ai container il tempo necessario per il bootstrap prima che i controlli dell'integrità non riusciti vengano conteggiati rispetto al numero massimo di nuovi tentativi. È possibile specificare un valore compreso tra 0 e 300 secondi. Per impostazione predefinita, `startPeriod` è disabilitato.

Se un controllo dello stato va a buon fine all'interno di `startPeriod`, il container è considerato integro e gli errori successivi vengono conteggiati rispetto al numero massimo di nuovi tentativi.

Ambiente

cpu

Tipo: integer

Campo obbligatorio: no

Il numero di unità `cpu` che l'agente del container Amazon ECS riserverà per il container. In Linux, questo parametro è `CpuShares` mappato alla sezione [Crea un contenitore](#).

Questo campo è facoltativo solo per le attività che utilizzano il tipo di avvio Fargate. La quantità totale di CPU riservata per tutti i container all'interno di un'attività deve essere inferiore al valore `cpu` a livello di attività.

Linux i contenitori condividono le unità CPU non allocate con altri contenitori sull'istanza del contenitore con lo stesso rapporto della loro quantità allocata. Ad esempio, supponiamo di eseguire un'attività di container singolo in un tipo di istanza single core con 512 unità di CPU specificate per tale container. Inoltre, tale attività è l'unica in esecuzione sull'istanza di container. In questo esempio, il container può utilizzare la condivisione completa di 1.024 unità CPU in qualsiasi momento. Tuttavia, si supponga di aver avviato un'altra copia della stessa attività su quell'istanza di container. A ogni attività viene garantito un minimo di 512 unità CPU quando necessario. Analogamente, se l'altro container non utilizza la CPU rimanente, ogni container può passare a un maggiore utilizzo della CPU. Tuttavia, se entrambe le attività sono sempre attive al 100%, sono limitate a 512 unità CPU.

Abilitato Linux istanze del contenitore, il Docker il demone sull'istanza del contenitore utilizza il valore della CPU per calcolare i relativi rapporti di condivisione della CPU per i contenitori in esecuzione. Il valore minimo di condivisione della CPU valido consentito dal kernel Linux è 2 e il

valore massimo di condivisione della CPU valido consentito dal kernel Linux è 262144. Tuttavia, il parametro CPU non è richiesto ed è possibile utilizzare valori della CPU inferiori a due e superiori a 262144 nelle definizioni dei contenitori. Per i valori della CPU inferiori a due (incluso null) e superiori a 262144, il comportamento varia in base alla versione dell'agente container Amazon ECS:

Abilitato Windows nelle istanze di container, la quota di CPU viene applicata come quota assoluta. Windows i contenitori hanno accesso solo alla quantità specificata di CPU definita nella definizione dell'attività. Un valore CPU nullo o pari a zero viene passato a Docker come 0. Windows quindi interpreta questo valore come 1% di una CPU.

Per ulteriori esempi, consulta [Modalità di gestione delle risorse di CPU e memoria di Amazon ECS](#).

gpu

Questo parametro non è supportato per i contenitori ospitati su Fargate.

Tipo: oggetto [ResourceRequirement](#)

Campo obbligatorio: no

Il numero di GPUs fisiche che l'agente del container Amazon ECS riserva per il container. Il numero di contenitori GPUs riservati per tutti i contenitori di un'attività non deve superare il numero di quelli disponibili GPUs sull'istanza del contenitore su cui viene avviata l'attività. Per ulteriori informazioni, consulta [Definizioni di attività Amazon ECS per carichi di lavoro GPU](#).

Elastic Inference accelerator

Questo parametro non è supportato per i contenitori ospitati su Fargate.

Tipo: oggetto [ResourceRequirement](#)

Campo obbligatorio: no

Per il tipo InferenceAccelerator, il value corrisponde a deviceName per un InferenceAccelerator specificato in una definizione di attività. Per ulteriori informazioni, consulta [the section called “Nome dell'acceleratore di inferenza elastica”](#).

essential

Tipo: Booleano

Campo obbligatorio: no

Si supponga che il parametro `essential` di un container sia contrassegnato come `true` e che tale container abbia esito negativo o si arresti per qualsiasi motivo. Di conseguenza, tutti gli altri container che fanno parte dell'attività vengono arrestati. Se il parametro `essential` di un container è contrassegnato come `false`, il suo esito negativo non influenza il resto dei container in un'attività. Se questo parametro viene omesso, un container si considera essenziale.

Tutti i processi devono avere almeno un container essenziale. Supponiamo di avere un'applicazione composta da più container. In questo caso, si raggruppano i container utilizzati per uno scopo comune in componenti e si separano i diversi componenti in più definizioni delle attività. Per ulteriori informazioni, consulta [Progetta la tua applicazione per Amazon ECS](#).

```
"essential": true|false
```

entryPoint

Important

Le versioni precedenti dell'agente del container Amazon ECS non gestiscono correttamente i parametri `entryPoint`. In caso di problemi durante l'utilizzo `entryPoint`, aggiorna l'agente del container o inserisci i comandi e gli argomenti come elementi di matrice `command`.

Tipo: array di stringhe

Campo obbligatorio: no

Il punto di ingresso che viene trasmesso al container.

```
"entryPoint": ["string", ...]
```

command

Tipo: array di stringhe

Campo obbligatorio: no

Il comando che viene inviato al container. Questo parametro è `Cmd` mappato al comando `create-container` e il `COMMAND` parametro a `docker run`. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

workingDirectory

Tipo: string

Campo obbligatorio: no

La directory di lavoro nel container in cui eseguire i comandi. Questo parametro è mappato a `WorkingDir` nella sezione [Create a container](#) di [Docker Remote API](#) e l'opzione `--workdir` a [docker run](#).

```
"workingDirectory": "string"
```

environmentFiles

Non è disponibile per i contenitori Windows su Fargate.

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'opzione `--env-file` del comando `docker run`.

Puoi specificare fino a 10 file di ambiente. Il file deve avere un'estensione `.env`. Ogni riga di un file di ambiente deve contenere una variabile di ambiente nel formato `VARIABLE=VALUE`. Le righe che iniziano con `#` vengono trattate come commenti e vengono ignorate.

Se nella definizione del container sono specificate singole variabili di ambiente, hanno la precedenza sulle variabili contenute in un file di ambiente. Se vengono specificati più file di ambiente che contengono la stessa variabile, vengono elaborati dall'alto verso il basso. Consigliamo di utilizzare nomi di variabili univoci. Per ulteriori informazioni, consulta [Passa una singola variabile di ambiente a un contenitore Amazon ECS](#).

value

Tipo: stringa

Campo obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

`type`

Tipo: stringa

Campo obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è s3.

`environment`

Tipo: array di oggetti

Campo obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è Env mappato al comando `docker create-container` e l'opzione `--env` al comando `docker run`.

 Important

Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

`name`

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

`value`

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il valore della variabile di ambiente.

```
"environment" : [
```



```
{ "name" : "string", "value" : "string" },  
{ "name" : "string", "value" : "string" }  
]
```

secrets

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

name

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del AWS Secrets Manager segreto o l'ARN completo del parametro nel Parameter Store. AWS Systems Manager

Note

Se il parametro Systems Manager Parameter Store o il parametro Secrets Manager esiste nella Regione AWS stessa operazione che si sta avviando, è possibile utilizzare l'ARN completo o il nome del segreto. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

```
"secrets": [  
  {  
    "name": "environment_variable_name",
```

```
    "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
  }
]
```

Impostazioni di rete

disableNetworking

Questo parametro non è supportato per le attività in esecuzione su Fargate.

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è true, le reti sono disabilite all'interno del container.

Il valore predefinito è false.

```
"disableNetworking": true|false
```

links

Questo parametro non è supportato per le attività che utilizzano la modalità aws-vpc di rete.

Tipo: array di stringhe

Campo obbligatorio: no

Il parametro `link` consente ai container di comunicare tra loro senza la necessità di mappatura delle porte. Questo parametro è supportato solo se la modalità di rete di una definizione delle attività è impostata su `bridge`. Il `name:internalName` costruito è analogo a `name:alias` Docker collegamenti. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura.

Important

I container che vengono posizionati nella stessa istanza di container potrebbero comunicare tra loro senza necessità di collegamenti o mappature delle porte dell'host. L'isolamento di rete su un'istanza di container è controllato da gruppi di sicurezza e impostazioni VPC.

```
"links": ["name:internalName", ...]
```

hostname

Tipo: string

Campo obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato `Hostname` al docker `create-container` e `--hostname` all'opzione docker `run`.

Note

Se utilizzi la modalità di rete `awsvpc`, il parametro `hostname` non è supportato.

```
"hostname": "string"
```

dnsServers

Questa funzionalità non è supportata per le attività in esecuzione su Fargate.

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di server DNS presentato al container.

```
"dnsServers": ["string", ...]
```

extraHosts

Questo parametro non è supportato per le attività che utilizzano la modalità `awsvpc` di rete.

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato ExtraHosts al comando docker create-container e --add-host all'opzione docker run.

```
"extraHosts": [  
  {  
    "hostname": "string",  
    "ipAddress": "string"  
  },  
  ...  
]
```

hostname

Tipo: stringa

Obbligatorio: sì, quando si utilizzano extraHosts

Il nome host da utilizzare nella voce /etc/hosts.

ipAddress

Tipo: stringa

Obbligatorio: sì, quando si utilizzano extraHosts

L'indirizzo IP da utilizzare nella voce /etc/hosts.

Archiviazione e registrazione

readonlyRootFilesystem

Tipo: Booleano

Campo obbligatorio: no

Se il parametro è true, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro associa ReadonlyRootfs nel comando docker create-container l'opzione --read-only docker run.

Note

Questo parametro non è supportato per i container Windows.

Il valore predefinito è `false`.

```
"readonlyRootFilesystem": true|false
```

mountPoints

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato `Volumes` all'API Docker `create-container` e `--volume` all'opzione `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

sourceVolume

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

containerPath

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

readOnly

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Per le attività eseguite su EC2 istanze che eseguono il sistema operativo Windows, lasciate il valore predefinito di `false`.

`volumesFrom`

Tipo: array di oggetti

Campo obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato `VolumesFrom` al comando docker `create-container` e `--volumes-from` all'opzione docker `run`.

`sourceContainer`

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

```
"volumesFrom": [  
  {  
    "sourceContainer": "string",  
    "readOnly": true|false  
  }  
]
```

`logConfiguration`

Tipo: oggetto [LogConfiguration](#)

Campo obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta [Esempi di definizioni di attività Amazon ECS](#).

Questo parametro è mappato LogConfig al comando docker create-container e --log-driver all'opzione docker run. Per impostazione predefinita, i contenitori utilizzano lo stesso driver di registrazione del Docker daemon utilizza. Tuttavia, il contenitore potrebbe utilizzare un driver di registrazione diverso dal Docker daemon specificando un driver di registro con questo parametro nella definizione del contenitore. Per utilizzare un altro driver di log per un container, il sistema di log deve essere configurato correttamente nell'istanza di container (o su un altro server di log per le opzioni di logging in remoto).

Si noti quanto segue quando si specifica una configurazione di log per i container:

- Amazon ECS supporta un sottoinsieme dei driver di registrazione disponibili per Docker demone. Ulteriori driver di log potranno essere disponibili nei rilasci futuri dell'agente del container Amazon ECS.
- Questo parametro richiede la versione 1.18 o successiva di Docker API remota sull'istanza del contenitore.
- È necessario installare qualsiasi software aggiuntivo al di fuori dell'attività. Ad esempio, Fluentd aggregatori di output o un host remoto che esegue Logstash da inviare Gelf accede a.

```
"logConfiguration": {
  "logDriver": "awslogs","fluentd","gelf","json-
file","journald","splunk","syslog","awsfirelens",
  "options": {"string": "string"
    ...},
  "secretOptions": [{
    "name": "string",
    "valueFrom": "string"
  }]
}
```

logDriver

Tipo: stringa

Valori validi: "awslogs","fluentd","gelf","json-file","journald","splunk","syslog","awsfirelens"

Obbligatorio: sì, quando viene utilizzato logConfiguration

Il driver di log da utilizzare per il container. Per impostazione predefinita, i valori validi elencati in precedenza sono driver di log con i quali l'agente del container Amazon ECS può comunicare.

I driver di registro supportati sono `awslogssplunk`, `eawsfirelens`.

Per ulteriori informazioni su come utilizzare il driver di `awslogs` registro nelle definizioni delle attività per inviare i log dei contenitori a CloudWatch Logs, vedere. [Invia i log di Amazon ECS a CloudWatch](#)

Per ulteriori informazioni sull'utilizzo del driver di log `awsfirelens`, consulta [Routing di log personalizzato](#).

Note

Se disponi di un driver personalizzato che non è elencato, puoi eseguire il fork del progetto Amazon ECS Container Agent [disponibile su GitHub](#) e personalizzarlo in modo che funzioni con quel driver. Ti consigliamo di inviare le richieste pull per le modifiche che desideri siano incluse. Tuttavia, attualmente non forniamo il supporto per eseguire copie modificate di questo software.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

options

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

La mappa chiave/valore delle opzioni di configurazione per inviare il driver di log.

Le opzioni che puoi specificare dipendono dal driver di registro. Alcune delle opzioni che puoi specificare quando utilizzi il `awslogs` router per indirizzare i log ad Amazon CloudWatch includono:

`awslogs-create-group`

Campo obbligatorio: no

Specifica se desideri che il gruppo di log venga creato automaticamente. Se questa opzione non è specificata, viene impostata in modo predefinito su `false`.

 Note

La tua policy IAM deve includere l'autorizzazione `logs:CreateLogGroup` prima di provare a utilizzare `awslogs-create-group`.

`awslogs-region`

Campo obbligatorio: sì

Specificate il Regione AWS driver di `awslogs` registro a cui inviare i log Docker. Puoi scegliere di inviare tutti i log dai cluster di diverse regioni a una singola regione in Logs. CloudWatch In questo modo saranno tutti visibili in un'unica posizione. In caso contrario, puoi separarli per regione per una maggiore granularità. Assicurati che il gruppo di log specificato esista nella regione definita tramite questa opzione.

`awslogs-group`

Campo obbligatorio: sì

Assicurati di specificare un gruppo di log a cui il driver di log `awslogs` invia i flussi di log.

`awslogs-stream-prefix`

Campo obbligatorio: sì

Utilizza l'opzione `awslogs-stream-prefix` per associare un flusso di log al prefisso selezionato, al nome del container e all'ID dell'attività Amazon ECS al quale appartiene il container. Se specifichi un prefisso con questa opzione, il flusso di log assume il formato seguente.

`prefix-name/container-name/ecs-task-id`

Se non specifichi un prefisso con questa opzione, il flusso di log viene denominato secondo l'ID del container assegnato dal daemon Docker nell'istanza di container. Poiché è difficile ricollegare i log al container che li ha inviati con il solo ID container Docker (disponibile solo sulle istanze di container), è consigliabile specificare un prefisso con questa opzione.

Per i servizi Amazon ECS, puoi utilizzare il nome del servizio come prefisso. In questo modo puoi ricollegare i flussi di log al servizio a cui appartiene il container, al nome del container che li ha inviati e all'ID dell'attività a cui appartiene il container.

Perché i log compaiano nell'apposito riquadro della console Amazon ECS, devi specificare un prefisso per il flusso dei log.

`awslogs-datetime-format`

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea nel formato `strftime` Python. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello. In questo modo la riga associata è il delimitatore tra i messaggi di log.

Un esempio di un caso d'uso per l'utilizzo di questo formato è per l'analisi di output, ad esempio uno dump dello stack, che potrebbe altrimenti essere registrato in più voci. Il modello corretto consente di acquisirlo in una sola voce.

Per ulteriori informazioni, consulta [awslogs-datetime-format](#).

Non è possibile configurare entrambe le opzioni `awslogs-datetime-format` e `awslogs-multiline-pattern`.

Note

Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

`awslogs-multiline-pattern`

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea che utilizza un'espressione regolare. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello. In questo modo la riga associata è il delimitatore tra i messaggi di log.

Per ulteriori informazioni, consulta [awslogs-multiline-pattern](#).

Questa opzione viene ignorata se anche `awslogs-datettime-format` è configurato.

Non è possibile configurare entrambe le opzioni `awslogs-datettime-format` e `awslogs-multiline-pattern`.

 Note

Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

Le seguenti opzioni si applicano a tutti i driver di registro supportati.

`mode`

Campo obbligatorio: no

Valori validi: `non-blocking` | `blocking`

Questa opzione definisce la modalità di consegna dei messaggi di registro dal contenitore al driver di registro specificato utilizzando `logDriver`. La modalità di consegna scelta influisce sulla disponibilità dell'applicazione quando il flusso di log dal contenitore viene interrotto.

Se si utilizza la `blocking` modalità e il flusso di log viene interrotto, le chiamate provenienti dal codice del contenitore su cui scrivere verso `stdout` e `stderr` gli stream verranno bloccate. Di conseguenza, il thread di registrazione dell'applicazione si bloccherà. Ciò può causare la mancata risposta dell'applicazione e la presenza di errori nel controllo d'integrità del container.

Se utilizzi la modalità `non-blocking`, i log del container vengono invece archiviati in un buffer intermedio in memoria configurato con l'opzione `max-buffer-size`. In questo modo si evita che l'applicazione non risponda quando i log non possono essere inviati. Ti consigliamo di utilizzare questa modalità se vuoi garantire la disponibilità del servizio nonostante una certa perdita di log. Per ulteriori informazioni, vedere [Prevenzione della perdita di log con la modalità non bloccante nel driver di registro del awslogs contenitore](#).

È possibile impostare un valore predefinito `mode` per tutti i contenitori in uno specifico Regione AWS utilizzando le impostazioni dell'`defaultLogDriverModeaccount`. Se

non specifichi l'opzione nell'impostazione dell'account `logConfiguration` o non configuri l'impostazione dell'account, Amazon ECS utilizzerà per impostazione predefinita la `blocking` modalità. Per ulteriori informazioni sull'impostazione dell'account, consulta [Modalità predefinita del driver di registro](#).

Quando viene utilizzata la modalità non bloccante, l'opzione `max-buffer-size` log controlla la dimensione del buffer utilizzato per l'archiviazione intermedia dei messaggi. Assicurati di specificare una dimensione del buffer adeguata in base all'applicazione in uso. Quando il buffer è pieno, non è possibile archiviare ulteriori log. I log che non possono essere archiviati vengono persi. La quantità totale di memoria allocata a livello di attività deve essere maggiore della quantità di memoria allocata per tutti i contenitori oltre al buffer di memoria del driver di registro. Quando il buffer è pieno, non è possibile archiviare ulteriori log. I log che non possono essere archiviati vengono persi. La quantità totale di memoria allocata a livello di attività deve essere maggiore della quantità di memoria allocata per tutti i contenitori oltre al buffer di memoria del driver di registro.

 Note

Il 25 giugno 2025, Amazon ECS cambierà la modalità predefinita del driver di registro da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto alla registrazione. Per continuare a utilizzare la `blocking` modalità dopo questa modifica, esegui una delle seguenti operazioni:

- Imposta l'opzione nella definizione del contenitore `logConfiguration` come `blocking`.
- Imposta l'impostazione `defaultLogDriverMode` dell'account su `blocking`.

`max-buffer-size`

Campo obbligatorio: no

Valore predefinito: 1m

Quando viene utilizzata la modalità `non-blocking`, l'opzione di log `max-buffer-size` controlla la dimensione del buffer utilizzato per l'archiviazione dei messaggi intermedi. Assicurati di specificare una dimensione del buffer adeguata in base all'applicazione in uso. Quando il buffer è pieno, non è possibile archiviare ulteriori log. I log che non possono essere archiviati vengono persi.

Per indirizzare i log utilizzando il `splunk log router`, è necessario specificare `splunk-token` e `asplunk-url`.

Quando si utilizza il router di `awsfirelens log` per indirizzare i log verso una AWS Partner Network destinazione Servizio AWS OR per l'archiviazione e l'analisi dei log, è possibile impostare l'opzione `log-driver-buffer-limit` per limitare il numero di eventi che vengono memorizzati nel buffer prima di essere inviati al contenitore del log router. Può aiutare a risolvere un potenziale problema di perdita di log, poiché un throughput elevato potrebbe causare l'esaurimento della memoria per il buffer interno di Docker. Per ulteriori informazioni, vedere [the section called “Configurazione dei log per un throughput elevato”](#).

Le altre opzioni che è possibile specificare quando si utilizzano i log di `awsfirelens to route` dipendono dalla destinazione. Quando esporti i log in Amazon Data Firehose, puoi specificare Regione AWS il `region` with e un nome per il flusso di log con. `delivery_stream`

Quando esporti i log in Amazon Kinesis Data Streams, puoi specificare Regione AWS un `region` with e un nome per il flusso di dati con. `stream`

Quando esporti i log su Amazon OpenSearch Service, puoi specificare opzioni come `Name`, `Host` (Endpoint del OpenSearch servizio senza protocollo), `Port`, `Index`, `Type`, `Aws_auth`, `Aws_region`, `Suppress_Type_Name`, e. `tls`

Quando esporti i log su Amazon S3, puoi specificare il bucket utilizzando l'opzione. `bucket`
Puoi anche specificare `region`, `total_file_size`, `upload_timeout`, e `use_put_object` come opzioni.

Questo parametro richiede la versione 1.19 o successiva di Docker Remote API sull'istanza di container.

`secretOptions`

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da inviare alla configurazione di log. I segreti utilizzati nella configurazione di log possono includere un token di autenticazione, un certificato o una chiave di crittografia. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

`name`

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre alla configurazione di log del container.

```
"logConfiguration": {
  "logDriver": "splunk",
  "options": {
    "splunk-url": "https://cloud.splunk.com:8080",
    "splunk-token": "...",
    "tag": "...",
    ...
  },
  "secretOptions": [{
    "name": "splunk-token",
    "valueFrom": "/ecs/logconfig/splunkcred"
  }]
}
```

firelensConfiguration

Tipo: [FirelensConfiguration](#) oggetto

Campo obbligatorio: no

La FireLens configurazione per il contenitore. Si utilizza per specificare e configurare un router di log per i log del container. Per ulteriori informazioni, consulta [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

```
{
  "firelensConfiguration": {
    "type": "fluentd",
    "options": {
      "KeyName": ""
    }
  }
}
```

```
}
```

options

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

La mappa chiave/valore delle opzioni da utilizzare durante la configurazione del router di log. Questo campo è facoltativo e può essere utilizzato per aggiungere ulteriori metadati, ad esempio il processo, la definizione di attività, il cluster e i dettagli dell'istanza di container all'evento di log. Se specificato, la sintassi da utilizzare è "options":{"enable-ecs-log-metadata":"true|false", "config-file-type":"s3|file", "config-file-value":"arn:aws:s3:::*amzn-s3-demo-bucket*/fluent.conf|filepath"}. Per ulteriori informazioni, consulta [Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens](#).

type

Tipo: stringa

Campo obbligatorio: sì

Il router di log da utilizzare. I valori validi sono fluentd o fluentbit.

Sicurezza

Per ulteriori informazioni sulla sicurezza dei container, consulta le [procedure consigliate per la sicurezza dei container e delle attività di Amazon ECS](#).

credentialSpecs

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di file ARNs in SSM o Amazon S3 in un file spec CredSpec () delle credenziali che configura il contenitore per l'autenticazione Active Directory. Consigliamo di utilizzare questo parametro anziché dockerSecurityOptions. Il numero massimo di è 1. ARNs

Esistono due formati per ogni ARN.

credentialspecdomainless:MyARN

Utilizza `credentialspecdomainless:MyARN` per fornire un file CredSpec con una sezione aggiuntiva per un segreto in Secrets Manager. Specifica le credenziali di accesso al dominio nel campo segreto.

Ogni attività eseguita su qualsiasi istanza di container può aggiungere domini diversi.

Puoi utilizzare questo formato senza aggiungere l'istanza di container a un dominio.

credentialspec:MyARN

Utilizza `credentialspec:MyARN` per fornire un nome a un file CredSpec per un singolo dominio.

Devi aggiungere l'istanza di container al dominio prima di iniziare qualsiasi attività che utilizzi questa definizione delle attività.

In entrambi i formati, sostituisci MyARN con l'ARN in SSM o Amazon S3.

Il file `credspec` deve fornire un ARN in Secrets Manager per un segreto contenente il nome utente, la password e il dominio a cui collegarsi. Per una maggiore sicurezza, l'istanza non viene aggiunta al dominio per l'autenticazione senza dominio. Le altre applicazioni sull'istanza non possono utilizzare le credenziali senza dominio. Puoi utilizzare questo parametro per eseguire attività sulla stessa istanza, anche se le attività devono aggiungere domini diversi. Per ulteriori informazioni, vedere [Usare g MSAs per Windows Containers](#) e [Using g MSAs for Linux Containers](#).

user

Tipo: string

Campo obbligatorio: no

L'utente da usare all'interno del container. Questo parametro è mappato `User` al comando `docker create-container` e `--user` all'opzione `docker run`.

Important

Quando esegui attività che utilizzano la modalità di rete `host`, non eseguire container utilizzando l'utente `root` (UID 0). Come best practice per la sicurezza, utilizza sempre un utente non `root`.

È possibile specificare `user` utilizzando i seguenti formati. Un eventuale UID o GID deve essere specificato come numero intero positivo.

- `user`
- `user:group`
- `uid`
- `uid:gid`
- `user:gid`
- `uid:group`

 Note

Questo parametro non è supportato per i container Windows.

```
"user": "string"
```

Limiti delle risorse

`ulimits`

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di valori `ulimit` da definire per un container. Questo valore sovrascrive l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è mappato `Ulimits` al comando `docker create-container` e `--ulimit` all'opzione `docker run`.

I processi di Amazon ECS ospitati su Fargate utilizzano i valori del limite di risorse predefinito impostato dal sistema operativo, ad eccezione del parametro del limite di risorse `nofile`. Il limite di risorse `nofile` imposta una restrizione sul numero di file aperti che un container può utilizzare. Su Fargate, il limite flessibile `nofile` predefinito è 65535 mentre il limite rigido è 65535. Puoi impostare i valori di entrambi i limiti fino a 1048576. Per ulteriori informazioni, consulta [Limiti delle risorse dei processi](#).

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

 Note

Questo parametro non è supportato per i container Windows.

```
"ulimits": [  
  {  
    "name":  
    "core"|"cpu"|"data"|"fsize"|"locks"|"memlock"|"msgqueue"|"nice"|"nofile"|"nproc"|"rss"|"rtpr  
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

name

Tipo: stringa

Valori validi: "core" | "cpu" | "data" | "fsize" | "locks" | "memlock" |
"msgqueue" | "nice" | "nofile" | "nproc" | "rss" | "rtprio" | "rttime"
| "sigpending" | "stack"

Obbligatorio: sì, quando si utilizzano ulimits

type di ulimit.

hardLimit

Tipo: integer

Obbligatorio: sì, quando si utilizzano ulimits

Il limite rigido per il tipo ulimit. Il valore può essere specificato in byte, secondi o come conteggio, a seconda del. type ulimit

softLimit

Tipo: integer

Obbligatorio: sì, quando si utilizzano ulimits

Il limite flessibile per il tipo ulimit. Il valore può essere specificato in byte, secondi o come conteggio, a seconda type del. ulimit

Etichette Docker

`dockerLabels`

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Una mappa chiave/valore di etichette da aggiungere al container. Questo parametro è mappato `Labels` al comando `docker create-container` e `--label` all'opzione `docker run`.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

```
"dockerLabels": {"string": "string"
  ...}
```

Altri parametri di definizione del container

I seguenti parametri di definizione del container possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS utilizzando l'opzione `Configure via JSON` (`Configura tramite JSON`). Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Argomenti

- [Parametri Linux](#)
- [Dipendenze per i container](#)
- [Timeout del container](#)
- [Controlli di sistema](#)
- [Interactive](#)
- [Pseudoterminale](#)

Parametri Linux

`linuxParameters`

Tipo: oggetto [LinuxParameters](#)

Campo obbligatorio: no

Linux-opzioni specifiche che vengono applicate al contenitore, ad esempio. [KernelCapabilities](#)

 Note

Questo parametro non è supportato per Windows contenitori.

```
"linuxParameters": {  
  "capabilities": {  
    "add": ["string", ...],  
    "drop": ["string", ...]  
  }  
}
```

capabilities

Tipo: oggetto [KernelCapabilities](#)

Campo obbligatorio: no

Il Linux funzionalità per il contenitore che vengono eliminate dalla configurazione predefinita fornita da Docker. Per ulteriori informazioni su questi Linux funzionalità, consulta la pagina di manuale di [capabilities \(7\)](#) Linux.

add

Tipo: array di stringhe

Valori validi: "SYS_PTRACE"

Campo obbligatorio: no

Il Linux funzionalità che il contenitore può aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato CapAdd al comando docker create-container e --cap-add all'opzione docker run.

drop

Tipo: array di stringhe

Valori validi: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Campo obbligatorio: no

Il Linux funzionalità per la rimozione del contenitore dalla configurazione predefinita fornita da Docker. Questo parametro è mappato CapDrop al comando docker create-container e --cap-drop all'opzione docker run.

devices

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato Devices al comando docker create-container e --device all'opzione docker run.

Note

Il devices parametro non è supportato quando si utilizza il tipo di avvio Fargate.

Tipo: matrice di oggetti [Device](#)

Campo obbligatorio: no

hostPath

Il percorso per il dispositivo nell'istanza di container dell'host.

Tipo: stringa

Campo obbligatorio: sì

containerPath

Il percorso nel container in cui esporre il dispositivo dell'host.

Tipo: string

Campo obbligatorio: no

`permissions`

Le autorizzazioni esplicite da fornire al container per il dispositivo. Di default, il container dispone di autorizzazioni per `read`, `write` e `mknod` sul dispositivo.

Tipo: matrice di stringhe

Valori validi: `read` | `write` | `mknod`

`initProcessEnabled`

Esegui un processo `init` nel container che inoltra segnali e raccoglie i processi. Questo parametro è mappato all' `--init` opzione `docker run`.

Questo parametro richiede la versione 1.25 o successiva di Docker Remote API sull'istanza di container.

`maxSwap`

Questa funzionalità non è supportata per le attività in esecuzione su Fargate.

La quantità totale di memoria di swap (in MiB) che un container può utilizzare. Questo parametro viene tradotto nell' `--memory-swap` opzione `docker run` dove il valore è la somma della memoria del contenitore più il `maxSwap` valore.

Se viene specificato il valore `maxSwap` di 0, il container non utilizzerà lo swap. I valori accettati sono 0 o qualsiasi numero intero positivo. Se il parametro `maxSwap` viene omesso, il container utilizza la configurazione di swap per l'istanza di container su cui è in esecuzione. È necessario impostare un valore `maxSwap` per il parametro `swappiness` da utilizzare.

`sharedMemorySize`

Valore per le dimensioni (in MiB) del volume `/dev/shm`. Questo parametro è mappato all' `--shm-size` opzione `docker run`.

 Note

Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro `sharedMemorySize` non è supportato.

Tipo: integer

tmpfs

Il percorso del contenitore, le opzioni di montaggio e la dimensione massima (in MiB) del tmpfs montare. Questo parametro è mappato all'--tmpfsopzione docker run.

Note

Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro tmpfs non è supportato.

Tipo: matrice di oggetti [Tmpfs](#)

Campo obbligatorio: no

containerPath

Il percorso assoluto del file in cui tmpfs il volume deve essere montato.

Tipo: stringa

Campo obbligatorio: sì

mountOptions

L'elenco di tmpfs opzioni di montaggio del volume.

Tipo: matrice di stringhe

Campo obbligatorio: no

Valori validi: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

size

La dimensione massima (in MiB) del tmpfs volume.

Tipo: integer

Campo obbligatorio: sì

Dipendenze per i container

dependsOn

Tipo: matrice di oggetti [ContainerDependency](#)

Campo obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un container può contenere più dipendenze. Se una dipendenza è definita per l'avvio del container, per la sua chiusura è invertita. Per un esempio, consulta [Dipendenze per i container](#).

Note

Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Questo parametro richiede che l'attività o il servizio utilizzi la versione della piattaforma 1.3.0 o successiva (Linux) o 1.0.0 (Windows).

```
"dependsOn": [  
  {  
    "containerName": "string",  
    "condition": "string"  
  }  
]
```

containerName

Tipo: stringa

Campo obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

condition

Tipo: stringa

Campo obbligatorio: sì

La condizione di dipendenza del container. Di seguito sono elencate le condizioni disponibili e il loro comportamento:

- **START**: questa condizione emula il comportamento dei collegamenti e dei volumi. La condizione convalida l'avvio di un container dipendente prima di consentire l'avvio di altri container.
- **COMPLETE**: questa condizione verifica l'esecuzione fino al completamento (uscita) di un container dipendente prima di consentire l'avvio di altri container. Può rivelarsi utile per container non essenziali che eseguono uno script e quindi escono. Questa condizione non può essere impostata su un container essenziale.
- **SUCCESS**: questa condizione è uguale a **COMPLETE**, ma richiede anche che il container esca con stato zero. Questa condizione non può essere impostata su un container essenziale.
- **HEALTHY**: questa condizione verifica che il container dipendente superi il controllo dell'integrità del container prima di consentire l'avvio di altri container. Ciò richiede che per il container dipendente siano configurati i controlli dell'integrità nella definizione di attività. Questa condizione viene confermata solo all'avvio dell'attività.

Timeout del container

startTimeout

Tipo: integer

Campo obbligatorio: no

Valori di esempio: 120

Tempo di attesa (in secondi) prima di rinunciare a risolvere le dipendenze per un container.

Ad esempio, vengono specificati due container in una definizione di attività: `containerA` ha una dipendenza sul `containerB` quando raggiunge lo stato **COMPLETE**, **SUCCESS** o **HEALTHY**. Se per `containerB` è specificato un valore `startTimeout` e non raggiunge lo stato desiderato entro tale periodo di tempo, allora `containerA` non viene avviato.

 Note

Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Questo parametro richiede che l'attività o il servizio utilizzi la versione della piattaforma 1.3.0 o successiva (Linux). Il valore massimo è 120 secondi.

stopTimeout

Tipo: integer

Campo obbligatorio: no

Valori di esempio: 120

Durata (in secondi) di attesa prima che sia forzata la chiusura se il container non si arresta da solo normalmente.

Questo parametro richiede che l'attività o il servizio utilizzi la versione della piattaforma 1.3.0 o successiva (Linux). Se il parametro non è specificato, viene utilizzato il valore predefinito di 30 secondi. Il valore massimo è 120 secondi.

Controlli di sistema

systemControls

Tipo: oggetto [SystemControl](#)

Campo obbligatorio: no

Un elenco di parametri del kernel dello spazio dei nomi da impostare nel contenitore. Questo parametro è mappato `Sysctl`s al comando `docker create-container` e all'opzione `docker run`. -- `sysctl` Ad esempio, puoi configurare l'impostazione `net.ipv4.tcp_keepalive_time` per mantenere le connessioni di lunga durata.

Non è consigliabile specificare i parametri `systemControls` correlati alla rete per più container in un'unica attività che utilizza anche la modalità di rete `awsvpc` o `host`. Ciò comporta i seguenti svantaggi:

- Se lo imposti `systemControls` per qualsiasi contenitore, si applica a tutti i contenitori dell'attività. Se hai impostato diversi `systemControls` per più container in un'unica attività, il container che viene avviato per ultimo determina quale `systemControls` diventa effettivo.

Se stai impostando uno spazio dei nomi della risorsa IPC per utilizzare i container nell'attività, ai controlli di sistema si applicano le seguenti condizioni. Per ulteriori informazioni, consulta [Modalità IPC](#).

- Per le attività che utilizzano la modalità host IPC, i valori `systemControls` dello spazio dei nomi IPC non sono supportati.
- Per le attività che utilizzano la modalità IPC task, i valori di `systemControls` dello spazio dei nomi IPC si applicano a tutti i container all'interno di un'attività.

 Note

Questo parametro non è supportato per i container Windows.

 Note

Questo parametro è supportato solo per le attività ospitate su AWS Fargate se le attività utilizzano la versione della piattaforma 1.4.0 o successive (Linux). Non è supportato per i container Windows su Fargate.

```
"systemControls": [  
  {  
    "namespace": "string",  
    "value": "string"  
  }  
]
```

namespace

Tipo: string

Campo obbligatorio: no

Il parametro del kernel dello spazio dei nomi per cui impostare un. value

Valori dello spazio dei nomi IPC validi: "kernel.msgmax" | "kernel.msgmnb" | "kernel.msgmni" | "kernel.sem" | "kernel.shmall" | "kernel.shmmax" | "kernel.shmmni" | "kernel.shm_rmid_forced" e Sysctls che iniziano con "fs.mqueue.*"

Valori validi dello spazio dei nomi di rete: che iniziano con. Sysctls "net.*" In Fargate, vengono accettati solo i namespace Sysctls esistenti all'interno del contenitore.

Tutti questi valori sono supportati da Fargate.

value

Tipo: string

Campo obbligatorio: no

Il valore per il parametro del kernel dello spazio dei nomi specificato in. namespace

Interactive

interactive

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è true, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di stdin o tty. Questo parametro è mappato OpenStdin al comando docker create-container e --interactive all'opzione docker run.

Il valore predefinito è false.

Pseudoterminale

pseudoTerminal

Tipo: Booleano

Campo obbligatorio: no

Quando il parametro è true, è allocato un TTY. Questo parametro è mappato Tty al comando docker create-container e --tty all'opzione docker run.

Il valore predefinito è `false`.

Nome dell'acceleratore di inferenza elastica

Il requisito della risorsa dell'acceleratore di Elastic Inference per la definizione di attività.

Note

Amazon Elastic Inference (EI) non è più disponibile per i clienti.

I parametri seguenti sono permessi nella definizione di un processo:

`deviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del dispositivo dell'acceleratore di inferenza elastica. Il `deviceName` deve anche essere riferito in una definizione del container; consulta [Elastic Inference accelerator](#).

`deviceType`

Tipo: stringa

Campo obbligatorio: sì

L'acceleratore di inferenza elastica da utilizzare.

Configurazione del proxy

`proxyConfiguration`

Tipo: oggetto [ProxyConfiguration](#)

Campo obbligatorio: no

I dettagli di configurazione del proxy App Mesh.

 Note

Questo parametro non è supportato per i container Windows.

```
"proxyConfiguration": {  
  "type": "APPMESH",  
  "containerName": "string",  
  "properties": [  
    {  
      "name": "string",  
      "value": "string"  
    }  
  ]  
}
```

type

Tipo: stringa

Valori del valore: APPMESH

Campo obbligatorio: no

Il tipo di proxy. L'unico valore supportato è APPMESH.

containerName

Tipo: stringa

Campo obbligatorio: sì

Il nome del container che agirà come proxy App Mesh.

properties

Tipo: matrice di oggetti [KeyValuePair](#)

Campo obbligatorio: no

Il set di parametri di configurazione di rete per il plug-in Container Network Interface (CNI), specificati come coppie chiave-valore.

- **IgnoredUID**: (obbligatorio) l'ID utente (UID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato **IgnoredGID**, questo campo può rimanere vuoto.
- **IgnoredGID**: (obbligatorio) l'ID gruppo (GID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato **IgnoredUID**, questo campo può rimanere vuoto.
- **AppPorts** (obbligatorio): l'elenco delle porte utilizzate dall'applicazione. Il traffico di rete per queste porte viene inoltrato alle porte **ProxyIngressPort** e **ProxyEgressPort**.
- **ProxyIngressPort** (obbligatorio): specifica la porta a cui è diretto il traffico in entrata per le **AppPorts**.
- **ProxyEgressPort** (obbligatorio): specifica la porta a cui è diretto il traffico in uscita da **AppPorts**.
- **EgressIgnoredPorts**: (obbligatorio) il traffico in uscita diretto verso le porte specificate viene ignorato e non reindirizzato alla **ProxyEgressPort**. Può essere un elenco vuoto.
- **EgressIgnoredIPs**: (obbligatorio) il traffico in uscita diretto verso gli indirizzi IP specificati viene ignorato e non reindirizzato alla **ProxyEgressPort**. Può essere un elenco vuoto.

`name`

Tipo: string

Campo obbligatorio: no

Nome della coppia chiave-valore.

`value`

Tipo: string

Campo obbligatorio: no

Valore della coppia chiave-valore.

Volumi

Quando registri una definizione di attività, puoi facoltativamente specificare un elenco di volumi da passare al Docker demone su un'istanza di contenitore, che diventa quindi disponibile per l'accesso da parte di altri contenitori sulla stessa istanza di contenitore.

Di seguito sono elencati i tipi di volumi di dati che è possibile utilizzare:

- Volumi Amazon EBS: fornisce storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Puoi collegare 1 volume Amazon EBS per attività Amazon ECS durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. I volumi Amazon EBS sono supportati per le attività Linux ospitate su Fargate. Per ulteriori informazioni, consulta [Usa i volumi Amazon EBS con Amazon ECS](#).
- Volumi Amazon EFS: offre uno spazio di archiviazione di file semplice, scalabile e persistente da utilizzare con i processi Amazon ECS. Con Amazon EFS, la capacità di storage è elastica. La capacità di storage aumenta e si riduce automaticamente quando si aggiungono e si rimuovono i file. Le tue applicazioni possono disporre dello storage di cui hanno bisogno e quando ne hanno bisogno. I volumi Amazon EFS sono supportati per le attività ospitate su Fargate. Per ulteriori informazioni, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).
- FSx per volumi Windows File Server: fornisce file server Microsoft Windows completamente gestiti. Questi server di file sono supportati da un file system Windows. Quando si utilizza FSx Windows File Server insieme ad Amazon ECS, è possibile effettuare il provisioning delle attività Windows con uno storage di file persistente, distribuito, condiviso e statico. Per ulteriori informazioni, consulta [Utilizzo FSx per volumi Windows File Server con Amazon ECS](#).

Questa opzione non è supportata per i container Windows su Fargate.

- Bind mounts: un file o una directory sulla macchina host montata in un contenitore. I volumi host Bind Mount sono supportati durante l'esecuzione delle attività. Per usare i volumi host di montaggio vincolati, specifica un valore `host` e un valore opzionale `sourcePath` nella definizione di attività.

Per ulteriori informazioni, consulta [Opzioni di storage per le attività di Amazon ECS](#).

I seguenti parametri sono consentiti in una definizione del container.

`name`

Tipo: string

Campo obbligatorio: no

Nome del volume. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e caratteri di sottolineatura (-). _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume mountPoints`

`host`

Campo obbligatorio: no

Il `host` parametro viene utilizzato per collegare il ciclo di vita del `bind mount` all' EC2 istanza Amazon host, anziché all'attività e alla posizione in cui è archiviata. Se il parametro `host` è vuoto, il daemon Docker assegna un percorso host per il tuo volume di dati, ma non è garantito che i dati vengano mantenuti dopo che viene interrotta l'esecuzione del container a essi associato.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`.

 Note

Il `sourcePath` parametro è supportato solo quando si utilizzano attività ospitate su EC2 istanze Amazon.

`sourcePath`

Tipo: string

Campo obbligatorio: no

Quando viene utilizzato il `host` parametro, specifica `sourcePath` a per dichiarare il percorso sull' EC2 istanza Amazon host che viene presentata al contenitore. Se questo parametro è vuoto, il daemon Docker assegna automaticamente un percorso host. Se il `host` parametro contiene una posizione di `sourcePath` file, il volume di dati persiste nella posizione specificata sull' EC2 istanza Amazon host finché non lo elimini manualmente. Se il `sourcePath` valore non esiste sull' EC2 istanza Amazon host, il demone Docker lo crea. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

`configuredAtLaunch`

Tipo: Booleano

Campo obbligatorio: no

Specifica se un volume è configurabile all'avvio. Se impostato su `true`, è possibile configurare il volume durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. Se impostato su `true`, non sarà possibile fornire un'altra configurazione del volume nella definizione dell'attività. Questo parametro deve essere impostato per `true` configurare un volume Amazon EBS da allegare a un'attività. L'impostazione `configuredAtLaunch` `true` e il rinvio della configurazione del volume alla fase di avvio consentono di creare definizioni di attività che non sono limitate a un tipo di volume o a impostazioni di volume specifiche. In questo modo la

definizione delle attività è riutilizzabile in diversi ambienti di esecuzione. Per ulteriori informazioni consulta [Amazon EBS volumes](#).

`dockerVolumeConfiguration`

Tipo: oggetto [DockerVolumeConfiguration](#)

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker. I volumi Docker sono supportati solo quando si eseguono attività su EC2 istanze. I contenitori Windows supportano solo l'uso del `local` driver. Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

Tipo: stringa

Valori validi: `task` | `shared`

Campo obbligatorio: no

L'ambito del volume Docker che determina il suo ciclo di vita. I volumi Docker che rientrano nell'ambito `task` vengono automaticamente assegnati all'avvio del processo e distrutti quando il processo viene arrestato. I volumi Docker che vengono definiti come `shared` vengono mantenuti dopo l'arresto del processo.

`autoprovision`

Tipo: Booleano

Valore predefinito: `false`

Campo obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo viene utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omesso.

`driver`

Tipo: string

Campo obbligatorio: no

Il driver del volume Docker da utilizzare. Il valore del driver deve corrispondere al nome del driver fornito da Docker perché questo nome viene utilizzato per il posizionamento delle

attività. Se il driver è stato installato utilizzando la CLI del plug-in Docker, `docker plugin ls` utilizzalo per recuperare il nome del driver dall'istanza del contenitore. Se il driver è stato installato utilizzando un altro metodo, utilizza Docker Plugin Discovery per recuperare il nome del driver.

`driverOpts`

Tipo: string

Campo obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da esaminare. Questo parametro viene mappato `DriverOpts` alla sezione Crea un volume di Docker.

`labels`

Tipo: string

Campo obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`efsVolumeConfiguration`

Tipo: oggetto [EFSVolumedi configurazione](#)

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

Tipo: string

Campo obbligatorio: no

La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omissso, verrà utilizzata la radice del volume Amazon EFS. La specifica di `/` avrà lo stesso effetto dell'omissione di questo parametro.

 Important

Se un punto di accesso EFS è specificato in `authorizationConfig`, il parametro della directory principale deve essere omesso o impostato su `/`, il che applicherà il percorso impostato sul punto di accesso EFS.

`transitEncryption`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se abilitare o meno la crittografia per i dati Amazon EFS in transito tra l'host Amazon ECS e il server Amazon EFS. Se si utilizza l'autorizzazione IAM di Amazon EFS, è necessario abilitare la crittografia di transito. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni, consulta [Crittografia dei dati in transito](#) nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: integer

Campo obbligatorio: no

La porta da utilizzare per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non specifichi una porta di crittografia di transito, l'attività utilizzerà la strategia di selezione delle porte utilizzata dall'helper di montaggio di Amazon EFS. Per ulteriori informazioni, consulta [Assistente per il montaggio di EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: oggetto [EFSAuthorizationConfig](#)

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

Tipo: string

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory principale in `efsVolumeConfiguration` deve essere omesso o impostato su `/`, il che applicherà il percorso impostato sul punto di accesso EFS. Se si utilizza un punto di accesso, la crittografia di transito deve essere abilitata in `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Amazon EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`iam`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se utilizzare il ruolo IAM dell'attività Amazon ECS definito in una definizione di attività durante il montaggio del file system Amazon EFS. Se abilitato, la crittografia di transito deve essere abilitata nella casella `EFSVolumeConfiguration`. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni consulta [Ruoli IAM per le attività](#).

`FSxWindowsFileServerVolumeConfiguration`

Tipo: oggetto [FSxWindowsFileServerVolumeConfiguration](#)

Campo obbligatorio: sì

Questo parametro viene specificato quando utilizzi un file system [Amazon FSx for Windows File Server](#) per l'archiviazione delle attività.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

Campo obbligatorio: sì

La directory all'interno del FSx file system Windows File Server da montare come directory principale all'interno dell'host.

`authorizationConfig`

`credentialsParameter`

Tipo: stringa

Campo obbligatorio: sì

Le opzioni delle credenziali di autorizzazione.

opzioni:

- Amazon Resource Name (ARN) di un [AWS Secrets Manager](#)segreto.
- ARN di un [AWS Systems Manager](#)parametro.

`domain`

Tipo: stringa

Campo obbligatorio: sì

Un nome di dominio completo ospitato da una directory [AWS Directory Service for Microsoft Active Directory](#)(AWS Managed Microsoft AD) o da un Active Directory ospitato autonomamente EC2 .

Tag

Quando registri una definizione di attività, puoi facoltativamente specificare tag di metadati applicati alla definizione di attività. I tag consentono di suddividere in categorie e organizzare la definizione di attività. Ciascun tag è formato da una chiave e da un valore facoltativo. Li definisci entrambi. Per ulteriori informazioni, consulta [Taggare le risorse Amazon ECS](#).

Important

Non aggiungere Informazioni personali di identificazione o altre informazioni riservate o sensibili nei tag. I tag sono accessibili a molti AWS servizi, inclusa la fatturazione. I tag non sono destinati a essere utilizzati per dati privati o sensibili.

I seguenti parametri sono consentiti in un oggetto di tag.

key

Tipo: string

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

value

Tipo: string

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

Altri parametri di definizione di attività

I seguenti parametri di definizione di attività possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS con l'opzione Configure via JSON (Configura tramite JSON). Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Argomenti

- [Archiviazione temporanea](#)
- [Modalità IPC](#)
- [Modalità PID](#)
- [Iniezione in errore](#)

Archiviazione temporanea

ephemeralStorage

Tipo: oggetto [EphemeralStorage](#)

Campo obbligatorio: no

La quantità di archiviazione temporanea in GB da allocare per l'attività. Questo parametro viene utilizzato per espandere la quantità totale di archiviazione temporanea disponibile, oltre l'importo predefinito, per le attività ospitate su AWS Fargate. Per ulteriori informazioni, consulta [the section called “Montaggi vincolati”](#).

 Note

Questo parametro è supportato solo nella versione della piattaforma 1.4.0 o successiva (Linux) 1.0.0 o successiva (Windows).

Modalità IPC

`ipcMode`

Questa funzionalità non è supportata per le attività in esecuzione su Fargate.

Tipo: string

Campo obbligatorio: no

Lo spazio dei nomi della risorsa IPC da utilizzare per i container nell'attività. I valori validi sono `host`, `task` o `none`. Se `host` specificato, tutti i contenitori che rientrano nelle attività che hanno specificato la modalità `host` IPC sulla stessa istanza di contenitore condividono le stesse risorse IPC con l'istanza Amazon EC2 host. Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono le stesse risorse IPC. Se è stato specificato `none`, le risorse IPC all'interno dei container di un'attività sono private e non condivise con altri container in un'attività o sull'istanza di container. Se non è stato specificato alcun valore, la condivisione dello spazio dei nomi della risorsa IPC dipende dalle impostazioni del daemon Docker sull'istanza del container.

Se viene utilizzata la modalità `host` IPC, tieni presente che esiste un maggiore rischio di esposizione a spazi dei nomi IPC indesiderati.

Se stai impostando parametri kernel associati a uno spazio dei nomi utilizzando `systemControls` per i container nell'attività, allo spazio dei nomi della risorsa IPC si applica quanto segue.

- Per le attività che utilizzano la modalità `host IPC`, spazi dei nomi IPC correlati a `systemControls` non sono supportati.
- Per le attività che utilizzano la modalità `task IPC`, `systemControls` correlati allo spazio dei nomi IPC vengono applicati a tutti i container all'interno di un'attività.

Note

Questo parametro non è supportato per i container o le attività Windows che utilizzano il tipo di avvio Fargate.

Modalità PID

`pidMode`

Tipo: stringa

Valori validi: `host` | `task`

Campo obbligatorio: no

Lo spazio dei nomi del processo da utilizzare per i container nell'attività. I valori validi sono `host` o `task`. Nei contenitori Linux, l'unico valore valido è `task`. Ad esempio, i sidecar di monitoraggio potrebbero aver bisogno di `pidMode` per accedere a informazioni su altri container in esecuzione nella stessa attività.

Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono lo stesso spazio dei nomi del processo.

Se non è stato specificato alcun valore, l'impostazione predefinita è uno spazio dei nomi privato per ogni container.

Note

Questo parametro è supportato solo per le attività ospitate su AWS Fargate se le attività utilizzano la versione della piattaforma `1.4.0` o successive (Linux). Non è supportato per i container Windows su Fargate.

Iniezione in errore

`enableFaultInjection`

Tipo: Booleano

Valori validi: `true` | `false`

Campo obbligatorio: no

Se questo parametro è impostato su `true`, nel payload di un'attività, Amazon ECS e Fargate accettano richieste di inserimento di errori dai contenitori dell'attività. Questo parametro è impostato su `false` per impostazione predefinita.

Parametri di definizione delle attività di Amazon ECS per il tipo di EC2 avvio

Le definizioni delle attività sono suddivise in parti separate: la famiglia di attività, il ruolo dell'attività AWS Identity and Access Management (IAM), la modalità di rete, le definizioni dei contenitori, i volumi, i vincoli di posizionamento delle attività e i tipi di avvio. Le definizioni della famiglia e del container sono richieste in una definizione di attività. Al contrario, il ruolo dell'attività, la modalità di rete, i volumi, i vincoli di posizionamento delle attività e il tipo di avvio sono facoltativi.

Puoi utilizzare questi parametri in un file JSON per configurare la definizione del processo.

Di seguito sono riportate descrizioni più dettagliate per ogni parametro di definizione delle attività per il tipo di EC2 avvio.

Family

`family`

Tipo: stringa

Campo obbligatorio: sì

Quando registri una definizione di attività, le assegni una famiglia, ovvero una sorta di nome per più versioni della definizione di attività, specificando un numero di revisione. Alla prima definizione

di attività registrata in una determinata famiglia viene assegnato il numero di revisione 1 e a qualsiasi definizione di attività registrata successivamente viene assegnato un numero di revisione sequenziale.

Tipi di avvio

Quando registri una definizione di attività, puoi specificare un tipo di avvio che Amazon ECS deve convalidare per la definizione di attività. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client. Per ulteriori informazioni, consulta [Tipi di avvio di Amazon ECS](#).

Il parametro seguente è permesso in una definizione di attività.

`requiresCompatibilities`

Tipo: array di stringhe

Campo obbligatorio: no

Valori validi: EC2

Il tipo di avvio per il quale è stata convalidata la definizione di attività. In questo modo viene avviato un controllo per garantire che tutti i parametri utilizzati nella definizione di attività soddisfino i requisiti del tipo di avvio.

Ruolo del processo

`taskRoleArn`

Tipo: string

Campo obbligatorio: no

Quando registri una definizione di task, puoi fornire un ruolo di task per un ruolo IAM che consenta ai contenitori inclusi nell'operazione di chiamare per tuo conto i dati specificati nelle policy associate. AWS APIs Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

All'avvio dell'AMI Windows Server ottimizzata per Amazon ECS, i ruoli IAM per le attività su Windows richiedono che l'opzione `-EnableTaskIAMRole` sia impostata. I container devono

anche eseguire un codice di configurazione per utilizzare la funzionalità. Per ulteriori informazioni, consulta [Configurazione aggiuntiva dell'istanza Amazon EC2 Windows](#).

Ruolo per l'esecuzione del processo

`executionRoleArn`

Tipo: stringa

Obbligatorio: condizionale

L'Amazon Resource Name (ARN) del ruolo di esecuzione dell'attività che concede all'agente container Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto.

Note

Il ruolo IAM di esecuzione del processo è richiesto in base ai requisiti del processo. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Modalità di rete

`networkMode`

Tipo: string

Campo obbligatorio: no

La modalità di rete Docker da utilizzare per i container nel processo. Per le attività di Amazon ECS ospitate su istanze Amazon EC2 Linux, i valori validi sono `none`, `bridgeawsvpc`, e `host`. Se non viene specificata alcuna modalità di rete, la modalità di rete predefinita è `bridge`. Per le attività di Amazon ECS ospitate su istanze Amazon EC2 Windows, i valori validi sono `default`, e `awsvpc`. Se non viene specificata alcuna modalità di rete, viene utilizzata la modalità `default` di rete.

Se la modalità di rete è impostata su `none`, i container dell'attività non dispongono di connettività esterna e non è possibile specificare le mappature delle porte nella definizione del container.

Se la modalità di rete è attivata su `bridge`, l'attività utilizza la rete virtuale integrata di Docker su Linux, che viene eseguita all'interno di ogni EC2 istanza Amazon che ospita l'attività. La rete virtuale integrata su Linux utilizza il driver di rete `bridge` Docker.

Se la modalità di rete è `host`, l'attività utilizza la rete dell'host che aggira la rete virtuale integrata di Docker mappando le porte dei container direttamente all'ENI dell' EC2 istanza Amazon che ospita l'attività. Le mappature dinamiche delle porte non possono essere utilizzate in questa modalità di rete. Un container in una definizione di attività che utilizza questa modalità deve specificare un numero di `hostPort` specifico. Un numero di porta su un host non può essere utilizzato da più attività. Di conseguenza, non puoi eseguire più attività con la stessa definizione di attività su una singola EC2 istanza Amazon.

 Important

Quando si eseguono attività che utilizzano la modalità di rete `host`, per una maggiore sicurezza, non eseguire container utilizzando l'utente `root` (UID 0). Come best practice per la sicurezza, utilizza sempre un utente non `root`.

Se la modalità di rete è `awsvpc`, al processo viene assegnata un'interfaccia di rete elastica e, quando crei un servizio o esegui un processo con la definizione di attività, devi specificare `NetworkConfiguration`. Per ulteriori informazioni, consulta [Opzioni di task networking di Amazon ECS per il tipo di EC2 avvio](#).

Se la modalità di rete è `awsvpc`, l'attività utilizza la rete virtuale integrata di Docker su Windows, che viene eseguita all'interno di ogni EC2 istanza Amazon che ospita l'attività. La rete virtuale integrata su Windows utilizza il driver di rete `nat` Docker.

Le modalità di `awsvpc` rete `host` e offrono le massime prestazioni di rete per i container perché utilizzano lo stack EC2 di rete Amazon. Con le modalità di rete `host` e `awsvpc`, le porte dei container esposte sono mappate direttamente alla porta host corrispondente (per la modalità di rete `host`) o alla porta dell'interfaccia di rete elastica collegata (per la modalità di rete `awsvpc`). Pertanto, non puoi utilizzare le mappature delle porte host dinamiche.

La modalità di rete consentita dipende dal sistema operativo dell' EC2 istanza sottostante. Con Linux, può essere utilizzata qualsiasi modalità di rete. Se Windows, possono essere utilizzate le modalità `default` e `awsvpc`.

Piattaforma di runtime

`operatingSystemFamily`

Tipo: stringa

Obbligatorio: condizionale

Di default: LINUX

Quando registri una definizione di attività, devi specificare la famiglia del sistema operativo.

I valori validi sono

LINUXWINDOWS_SERVER_2022_CORE,WINDOWS_SERVER_2022_FULL,WINDOWS_SERVER_2019_FULL,
eWINDOWS_SERVER_2019_CORE,
WINDOWS_SERVER_2016_FULLWINDOWS_SERVER_2004_CORE,
eWINDOWS_SERVER_20H2_CORE.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Quando una definizione di attività fa parte di un servizio, questo valore deve corrispondere al valore `platformFamily` del servizio.

cpuArchitecture

Tipo: stringa

Obbligatorio: condizionale

Valore di default: X86_64

Quando registri una definizione di attività, devi specificare l'architettura della CPU. I valori validi sono X86_64 e ARM64.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Quando sono presenti attività Linux, è possibile impostare il valore suARM64. Per ulteriori informazioni, consulta [the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”](#).

Dimensioni processo

Quando registri una definizione di attività, puoi specificare la quantità totale di CPU e memoria utilizzata per l'attività. Questo valore è separato dai valori `cpu` e `memory` a livello di definizione del container. Per le attività ospitate su EC2 istanze Amazon, questi campi sono facoltativi.

Note

I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows. Ti consigliamo di specificare risorse a livello di container per i container Windows.

cpu

Tipo: stringa

Obbligatorio: condizionale

Note

Questo parametro non è supportato per i container Windows.

Il limite rigido di unità CPU da presentare per il processo. Puoi specificare i valori della CPU nel file JSON come stringa in unità CPU o in modalità virtuale CPUs (vCPUs). Ad esempio, è possibile specificare un valore di CPU come 1024 in unità CPU o 1 vCPU in vCPUs. Quando la definizione dell'attività viene registrata, un valore vCPU viene convertito in un numero intero che indica le unità CPU.

Questo campo è facoltativo. Se il cluster non dispone di istanze di container registrate con le unità CPU richieste disponibili, l'attività non va a buon fine. I valori supportati sono compresi tra 0.125 v CPUs e 192 v. CPUs

memory

Tipo: stringa

Obbligatorio: condizionale

Note

Questo parametro non è supportato per i container Windows.

Il limite massimo di memoria da presentare all'attività. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibyte (MiB) o gigabyte (GB). Ad esempio, è

possibile specificare un valore di memoria 3072 in MiB o 3 GB in GB. Quando la definizione di attività è registrata, un valore GB viene convertito in un numero intero che indica il MiB.

Questo campo è facoltativo e può essere utilizzato qualsiasi valore. Se viene specificato un valore di memoria a livello di attività, il valore di memoria a livello di container è facoltativo. Se il cluster non dispone di istanze di container registrate con la memoria richiesta disponibile, l'attività non va a buon fine. Puoi ottimizzare l'utilizzo delle risorse assegnando alle attività quanta più memoria possibile per un determinato tipo di istanza. Per ulteriori informazioni, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

Definizioni del container

Quando si registra una definizione di attività, è necessario specificare un elenco di definizioni di contenitore che vengono passate a Docker demone su un'istanza del contenitore. I seguenti parametri sono consentiti in una definizione del container.

Argomenti

- [Parametri standard di definizione del container](#)
- [Parametri avanzati di definizione del container](#)
- [Altri parametri di definizione del container](#)

Parametri standard di definizione del container

I seguenti parametri di definizione di attività sono obbligatori o utilizzati nella maggior parte delle definizioni del container.

Argomenti

- [Nome](#)
- [Immagine](#)
- [Memoria](#)
- [Mappature di porte](#)
- [Credenziali del repository privato](#)

Nome

name

Tipo: stringa

Campo obbligatorio: sì

Il nome di un container. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. Se colleghi più container in una definizione di attività, il parametro name di uno dei container può essere inserito nel parametro links di un altro container. Questo per collegare i container.

Immagine

image

Tipo: stringa

Campo obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene passata direttamente al Docker demone. Per impostazione predefinita, le immagini in Docker Hub il registro è disponibile. Puoi anche specificare altri repository con *repository-url/image:tag* o *repository-url/image@digest*. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole); sono consentiti numeri, trattini, caratteri di sottolineatura, due punti, punti, barre e cancelletti. Questo parametro è Image mappato al comando docker create-container e al parametro IMAGE del comando docker run.

- Quando viene avviato un nuovo processo, l'agente del container Amazon ECS esegue il pull della versione più recente dell'immagine e del tag specificati per il container da utilizzare. Tuttavia, gli aggiornamenti successivi a un'immagine del repository non vengono propagate alle attività già in esecuzione.
- Quando non specifichi un tag o un digest nel percorso dell'immagine nella definizione dell'attività, l'agente container Amazon ECS recupera la versione più recente dell'immagine specificata.
- Tuttavia, gli aggiornamenti successivi a un'immagine del repository non vengono propagate alle attività già in esecuzione.
- Le immagini nei registri privati sono supportate. Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

- Le immagini nei repository Amazon ECR possono essere specificate utilizzando la convenzione di denominazione `registry/repository:tag` o `registry/repository@digest`, ad esempio, `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app:latest` o `aws_account_id.dkr.ecr.region.amazonaws.com/my-web-app@sha256:94afd1f2e64d908bc90dbca0035a5b567EXAMPLE`.
- Immagini nei repository ufficiali su Docker Hub usa un solo nome (ad esempio, `ubuntu` o `mongo`).
- Immagini in altri repository su Docker Hub sono qualificati con il nome di un'organizzazione (ad esempio, `amazon/amazon-ecs-agent`).
- Le immagini in altri archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

versionConsistency

Tipo: stringa

Valori validi: `enabled|disabled`

Campo obbligatorio: no

Specifica se Amazon ECS risolverà il tag dell'immagine del contenitore fornito nella definizione del contenitore in un digest di immagini. Per impostazione predefinita, questo comportamento è `enabled`. Se imposti il valore per un contenitore come `disabled`, Amazon ECS non risolverà il tag dell'immagine del contenitore in un digest e utilizzerà l'URI dell'immagine originale specificato nella definizione del contenitore per la distribuzione. Per ulteriori informazioni sulla risoluzione dell'immagine del contenitore, consulta [Risoluzione dell'immagine del contenitore](#)

Memoria

memory

Tipo: integer

Campo obbligatorio: no

La quantità (in MiB) della memoria da presentare al container. Se il container tenta di superare la memoria specificata qui, viene terminato. La quantità totale di memoria prenotata per tutti i container all'interno di un processo deve essere inferiore al valore `memory` del processo, se specificato. Questo parametro è mappato `Memory` al comando `docker create-container` e `--memory` all'opzione `docker run`.

È necessario specificare un valore di memoria a livello di attività o un valore di memoria a livello di contenitore. Se specifichi sia un valore di `memory` a livello di container che un valore di `memoryReservation`, `memory` deve essere maggiore del valore di `memoryReservation`. Se specifichi `memoryReservation`, tale valore viene sottratto dalle risorse di memoria disponibili per l'istanza di container in cui il container viene posizionato. In caso contrario, viene utilizzato il valore `memory`.

Il daemon Docker 20.10.0 o versione successiva prenota un minimo di 6 MiB di memoria per un container. Pertanto, non specificare meno di 6 MiB di memoria per i container.

Il daemon Docker 19.03.13-ce o versione precedente prenota un minimo di 4 MiB di memoria per un container. Pertanto, non specificare meno di 4 MiB di memoria per i container.

Note

Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

`memoryReservation`

Tipo: integer

Campo obbligatorio: no

Il limite flessibile (in MiB) della memoria da prenotare per il container. Quando la memoria di sistema è in conflitto, Docker tenta di mantenere la memoria del contenitore entro questo limite flessibile. Tuttavia, il container può utilizzare una quantità maggiore di memoria, se necessario. Il container può utilizzare memoria fino al limite rigido specificato con il parametro `memory` (se applicabile) o tutta la memoria disponibile sull'istanza di container, a seconda di quale evento si verifica prima. Questo parametro è mappato `MemoryReservation` al comando `docker create-container` e `--memory-reservation` all'opzione `docker run`.

Se non viene specificato un valore di memoria a livello di attività, è necessario specificare un numero intero diverso da zero per uno o entrambi i codici `memory` o `memoryReservation` in una definizione del container. Se specifichi entrambe, `memory` deve essere superiore a `memoryReservation`. Se specifichi `memoryReservation`, tale valore viene sottratto dalle risorse di memoria disponibili per l'istanza di container in cui il container viene posizionato. In caso contrario, viene utilizzato il valore `memory`.

Ad esempio, supponiamo che il container normalmente utilizza 128 MiB di memoria, ma con picchi occasionali di 256 MiB di memoria per brevi periodi di tempo. Puoi impostare un valore di `memoryReservation` di 128 MiB e un limite rigido `memory` di 300 MiB. Questa configurazione consente al container di riservare 128 MiB di memoria solo dalle risorse restanti nell'istanza di container. Allo stesso tempo, questa configurazione consente al container di utilizzare più risorse di memoria quando necessario.

 Note

Questo parametro non è supportato per i container Windows.

Il daemon Docker 20.10.0 o versione successiva prenota un minimo di 6 MiB di memoria per un container. Pertanto, non specificare meno di 6 MiB di memoria per i container.

Il daemon Docker 19.03.13-ce o versione precedente prenota un minimo di 4 MiB di memoria per un container. Pertanto, non specificare meno di 4 MiB di memoria per i container.

 Note

Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

Mappature di porte

`portMappings`

Tipo: array di oggetti

Campo obbligatorio: no

Le mappature delle porte espongono le porte di rete del contenitore al mondo esterno. Ciò consente ai client di accedere all'applicazione. Viene anche utilizzato per la comunicazione tra container nell'ambito della stessa attività.

Per le definizioni di attività che utilizzano la modalità di rete `awsvpc`, specifica solo il parametro `containerPort`. `hostPort` viene sempre ignorato e la porta del contenitore viene automaticamente mappata su una porta casuale con numero alto sull'host.

La mappatura delle porte su Windows usa l'indirizzo gateway NetNAT anziché localhost. Non vi è alcun loopback per le mappature delle porte su Windows, perciò non è possibile accedere alla porta mappata di un container dall'host stesso.

La maggior parte dei campi di questo parametro (incluso `containerPort`, `hostPort`, `protocol`) è mappata `PortBindings` al comando `docker create-container` e `--publish` all'opzione `docker run`. Se la modalità di rete di una definizione di attività è impostata su `host`, le porte `host` devono essere non definite o devono corrispondere alla porta del container nella mappatura della porta.

Note

Dopo che un processo raggiunge lo stato `RUNNING`, gli incarichi manuali e automatici relativi alle porte del container e dell'host sono visibili nelle posizioni seguenti:

- Console: sezione `Binding` di rete della descrizione di un container per un processo selezionato.
- AWS CLI: la sezione `networkBindings` dell'output del comando `describe-tasks`.
- API: risposta `DescribeTasks`.
- Metadati: l'endpoint dei metadati dell'attività.

`appProtocol`

Tipo: `string`

Campo obbligatorio: no

Il protocollo dell'applicazione utilizzato per la mappatura delle porte. Questo parametro si applica solo a `Service Connect`. Ti consigliamo di impostare questo parametro in maniera coerente con il protocollo utilizzato dall'applicazione. Se imposti questo parametro, Amazon ECS aggiunge la gestione delle connessioni specifica del protocollo al proxy `Service Connect`. Se imposti questo parametro, Amazon ECS aggiunge la telemetria specifica del protocollo nella console Amazon ECS e `CloudWatch`.

Se non imposti un valore per questo parametro, viene utilizzato `TCP`. Amazon ECS, tuttavia, non aggiunge la telemetria specifica del protocollo `TCP`.

Per ulteriori informazioni, consulta [the section called “Service Connect”](#).

Valori di protocollo validi: `"http"` | `"http2"` | `"grpc"`

containerPort

Tipo: integer

Obbligatorio: sì, quando si utilizzano portMappings

Il numero di porta nel container associato alla porta dell'host definito dall'utente o assegnata automaticamente.

Per le attività che utilizzano la modalità di awsvpc rete, è necessario specificare le porte esposte. containerPort

Supponiamo di utilizzare contenitori in un'attività con il tipo di EC2 avvio e di specificare una porta contenitore e non una porta host. Il container riceve quindi automaticamente una porta dell'host nell'intervallo delle porte temporanee. Per ulteriori informazioni, consulta hostPort. La mappatura delle porte che sono assegnate automaticamente in questo modo non contano ai fini della quota di 100 porte riservate di un'istanza di container.

containerPortRange

Tipo: string

Campo obbligatorio: no

L'intervallo dei numeri di porta nel container associato all'intervallo di porte host mappato in maniera dinamica.

È possibile impostare questo parametro solo utilizzando l'API `register-task-definition`. L'opzione è disponibile nel parametro `portMappings`. Per ulteriori informazioni, consulta [register-task-definition](#) nella documentazione di riferimento AWS Command Line Interface .

Quando specifichi un `containerPortRange`, si applicano le seguenti regole:

- Devi utilizzare la modalità di rete `bridge` o la modalità di rete `awsvpc`.
- Questo parametro è disponibile per sistemi operativi sia Linux che Windows.
- L'istanza di container deve avere almeno la versione 1.67.0 dell'agente del container e almeno la versione 1.67.0-1 del pacchetto `ecs-init`.
- Puoi specificare fino a 100 intervalli di porte per container.
- Non specificare un `hostPortRange`. Il valore dell'`hostPortRange` è impostato come indicato di seguito:
 - Per i container in un'attività con la modalità di rete `awsvpc`, la `hostPort` è impostata sullo stesso valore della `containerPort`. Questa è una strategia di mappatura statica.

- Per i container in un'attività con la modalità di rete bridge, l'agente Amazon ECS trova le porte host aperte dall'intervallo effimero predefinito e lo passa a docker per associarle alle porte del container.
- I valori validi di `containerPortRange` sono compresi tra 1 e 65535.
- Una porta può essere inclusa solo in una sola mappatura delle porte per ogni container.
- Non puoi specificare intervalli di porte sovrapposti.
- La prima porta nell'intervallo deve essere minore dell'ultima porta nell'intervallo.
- Docker consiglia di disattivare il docker-proxy nel Docker file di configurazione del demone quando si dispone di un numero elevato di porte.

Per ulteriori informazioni, consultate il [numero #11185 su](#). GitHub

Per informazioni su come disattivare il docker-proxy nel Docker file di configurazione del demone, vedere [Docker daemon](#) nella Amazon ECS Developer Guide.

Puoi effettuare la chiamata a [DescribeTasks](#) per visualizzare l'`hostPortRange`, cioè le porte dell'host associate alle porte del container.

Gli intervalli di porte non sono inclusi negli eventi delle attività di Amazon ECS, a EventBridge cui vengono inviati. Per ulteriori informazioni, consulta [the section called “Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge”](#).

hostPortRange

Tipo: string

Campo obbligatorio: no

L'intervallo di numeri di porta sull'host utilizzato con il collegamento di rete. Questo viene assegnato da Docker e fornito dall'agente Amazon ECS.

hostPort

Tipo: integer

Campo obbligatorio: no

Il numero di porta nell'istanza di container per prenotare per il container.

Puoi specificare una porta dell'host non riservata per la mappatura delle porte del container. Questa operazione viene definita mappatura statica delle porte dell'host. In alternativa, puoi omettere `hostPort` (o impostarlo su 0) specificando `containerPort`. Il container

riceve automaticamente una porta nell'intervallo di porte temporanee per il sistema operativo dell'istanza di container e Docker versione. Questa operazione viene definita mappatura dinamica delle porte dell'host.

L'intervallo di porte effimere predefinito Docker la versione 1.6.0 e successive è elencata nell'istanza sotto. `/proc/sys/net/ipv4/ip_local_port_range` Se questo parametro kernel non è disponibile, viene utilizzato l'intervallo delle porte temporaneo di default da 49153–65535. Non tentare di specificare una porta dell'host nell'intervallo di porte effimere. Questo perché sono riservate per l'assegnazione automatica. In generale, le porte al di sotto di 32768 non rientrano nell'intervallo delle porte temporanee.

Le porte riservate predefinite sono 22 per SSH, le Docker porte 2375 e 2376 porte Amazon ECS Container Agent. 51678-51680 Qualsiasi porta dell'host precedentemente specificata dall'utente per un'attività in esecuzione viene prenotata anche mentre tale attività è in esecuzione. Dopo l'arresto di un'attività, la porta dell'host viene rilasciata. Le porte riservate correnti vengono visualizzate `remainingResources` nell'`describe-container-instances` output. Un'istanza di container può contenere fino a 100 porte prenotate alla volta, incluse quelle predefinite. Le porte assegnate automaticamente non vengono conteggiate ai fini della quota di 100 porte prenotate.

name

Tipo: stringa

Obbligatorio: No, richiesto per la configurazione di Service Connect e VPC Lattice in un servizio

Il nome utilizzato per la mappatura delle porte. Questo parametro si applica solo a Service Connect e VPC Lattice. Questo parametro è il nome utilizzato nella configurazione Service Connect e VPC Lattice di un servizio.

Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

Nell'esempio seguente, vengono utilizzati entrambi i campi obbligatori per Service Connect e VPC Lattice.

```
"portMappings": [  
  {  
    "name": string,  }  
]
```



```
    "containerPort": integer
  }
]
```

protocol

Tipo: string

Campo obbligatorio: no

Il protocollo utilizzato per la mappatura delle porte. I valori validi sono tcp e udp. Il valore predefinito è tcp.

Important

È supportato solo tcp per Service Connect. Ricorda che tcp è implicito se questo campo non è impostato.

Important

Il supporto UDP è disponibile solo nelle istanze di container che sono state avviate con la versione 1.2.0 dell'agente del container Amazon ECS (ad esempio l'AMI `amzn-ami-2015.03.c-amazon-ecs-optimized`) o successiva o con agenti del container che sono stati aggiornati alla versione 1.3.0 o successiva. Per passare all'ultima versione dell'agente del container, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Se specifichi una porta dell'host, utilizza la seguente sintassi.

```
"portMappings": [
  {
    "containerPort": integer,
    "hostPort": integer
  }
  ...
]
```

Se desideri una porta dell'host assegnata automaticamente, utilizza la seguente sintassi.

```
"portMappings": [  
  {  
    "containerPort": integer  
  }  
  ...  
]
```

Credenziali del repository privato

`repositoryCredentials`

Tipo: oggetto [RepositoryCredentials](#)

Campo obbligatorio: no

Le credenziali dell'archivio per l'autenticazione di un registro privato.

Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

`credentialsParameter`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `repositoryCredentials`

L'Amazon Resource Name (ARN) del segreto contenente le credenziali dell'archivio privato.

Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

Note

Quando utilizzi l'API Amazon ECS o AWS SDKs se il segreto esiste nella stessa regione dell'attività che stai avviando, puoi utilizzare l'ARN completo o il nome del segreto. AWS CLI Quando si utilizza il AWS Management Console, è necessario specificare l'ARN completo del segreto.

Di seguito viene riportato un frammento di una definizione di attività che mostra i parametri obbligatori:

```
"containerDefinitions": [  
  {  
    "image": "private-repo/private-image",  
    "repositoryCredentials": {  
      "credentialsParameter":  
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"  
    }  
  }  
]
```

Parametri avanzati di definizione del container

I seguenti parametri avanzati di definizione dei contenitori forniscono funzionalità estese al comando `docker run` utilizzato per avviare contenitori sulle istanze di container Amazon ECS.

Argomenti

- [Politica di riavvio](#)
- [Controllo dello stato](#)
- [Ambiente](#)
- [Impostazioni di rete](#)
- [Archiviazione e registrazione](#)
- [Sicurezza](#)
- [Limiti delle risorse](#)
- [Etichette Docker](#)

Politica di riavvio

`restartPolicy`

La politica di riavvio del contenitore e i parametri di configurazione associati. Quando configuri una politica di riavvio per un contenitore, Amazon ECS può riavviare il contenitore senza dover sostituire l'attività. Per ulteriori informazioni, consulta [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#).

`enabled`

Tipo: Booleano

Campo obbligatorio: sì

Specifica se è abilitata una politica di riavvio per il contenitore.

`ignoredExitCodes`

Tipo: array di numeri interi

Campo obbligatorio: no

Un elenco di codici di uscita che Amazon ECS ignorerà e non tenterà di riavviare. Puoi specificare un massimo di 50 codici di uscita del contenitore. Per impostazione predefinita, Amazon ECS non ignora alcun codice di uscita.

`restartAttemptPeriod`

Tipo: integer

Campo obbligatorio: no

Un periodo di tempo (in secondi) durante il quale il contenitore deve funzionare prima di poter tentare un riavvio. Un contenitore può essere riavviato solo una volta ogni secondo. `restartAttemptPeriod` Se un contenitore non è in grado di funzionare per questo periodo di tempo e esce in anticipo, non verrà riavviato. È possibile impostare un minimo `restartAttemptPeriod` di 60 secondi e un massimo `restartAttemptPeriod` di 1800 secondi. Per impostazione predefinita, un contenitore deve funzionare per 300 secondi prima di poter essere riavviato.

Controllo dello stato

`healthCheck`

Comando di controllo dell'integrità del container e parametri di configurazione associati per il container. Per ulteriori informazioni, consulta [Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container](#).

`command`

Matrice di stringhe che rappresenta il comando eseguito dal container per determinare l'integrità. La matrice di stringhe può iniziare con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell predefinita del container. Se non è specificato nessuno dei due, viene utilizzato `CMD`.

Quando si registra una definizione di attività in AWS Management Console, utilizzare un elenco di comandi separati da virgole. Questi comandi vengono convertiti in stringa dopo la creazione della definizione delle attività. Di seguito è riportato un esempio di input per il controllo dell'integrità.

```
CMD-SHELL, curl -f http://localhost/ || exit 1
```

Quando si registra una definizione di attività utilizzando il pannello AWS Management Console JSON, o il AWS CLI APIs, racchiude l'elenco dei comandi tra parentesi. Di seguito è riportato un esempio di input per il controllo dell'integrità.

```
[ "CMD-SHELL", "curl -f http://localhost/ || exit 1" ]
```

Un codice di uscita 0, senza output `stderr`, indica l'esito positivo, mentre un codice di uscita diverso da zero indica un errore.

`interval`

Intervallo di tempo (in secondi) tra ogni controllo dell'integrità. Puoi specificare un valore compreso tra 5 e 300 secondi. Il valore di predefinito è 30 secondi.

`timeout`

Periodo di tempo in secondi per cui attendere che un controllo dell'integrità venga superato prima di considerarlo un errore. Puoi specificare un valore compreso tra 2 e 60 secondi. Il valore di default è 5 secondi.

`retries`

Numero di tentativi per cui riprovare un controllo dello stato non riuscito prima che il container venga considerato non integro. Puoi specificare un valore compreso tra 1 e 10 tentativi. Il valore predefinito è tre tentativi.

`startPeriod`

Periodo di tolleranza facoltativo entro il quale concedere ai container il tempo necessario per il bootstrap prima che i controlli dell'integrità non riusciti vengano conteggiati rispetto al numero massimo di nuovi tentativi. Puoi specificare un valore compreso tra 0 e 300 secondi. Per impostazione predefinita, `startPeriod` è disabilitato.

Se un controllo dello stato va a buon fine all'interno di `startPeriod`, il container è considerato integro e gli errori successivi vengono conteggiati rispetto al numero massimo di nuovi tentativi.

Ambiente

cpu

Tipo: integer

Campo obbligatorio: no

Il numero di unità `cpu` che l'agente del container Amazon ECS riserverà per il container. In Linux, questo parametro viene mappato alla sezione `CpuShares` [Crea un](#) contenitore.

Note

Puoi determinare il numero di unità CPU disponibili per ogni tipo di EC2 istanza Amazon. A tale scopo, moltiplica il numero di v CPUs indicato per quel tipo di istanza nella pagina dei dettagli di [Amazon EC2 Instances](#) per 1.024.

Linux i contenitori condividono unità CPU non allocate con altri contenitori sull'istanza del contenitore con lo stesso rapporto tra la quantità allocata. Ad esempio, supponiamo di eseguire un'attività di container singolo in un tipo di istanza single core con 512 unità di CPU specificate per tale container. Inoltre, tale attività è l'unica in esecuzione sull'istanza di container. In questo esempio, il container può utilizzare la condivisione completa di 1.024 unità CPU in qualsiasi momento. Tuttavia, si supponga di aver avviato un'altra copia della stessa attività su quell'istanza di container. A ogni attività viene garantito un minimo di 512 unità CPU quando necessario. Analogamente, se l'altro container non utilizza la CPU rimanente, ogni container può passare a un maggiore utilizzo della CPU. Tuttavia, se entrambe le attività sono sempre attive al 100%, sono limitate a 512 unità CPU.

Abilitato Linux istanze del contenitore, il Docker il demone sull'istanza del contenitore utilizza il valore della CPU per calcolare i relativi rapporti di condivisione della CPU per i contenitori in esecuzione. Il valore minimo di condivisione della CPU valido consentito dal kernel Linux è 2 e il valore massimo di condivisione della CPU valido consentito dal kernel Linux è 262144. Tuttavia, il parametro CPU non è richiesto ed è possibile utilizzare valori della CPU inferiori a due e superiori

a 262144 nelle definizioni dei contenitori. Per i valori della CPU inferiori a due (incluso null) e superiori a 262144, il comportamento varia in base alla versione dell'agente container Amazon ECS:

- Versioni dell'agente $\leq 1.1.0$: i valori CPU nulli e zero vengono passati a Docker come 0. Docker quindi converte questo valore in 1.024 condivisioni di CPU. I valori di una CPU vengono passati a Docker come uno, quale Linux il kernel si converte in due condivisioni di CPU.
- Versioni dell'agente $\geq 1.2.0$: i valori Null, zero e CPU di uno vengono passati a Docker come due condivisioni di CPU.
- Versioni dell'agente $\geq 1.84.0$: i valori della CPU superiori a 256 vCPU vengono passati a Docker come 256, che equivale a 262144 condivisioni di CPU.

Abilitato Windows nelle istanze di container, la quota di CPU viene applicata come quota assoluta. Windows i contenitori hanno accesso solo alla quantità specificata di CPU definita nella definizione dell'attività. Un valore CPU nullo o pari a zero viene passato a Docker come 0. Windows quindi interpreta questo valore come 1% di una CPU.

Per ulteriori esempi, consulta [Modalità di gestione delle risorse di CPU e memoria di Amazon ECS](#).

gpu

Tipo: oggetto [ResourceRequirement](#)

Campo obbligatorio: no

Il numero di GPUs fisiche che l'agente del container Amazon ECS riserva per il container. Il numero di contenitori GPUs riservati per tutti i contenitori di un'attività non deve superare il numero di disponibili GPUs sull'istanza del contenitore su cui viene avviata l'attività. Per ulteriori informazioni, consulta [Definizioni di attività Amazon ECS per carichi di lavoro GPU](#).

 Note

Questo parametro non è supportato per Windows contenitori.

Elastic Inference accelerator

Tipo: oggetto [ResourceRequirement](#)

Campo obbligatorio: no

Per il tipo `InferenceAccelerator`, il value corrisponde a `deviceName` per un `InferenceAccelerator` specificato in una definizione di attività. Per ulteriori informazioni, consulta [the section called “Nome dell'acceleratore di inferenza elastica”](#).

 Note

Questo parametro non è supportato per Windows contenitori.

`essential`

Tipo: Booleano

Campo obbligatorio: no

Si supponga che il parametro `essential` di un container sia contrassegnato come `true` e che tale container abbia esito negativo o si arresti per qualsiasi motivo. Di conseguenza, tutti gli altri container che fanno parte dell'attività vengono arrestati. Se il parametro `essential` di un container è contrassegnato come `false`, il suo esito negativo non influenza il resto dei container in un'attività. Se questo parametro viene omesso, un container si considera essenziale.

Tutti i processi devono avere almeno un container essenziale. Supponiamo di avere un'applicazione composta da più container. In questo caso, si raggruppano i container utilizzati per uno scopo comune in componenti e si separano i diversi componenti in più definizioni delle attività. Per ulteriori informazioni, consulta [Progetta la tua applicazione per Amazon ECS](#).

```
"essential": true|false
```

`entryPoint`

 Important

Le versioni precedenti dell'agente del container Amazon ECS non gestiscono correttamente i parametri `entryPoint`. In caso di problemi durante l'utilizzo `entryPoint`, aggiorna l'agente del container o inserisci i comandi e gli argomenti come elementi di matrice `command`.

Tipo: array di stringhe

Campo obbligatorio: no

Il punto di ingresso che viene trasmesso al container.

```
"entryPoint": ["string", ...]
```

command

Tipo: array di stringhe

Campo obbligatorio: no

Il comando che viene inviato al container. Questo parametro è Cmd mappato al comando create-container e il COMMAND parametro a docker run. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

workingDirectory

Tipo: string

Campo obbligatorio: no

La directory di lavoro nel container in cui eseguire i comandi. Questo parametro è mappato a WorkingDir nella sezione [Create a container](#) di [Docker Remote API](#) e l'opzione `--workdir` a [docker run](#).

```
"workingDirectory": "string"
```

environmentFiles

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'`--env-file` opzione del comando docker run.

Quando FIPS è abilitato, nomi di bucket con punti (.) (ad esempio, amzn-s3-demo-bucket1.name.example) non sono supportati. La presenza di punti (.) nel nome del bucket impedisce l'avvio dell'attività perché l'agente non può estrarre il file della variabile di ambiente da Amazon S3.

Non è disponibile per Windows contenitori.

Puoi specificare fino a 10 file di ambiente. Il file deve avere un'estensione .env. Ogni riga di un file di ambiente deve contenere una variabile di ambiente nel formato VARIABLE=VALUE. Le righe che iniziano con # vengono trattate come commenti e vengono ignorate.

Se nella definizione del container sono specificate singole variabili di ambiente, hanno la precedenza sulle variabili contenute in un file di ambiente. Se vengono specificati più file di ambiente che contengono la stessa variabile, vengono elaborati dall'alto verso il basso. Consigliamo di utilizzare nomi di variabili univoci. Per ulteriori informazioni, consulta [Passa una singola variabile di ambiente a un contenitore Amazon ECS](#).

value

Tipo: stringa

Campo obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

type

Tipo: stringa

Campo obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è s3.

environment

Tipo: array di oggetti

Campo obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è Env mappato al comando docker create-container e l'opzione --env al comando docker run.

 Important

Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

name

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

value

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il valore della variabile di ambiente.

```
"environment" : [  
  { "name" : "string", "value" : "string" },  
  { "name" : "string", "value" : "string" }  
]
```

secrets

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

name

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del AWS Secrets Manager segreto o l'ARN completo del parametro nel Parameter Store. AWS Systems Manager

Note

Se il parametro Systems Manager Parameter Store o il parametro Secrets Manager esiste nella Regione AWS stessa operazione che si sta avviando, è possibile utilizzare l'ARN completo o il nome del segreto. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

```
"secrets": [  
  {  
    "name": "environment_variable_name",  
    "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"  
  }  
]
```

Impostazioni di rete

disableNetworking

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è true, le reti sono disabilitate all'interno del container.

Note

Questo parametro non è supportato per Windows contenitori o attività che utilizzano la modalità aws-vpc di rete.

Il valore predefinito è `false`.

```
"disableNetworking": true|false
```

links

Tipo: array di stringhe

Campo obbligatorio: no

Il parametro `link` consente ai container di comunicare tra loro senza la necessità di mappatura delle porte. Questo parametro è supportato solo se la modalità di rete di una definizione delle attività è impostata su `bridge`. Il `name:internalName` costruito è analogo a `in name:alias` Docker collegamenti. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura.

Note

Questo parametro non è supportato per Windows contenitori o attività che utilizzano la modalità `awsvpc` di rete.

Important

I container che vengono posizionati nella stessa istanza di container potrebbero comunicare tra loro senza necessità di collegamenti o mappature delle porte dell'host. L'isolamento di rete su un'istanza di container è controllato da gruppi di sicurezza e impostazioni VPC.

```
"links": ["name:internalName", ...]
```

hostname

Tipo: string

Campo obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato `Hostname` al `docker create-container` e `--hostname` all'opzione `docker run`.

 Note

Se utilizzi la modalità di rete `awsvpc`, il parametro `hostname` non è supportato.

```
"hostname": "string"
```

dnsServers

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di server DNS presentato al container.

 Note

Questo parametro non è supportato per Windows contenitori o attività che utilizzano la modalità `awsvpc` di rete.

```
"dnsServers": ["string", ...]
```

dnsSearchDomains

Tipo: array di stringhe

Campo obbligatorio: no

Modello: `^[a-zA-Z0-9-]{0,253}[a-zA-Z0-9]$`

Un elenco di domini di ricerca DNS presentato al container. Questo parametro associa `DnsSearch` nel comando `docker create-container` l'opzione `--dns-search` `docker run`.

 Note

Questo parametro non è supportato per i container o le attività di Windows che utilizzano la modalità di rete `awsvpc`.

```
"dnsSearchDomains": ["string", ...]
```

extraHosts

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato ExtraHosts al comando docker create-container e `--add-host` all'opzione docker run.

Note

Questo parametro non è supportato per i container o le attività di Windows che utilizzano la modalità di rete awsvpc.

```
"extraHosts": [  
  {  
    "hostname": "string",  
    "ipAddress": "string"  
  },  
  ...  
]
```

hostname

Tipo: stringa

Obbligatorio: sì, quando si utilizzano extraHosts

Il nome host da utilizzare nella voce `/etc/hosts`.

ipAddress

Tipo: stringa

Obbligatorio: sì, quando si utilizzano extraHosts

L'indirizzo IP da utilizzare nella voce `/etc/hosts`.

Archiviazione e registrazione

`readonlyRootFilesystem`

Tipo: Booleano

Campo obbligatorio: no

Se il parametro è `true`, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro associa `ReadonlyRootfs` nel comando `docker create-container` l'opzione `--read-only` `docker run`.

Note

Questo parametro non è supportato per i container Windows.

Il valore predefinito è `false`.

```
"readonlyRootFilesystem": true|false
```

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato `Volumes` all'API Docker `create-container` e `--volume` all'opzione `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Per le attività che eseguono il sistema operativo Windows, lasciate il valore predefinito `false`.

`volumesFrom`

Tipo: array di oggetti

Campo obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato `VolumesFrom` al comando `docker create-container` e `--volumes-from` all'opzione `docker run`.

`sourceContainer`

Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

```
"volumesFrom": [  
  {  
    "sourceContainer": "string",  
    "readOnly": true|false  
  }  
]
```

logConfiguration

Tipo: oggetto [LogConfiguration](#)

Campo obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta [Esempi di definizioni di attività Amazon ECS](#).

Questo parametro è mappato LogConfig al comando docker create-container e --log-driver all'opzione docker run. Per impostazione predefinita, i contenitori utilizzano lo stesso driver di registrazione del Docker daemon utilizza. Tuttavia, il contenitore potrebbe utilizzare un driver di registrazione diverso dal Docker daemon specificando un driver di registro con questo parametro nella definizione del contenitore. Per utilizzare un altro driver di log per un container, il sistema di log deve essere configurato correttamente nell'istanza di container (o su un altro server di log per le opzioni di logging in remoto).

Si noti quanto segue quando si specifica una configurazione di log per i container:

- Amazon ECS supporta un sottoinsieme dei driver di registrazione disponibili per Docker demone. Ulteriori driver di log potranno essere disponibili nei rilasci futuri dell'agente del container Amazon ECS.
- Questo parametro richiede la versione 1.18 o successiva di Docker API remota sull'istanza del contenitore.
- L'agente contenitore Amazon ECS che viene eseguito su un'istanza di contenitore deve registrare i driver di registrazione disponibili su quell'istanza con la variabile di ECS_AVAILABLE_LOGGING_DRIVERS ambiente prima che i contenitori posizionati su quell'istanza possano utilizzare queste opzioni di configurazione del registro. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

```
"logConfiguration": {
```

```
"logDriver": "awslogs","fluentd","gelf","json-  
file","journald","splunk","syslog","awsfirelens",  
  "options": {"string": "string"  
    ...},  
"secretOptions": [{  
  "name": "string",  
  "valueFrom": "string"  
}]  
}
```

logDriver

Tipo: stringa

Valori validi: "awslogs", "fluentd", "gelf", "json-file", "journald", "splunk", "syslog", "awsfirelens"

Obbligatorio: sì, quando viene utilizzato logConfiguration

Il driver di log da utilizzare per il container. Per impostazione predefinita, i valori validi elencati in precedenza sono driver di log con i quali l'agente del container Amazon ECS può comunicare.

I driver di log supportati sono awslogs, fluentd, gelf, json-file, journald, syslog, splunk e awsfirelens.

Per ulteriori informazioni su come utilizzare il driver di awslogs registro nelle definizioni delle attività per inviare i log dei contenitori a Logs, consulta CloudWatch . [Invia i log di Amazon ECS a CloudWatch](#)

Per ulteriori informazioni sull'utilizzo del driver di log awsfirelens, consulta [Routing di log personalizzato](#).

Note

Se disponi di un driver personalizzato che non è elencato, puoi eseguire il fork del progetto Amazon ECS Container Agent [disponibile su GitHub](#) e personalizzarlo in modo che funzioni con quel driver. Ti consigliamo di inviare le richieste pull per le modifiche che desideri siano incluse. Tuttavia, attualmente non forniamo il supporto per eseguire copie modificate di questo software.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

options

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

La mappa chiave/valore delle opzioni di configurazione per inviare il driver di log.

Le opzioni che puoi specificare dipendono dal driver di registro. Alcune delle opzioni che puoi specificare quando utilizzi il `awslogs` router per indirizzare i log ad Amazon CloudWatch includono:

`awslogs-create-group`

Campo obbligatorio: no

Specifica se desideri che il gruppo di log venga creato automaticamente. Se questa opzione non è specificata, viene impostata in modo predefinito su `false`.

Note

La tua policy IAM deve includere l'autorizzazione `logs:CreateLogGroup` prima di provare a utilizzare `awslogs-create-group`.

`awslogs-region`

Campo obbligatorio: sì

Specificate il Regione AWS driver di `awslogs` registro a cui inviare i log Docker. Puoi scegliere di inviare tutti i log dai cluster di diverse regioni a una singola regione in Logs. CloudWatch In questo modo saranno tutti visibili in un'unica posizione. In caso contrario, puoi separarli per regione per una maggiore granularità. Assicurati che il gruppo di log specificato esista nella regione definita tramite questa opzione.

`awslogs-group`

Campo obbligatorio: sì

Assicurati di specificare un gruppo di log a cui il driver di log `awslogs` invia i flussi di log.

awslogs-stream-prefix

Obbligatorio: facoltativo

Utilizza l'opzione `awslogs-stream-prefix` per associare un flusso di log al prefisso selezionato, al nome del container e all'ID dell'attività Amazon ECS al quale appartiene il container. Se specifichi un prefisso con questa opzione, il flusso di log assume il formato seguente.

```
prefix-name/container-name/ecs-task-id
```

Se non specifichi un prefisso con questa opzione, il flusso di log viene denominato secondo l'ID del container assegnato dal daemon Docker nell'istanza di container. Poiché è difficile ricollegare i log al container che li ha inviati con il solo ID container Docker (disponibile solo sulle istanze di container), è consigliabile specificare un prefisso con questa opzione.

Per i servizi Amazon ECS, puoi utilizzare il nome del servizio come prefisso. In questo modo puoi ricollegare i flussi di log al servizio a cui appartiene il container, al nome del container che li ha inviati e all'ID dell'attività a cui appartiene il container.

Perché i log compaiano nell'apposito riquadro della console Amazon ECS, devi specificare un prefisso per il flusso dei log.

awslogs-datetime-format

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea nel formato `strftime` Python. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello. In questo modo la riga associata è il delimitatore tra i messaggi di log.

Un esempio di un caso d'uso per l'utilizzo di questo formato è per l'analisi di output, ad esempio uno dump dello stack, che potrebbe altrimenti essere registrato in più voci. Il modello corretto consente di acquisirlo in una sola voce.

Per ulteriori informazioni, consulta [awslogs-datetime-format](#).

Non è possibile configurare entrambe le opzioni `awslogs-datetime-format` e `awslogs-multiline-pattern`.

 Note

Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

`awslogs-multiline-pattern`

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea che utilizza un'espressione regolare. Un messaggio di log è composto da una riga corrispondente al modello e da tutte le righe successive non corrispondenti al modello. In questo modo la riga associata è il delimitatore tra i messaggi di log.

Per ulteriori informazioni, consulta [awslogs-multiline-pattern](#).

Questa opzione viene ignorata se anche `awslogs-datetime-format` è configurato.

Non è possibile configurare entrambe le opzioni `awslogs-datetime-format` e `awslogs-multiline-pattern`.

 Note

Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

Le seguenti opzioni si applicano a tutti i driver di registro supportati.

`mode`

Campo obbligatorio: no

Valori validi: `non-blocking` | `blocking`

Questa opzione definisce la modalità di consegna dei messaggi di registro dal contenitore al driver di registro specificato utilizzando `logDriver`. La modalità di consegna scelta

influisce sulla disponibilità dell'applicazione quando il flusso di log dal contenitore viene interrotto.

Se si utilizza la `blocking` modalità e il flusso di log viene interrotto, le chiamate provenienti dal codice del contenitore su cui scrivere verso `stdout` e `stderr` gli stream verranno bloccate. Di conseguenza, il thread di registrazione dell'applicazione si bloccherà. Ciò può causare la mancata risposta dell'applicazione e la presenza di errori nel controllo d'integrità del container.

Se utilizzi la modalità `non-blocking`, i log del container vengono invece archiviati in un buffer intermedio in memoria configurato con l'opzione `max-buffer-size`. In questo modo si evita che l'applicazione non risponda quando i log non possono essere inviati. Ti consigliamo di utilizzare questa modalità se vuoi garantire la disponibilità del servizio nonostante una certa perdita di log. Per ulteriori informazioni, vedere [Prevenzione della perdita di log con la modalità non bloccante nel driver di registro del awslogs contenitore](#).

È possibile impostare un valore predefinito `mode` per tutti i contenitori in uno specifico Regione AWS utilizzando le impostazioni dell'`defaultLogDriverMode` account. Se non specifichi l'opzione `mode` nell'impostazione dell'account `logConfiguration` o non configuri l'impostazione dell'account, Amazon ECS utilizzerà per impostazione predefinita la `blocking` modalità. Per ulteriori informazioni sull'impostazione dell'account, consulta [Modalità predefinita del driver di registro](#).

 Note

Il 25 giugno 2025, Amazon ECS cambierà la modalità predefinita del driver di registro da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto alla registrazione. Per continuare a utilizzare la `blocking` modalità dopo questa modifica, esegui una delle seguenti operazioni:

- Imposta l'opzione `mode` nella definizione del contenitore `logConfiguration` come `blocking`.
- Imposta l'impostazione `defaultLogDriverMode` dell'account su `blocking`.

`max-buffer-size`

Campo obbligatorio: no

Valore predefinito: 10 m

Quando viene utilizzata la modalità non-blocking, l'opzione di log `max-buffer-size` controlla la dimensione del buffer utilizzato per l'archiviazione dei messaggi intermedi. Assicurati di specificare una dimensione del buffer adeguata in base all'applicazione in uso. Quando il buffer è pieno, non è possibile archiviare ulteriori log. I log che non possono essere archiviati vengono persi.

Per indirizzare i log utilizzando il `splunk log router`, è necessario specificare `splunk-token` e `asplunk-url`.

Quando si utilizza il router di `awsfirelens log` per indirizzare i log verso una AWS Partner Network destinazione Servizio AWS OR per l'archiviazione e l'analisi dei log, è possibile impostare l'opzione `log-driver-buffer-limit` per limitare il numero di eventi che vengono memorizzati nel buffer prima di essere inviati al contenitore del log router. Può aiutare a risolvere un potenziale problema di perdita di log, poiché un throughput elevato potrebbe causare l'esaurimento della memoria per il buffer interno di Docker. Per ulteriori informazioni, vedere [the section called “Configurazione dei log per un throughput elevato”](#).

Le altre opzioni che è possibile specificare quando si utilizzano i log di `awsfirelens to route` dipendono dalla destinazione. Quando esporti i log in Amazon Data Firehose, puoi specificare Regione AWS il `region` with e un nome per il flusso di log con. `delivery_stream`

Quando esporti i log in Amazon Kinesis Data Streams, puoi specificare Regione AWS un `region` with e un nome per il flusso di dati con. `stream`

Quando esporti i log su Amazon OpenSearch Service, puoi specificare opzioni come `name`, `Host` (Endpoint del OpenSearch servizio senza protocollo), `Port`, `Index`, `Type`, `Aws_auth` `Aws_region` `Suppress_Type_Name`, e. `tls`

Quando esporti i log su Amazon S3, puoi specificare il bucket utilizzando l'opzione. `bucket` Puoi anche specificare `region`, `total_file_size` `upload_timeout`, e `use_put_object` come opzioni.

Questo parametro richiede la versione 1.19 o successiva di Docker Remote API sull'istanza di container.

`secretOptions`

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da inviare alla configurazione di log. I segreti utilizzati nella configurazione di log possono includere un token di autenticazione, un certificato o una chiave di crittografia. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

name

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre alla configurazione di log del container.

```
"logConfiguration": {
  "logDriver": "splunk",
  "options": {
    "splunk-url": "https://cloud.splunk.com:8080",
    "splunk-token": "...",
    "tag": "...",
    ...
  },
  "secretOptions": [{
    "name": "splunk-token",
    "valueFrom": "/ecs/logconfig/splunkcred"
  }]
}
```

firelensConfiguration

Tipo: [FirelensConfiguration](#) oggetto

Campo obbligatorio: no

La FireLens configurazione per il contenitore. Si utilizza per specificare e configurare un router di log per i log del container. Per ulteriori informazioni, consulta [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

```
{
  "firelensConfiguration": {
    "type": "fluentd",
    "options": {
      "KeyName": ""
    }
  }
}
```

options

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

La mappa chiave/valore delle opzioni da utilizzare durante la configurazione del router di log. Questo campo è facoltativo e può essere utilizzato per aggiungere ulteriori metadati, ad esempio il processo, la definizione di attività, il cluster e i dettagli dell'istanza di container all'evento di log. Se specificato, la sintassi da utilizzare è "options":{"enable-ecs-log-metadata":"true|false", "config-file-type":"s3|file", "config-file-value":"arn:aws:s3:::*amzn-s3-demo-bucket*/fluent.conf|filepath"}. Per ulteriori informazioni, consulta [Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens](#).

type

Tipo: stringa

Campo obbligatorio: sì

Il router di log da utilizzare. I valori validi sono fluentd o fluentbit.

Sicurezza

Per ulteriori informazioni sulla sicurezza dei container, consulta le [procedure consigliate per la sicurezza dei container e delle attività di Amazon ECS](#).

credentialSpecs

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di file ARNs in SSM o Amazon S3 in un file `CredSpec` () delle credenziali che configura il contenitore per l'autenticazione Active Directory. Consigliamo di utilizzare questo parametro anziché `dockerSecurityOptions`. Il numero massimo di è 1. ARNs

Esistono due formati per ogni ARN.

`credentialsspecdomainless:MyARN`

Utilizza `credentialsspecdomainless:MyARN` per fornire un file `CredSpec` con una sezione aggiuntiva per un segreto in Secrets Manager. Specifica le credenziali di accesso al dominio nel campo segreto.

Ogni attività eseguita su qualsiasi istanza di container può aggiungere domini diversi.

Puoi utilizzare questo formato senza aggiungere l'istanza di container a un dominio.

`credentialspec:MyARN`

Utilizza `credentialspec:MyARN` per fornire un nome a un file `CredSpec` per un singolo dominio.

Devi aggiungere l'istanza di container al dominio prima di iniziare qualsiasi attività che utilizzi questa definizione delle attività.

In entrambi i formati, sostituisci `MyARN` con l'ARN in SSM o Amazon S3.

Il file `credspec` deve fornire un ARN in Secrets Manager per un segreto contenente il nome utente, la password e il dominio a cui collegarsi. Per una maggiore sicurezza, l'istanza non viene aggiunta al dominio per l'autenticazione senza dominio. Le altre applicazioni sull'istanza non possono utilizzare le credenziali senza dominio. Puoi utilizzare questo parametro per eseguire attività sulla stessa istanza, anche se le attività devono aggiungere domini diversi. Per ulteriori informazioni, vedere [Usare g MSAs per Windows Containers](#) e [Using g MSAs for Linux Containers](#).

`privileged`

Tipo: Booleano

Campo obbligatorio: no

Se il parametro è `true`, al container vengono assegnati privilegi elevati nell'istanza di container host (simile all'utente `root`). Si consiglia di non far funzionare i container con `privileged`. Nella maggior parte dei casi, puoi specificare i privilegi esatti necessari utilizzando i parametri specifici anziché `privileged`.

Questo parametro è mappato `Privileged` al comando `docker create-container` e `--privileged` all'opzione `docker run`.

 Note

Questo parametro non è supportato per i container o le attività Windows che utilizzano il tipo di avvio Fargate.

Il valore predefinito è `false`.

```
"privileged": true|false
```

`user`

Tipo: `string`

Campo obbligatorio: `no`

L'utente da usare all'interno del container. Questo parametro è mappato `User` al comando `docker create-container` e `--user` all'opzione `docker run`.

 Important

Quando esegui attività che utilizzano la modalità di rete `host`, non eseguire container utilizzando l'utente `root` (UID 0). Come best practice per la sicurezza, utilizza sempre un utente non `root`.

È possibile specificare `user` utilizzando i seguenti formati. Un eventuale UID o GID deve essere specificato come numero intero positivo.

- `user`
- `user:group`
- `uid`
- `uid:gid`
- `user:gid`
- `uid:group`

Note

Questo parametro non è supportato per i container Windows.

```
"user": "string"
```

dockerSecurityOptions

Tipo: array di stringhe

Valori validi: "no-new-privileges" | «AppArmor:profile» | «label:" | «credentialSpec:» *value CredentialSpecFilePath*

Campo obbligatorio: no

Un elenco di stringhe per fornire una configurazione personalizzata per più sistemi di sicurezza.

In Linux task, questo parametro può essere usato per fare riferimento alle etichette personalizzate per SELinux e AppArmor sistemi di sicurezza a più livelli.

Questo parametro può essere utilizzato per fare riferimento a un file di specifiche delle credenziali che configura un contenitore per l'autenticazione di Active Directory. Per ulteriori informazioni, consultare [Scopri come usare i contenitori g MSAs for EC2 Windows per Amazon ECS](#) e [Utilizzo gMSA per EC2 Linux contenitori su Amazon ECS](#).

Questo parametro è mappato SecurityOpt al comando docker create-container e --security-opt all'opzione docker run.

```
"dockerSecurityOptions": ["string", ...]
```

Note

L'agente del container Amazon ECS che viene eseguito su un'istanza di container deve registrarsi con le variabili di ambiente ECS_SELINUX_CAPABLE=true o ECS_APPARMOR_CAPABLE=true prima che i container posizionati su tale istanza possano utilizzare queste opzioni di sicurezza. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Limiti delle risorse

ulimits

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di valori `ulimit` da definire per un container. Questo valore sovrascrive l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è mappato `Ulimits` al comando `docker create-container` e `--ulimit` all'opzione `docker run`.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

Note

Questo parametro non è supportato per i container Windows.

```
"ulimits": [  
  {  
    "name":  
    "core"|"cpu"|"data"|"fsize"|"locks"|"memlock"|"msgqueue"|"nice"|"nofile"|"nproc"|"rss"|"rtpr  
    "softLimit": integer,  
    "hardLimit": integer  
  }  
  ...  
]
```

name

Tipo: stringa

Valori validi: `"core"` | `"cpu"` | `"data"` | `"fsize"` | `"locks"` | `"memlock"` | `"msgqueue"` | `"nice"` | `"nofile"` | `"nproc"` | `"rss"` | `"rtprio"` | `"rttime"` | `"sigpending"` | `"stack"`

Obbligatorio: sì, quando si utilizzano `ulimits`

type di `ulimit`.

hardLimit

Tipo: integer

Obbligatorio: sì, quando si utilizzano `ulimits`

Il limite rigido per il tipo `ulimit`. Il valore può essere specificato in byte, secondi o come conteggio, a seconda del `type ulimit`

softLimit

Tipo: integer

Obbligatorio: sì, quando si utilizzano `ulimits`

Il limite flessibile per il tipo `ulimit`. Il valore può essere specificato in byte, secondi o come conteggio, a seconda `type` del `ulimit`

Etichette Docker

`dockerLabels`

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Una mappa chiave/valore di etichette da aggiungere al container. Questo parametro è mappato `Labels` al comando `docker create-container` e `--label` all'opzione `docker run`.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

```
"dockerLabels": {"string": "string"
...}
```

Altri parametri di definizione del container

I seguenti parametri di definizione del container possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS utilizzando l'opzione `Configure via JSON` (Configura tramite JSON). Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Argomenti

- [Parametri Linux](#)
- [Dipendenze per i container](#)
- [Timeout del container](#)
- [Controlli di sistema](#)
- [Interactive](#)
- [Pseudoterminale](#)

Parametri Linux

linuxParameters

Tipo: oggetto [LinuxParameters](#)

Campo obbligatorio: no

Linux-opzioni specifiche che vengono applicate al contenitore, ad esempio. [KernelCapabilities](#)

Note

Questo parametro non è supportato per Windows contenitori.

```
"linuxParameters": {
  "capabilities": {
    "add": ["string", ...],
    "drop": ["string", ...]
  }
}
```

capabilities

Tipo: oggetto [KernelCapabilities](#)

Campo obbligatorio: no

Il Linux funzionalità per il contenitore che vengono aggiunte o eliminate dalla configurazione predefinita fornita da Docker. Per ulteriori informazioni su questi Linux funzionalità, consulta la pagina di manuale di [capabilities \(7\)](#) Linux.

add

Tipo: array di stringhe

Valori validi: "ALL" | "AUDIT_CONTROL" | "AUDIT_READ" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT" | "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" | "SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Campo obbligatorio: no

Il Linux funzionalità per il contenitore da aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato CapAdd al comando docker create-container e --cap-add all'opzione docker run.

add

Tipo: array di stringhe

Valori validi: "SYS_PTRACE"

Campo obbligatorio: no

Il Linux funzionalità che il contenitore può aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato CapAdd al comando docker create-container e --cap-add all'opzione docker run.

drop

Tipo: array di stringhe

Valori validi: "ALL" | "AUDIT_CONTROL" | "AUDIT_WRITE" | "BLOCK_SUSPEND" | "CHOWN" | "DAC_OVERRIDE" | "DAC_READ_SEARCH" | "FOWNER" | "FSETID" | "IPC_LOCK" | "IPC_OWNER" | "KILL" | "LEASE" | "LINUX_IMMUTABLE" | "MAC_ADMIN" | "MAC_OVERRIDE" | "MKNOD" | "NET_ADMIN" | "NET_BIND_SERVICE" | "NET_BROADCAST" | "NET_RAW" | "SETFCAP" | "SETGID" | "SETPCAP" | "SETUID" | "SYS_ADMIN" | "SYS_BOOT" | "SYS_CHROOT" | "SYS_MODULE" | "SYS_NICE" | "SYS_PACCT"

| "SYS_PTRACE" | "SYS_RAWIO" | "SYS_RESOURCE" | "SYS_TIME" |
"SYS_TTY_CONFIG" | "SYSLOG" | "WAKE_ALARM"

Campo obbligatorio: no

Il Linux funzionalità per la rimozione del contenitore dalla configurazione predefinita fornita da Docker. Questo parametro è mappato CapDrop al comando docker create-container e --cap-drop all'opzione docker run.

devices

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato Devices al comando docker create-container e --device all'opzione docker run.

Tipo: matrice di oggetti [Device](#)

Campo obbligatorio: no

hostPath

Il percorso per il dispositivo nell'istanza di container dell'host.

Tipo: stringa

Campo obbligatorio: sì

containerPath

Il percorso nel container in cui esporre il dispositivo dell'host.

Tipo: string

Campo obbligatorio: no

permissions

Le autorizzazioni esplicite da fornire al container per il dispositivo. Di default, il container dispone di autorizzazioni per read, write e mknod sul dispositivo.

Tipo: matrice di stringhe

Valori validi: read | write | mknod

initProcessEnabled

Esegui un processo init nel container che inoltra segnali e raccoglie i processi. Questo parametro è mappato all'--initopzione docker run.

Questo parametro richiede la versione 1.25 o successiva di Docker Remote API sull'istanza di container.

maxSwap

La quantità totale di memoria di swap (in MiB) che un container può utilizzare. Questo parametro viene tradotto nell' `--memory-swap` opzione docker run dove il valore è la somma della memoria del contenitore più il maxSwap valore.

Se viene specificato il valore maxSwap di 0, il container non utilizzerà lo swap. I valori accettati sono 0 o qualsiasi numero intero positivo. Se il parametro maxSwap viene omesso, il container utilizza la configurazione di swap per l'istanza di container su cui è in esecuzione. È necessario impostare un valore maxSwap per il parametro swappiness da utilizzare.

sharedMemorySize

Valore per le dimensioni (in MiB) del volume `/dev/shm`. Questo parametro è mappato all' `--shm-size` opzione docker run.

Tipo: integer

swappiness

Puoi utilizzare questo parametro per ottimizzare il funzionamento swappiness della memoria di un container. Un valore swappiness di 0 impedisce che si verifichi lo scambio, a meno che non sia necessario. Un valore swappiness di 100 fa sì che le pagine vengano scambiate frequentemente. I valori accettati sono numeri interi compresi tra 0 e 100. Se non specifichi un valore, viene utilizzato il valore predefinito di 60. Inoltre, se non specifichi un valore per maxSwap, questo parametro verrà ignorato. Questo parametro è mappato all' `--memory-swappiness` opzione docker run.

Note

Se utilizzi le attività su Amazon Linux 2023, il parametro swappiness non è supportato.

tmpfs

Il percorso del contenitore, le opzioni di montaggio e la dimensione massima (in MiB) del tmpfs montare. Questo parametro è mappato all' `--tmpfs` opzione docker run.

Tipo: matrice di oggetti [Tmpfs](#)

Campo obbligatorio: no

`containerPath`

Il percorso assoluto del file in cui tmpfs il volume deve essere montato.

Tipo: stringa

Campo obbligatorio: sì

`mountOptions`

L'elenco di tmpfs opzioni di montaggio del volume.

Tipo: matrice di stringhe

Campo obbligatorio: no

Valori validi: "defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"

`size`

La dimensione massima (in MiB) del tmpfs volume.

Tipo: integer

Campo obbligatorio: sì

Dipendenze per i container

`dependsOn`

Tipo: matrice di oggetti [ContainerDependency](#)

Campo obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un container può contenere più dipendenze. Se una dipendenza è definita per l'avvio del container, per la sua chiusura è invertita. Per un esempio, consulta [Dipendenze per i container](#).

 Note

Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Le istanze richiedono almeno la versione 1.26.0 dell'agente contenitore per abilitare le dipendenze del contenitore. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#). Se utilizzi l'AMI Amazon Linux ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione 1.26.0-1 del pacchetto `ecs-init`. Le istanze di container avviate dalla versione 20190301 o successive contengono già le versioni richieste dell'agente del container e di `ecs-init`. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

```
"dependsOn": [  
  {  
    "containerName": "string",  
    "condition": "string"  
  }  
]
```

containerName

Tipo: stringa

Campo obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

condition

Tipo: stringa

Campo obbligatorio: sì

La condizione di dipendenza del container. Di seguito sono elencate le condizioni disponibili e il loro comportamento:

- **START**: questa condizione emula il comportamento dei collegamenti e dei volumi. La condizione convalida l'avvio di un container dipendente prima di consentire l'avvio di altri container.
- **COMPLETE**: questa condizione verifica l'esecuzione fino al completamento (uscita) di un container dipendente prima di consentire l'avvio di altri container. Può rivelarsi utile per container non essenziali che eseguono uno script e quindi escono. Questa condizione non può essere impostata su un container essenziale.
- **SUCCESS**: questa condizione è uguale a **COMPLETE**, ma richiede anche che il container esca con stato `zero`. Questa condizione non può essere impostata su un container essenziale.
- **HEALTHY**: questa condizione verifica che il container dipendente superi il controllo dell'integrità del container prima di consentire l'avvio di altri container. Ciò richiede che per il container dipendente siano configurati i controlli dell'integrità nella definizione di attività. Questa condizione viene confermata solo all'avvio dell'attività.

Timeout del container

`startTimeout`

Tipo: `integer`

Campo obbligatorio: `no`

Valori di esempio: `120`

Tempo di attesa (in secondi) prima di rinunciare a risolvere le dipendenze per un container.

Ad esempio, vengono specificati due container in una definizione di attività: `containerA` ha una dipendenza sul `containerB` quando raggiunge lo stato **COMPLETE**, **SUCCESS** o **HEALTHY**. Se per `containerB` è specificato un valore `startTimeout` e non raggiunge lo stato desiderato entro tale periodo di tempo, allora `containerA` non viene avviato.

Note

Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Il valore massimo è 120 secondi.

stopTimeout

Tipo: integer

Campo obbligatorio: no

Valori di esempio: 120

Durata (in secondi) di attesa prima che sia forzata la chiusura se il container non si arresta da solo normalmente.

Se il `stopTimeout` parametro non è specificato, viene utilizzato il valore impostato per la variabile `ECS_CONTAINER_STOP_TIMEOUT` di configurazione dell'agente container Amazon ECS. Se non sono impostati né il `stopTimeout` parametro né la variabile di configurazione dell'`ECS_CONTAINER_STOP_TIMEOUT` agente, i valori predefiniti di 30 secondi per Linux contenitori e 30 secondi accesi Windows vengono utilizzati contenitori. Per consentire il valore di timeout di arresto per un container, le istanze di container richiedono almeno la versione 1.26.0 dell'agente del container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#). Se utilizzi l'AMI Amazon Linux ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione 1.26.0-1 del pacchetto `ecs-init`. Le istanze di container avviate dalla versione 20190301 o successive contengono già le versioni richieste dell'agente del container e di `ecs-init`. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Controlli di sistema

systemControls

Tipo: oggetto [SystemControl](#)

Campo obbligatorio: no

Un elenco di parametri del kernel dello spazio dei nomi da impostare nel contenitore. Questo parametro è mappato `Sysctl`s al comando `docker create-container` e all'opzione `docker run`. -- `sysctl` Ad esempio, puoi configurare l'impostazione `net.ipv4.tcp_keepalive_time` per mantenere le connessioni di lunga durata.

Non è consigliabile specificare i parametri `systemControls` correlati alla rete per più container in un'unica attività che utilizza anche la modalità di rete `awsvpc` o `host`. Ciò comporta i seguenti svantaggi:

- Per le attività che utilizzano la modalità di rete `awsvpc`, se hai impostato `systemControls` per qualsiasi container, si applica a tutti i container nell'attività. Se hai impostato diversi `systemControls` per più container in un'unica attività, il container che viene avviato per ultimo determina quale `systemControls` diventa effettivo.
- Per attività che utilizzano la modalità di rete `host`, lo spazio dei nomi della rete `systemControls` non è supportato.

Se stai impostando uno spazio dei nomi della risorsa IPC per utilizzare i container nell'attività, ai controlli di sistema si applicano le seguenti condizioni. Per ulteriori informazioni, consulta [Modalità IPC](#).

- Per le attività che utilizzano la modalità `host IPC`, i valori `systemControls` dello spazio dei nomi IPC non sono supportati.
- Per le attività che utilizzano la modalità `IPC task`, i valori di `systemControls` dello spazio dei nomi IPC si applicano a tutti i container all'interno di un'attività.

 Note

Questo parametro non è supportato per i container Windows.

```
"systemControls": [  
  {  
    "namespace": "string",  
    "value": "string"  
  }  
]
```

namespace

Tipo: string

Campo obbligatorio: no

Il parametro del kernel dello spazio dei nomi per cui impostare un. value

Valori dello spazio dei nomi IPC validi: "kernel.msgmax" | "kernel.msgmnb" | "kernel.msgmni" | "kernel.sem" | "kernel.shmall" | "kernel.shmmax" | "kernel.shmmni" | "kernel.shm_rmid_forced" e Sysctls che iniziano con "fs.mqueue.*"

Valori dello spazio dei nomi di rete validi: Sysctls che iniziano con "net.*"

value

Tipo: string

Campo obbligatorio: no

Il valore per il parametro del kernel dello spazio dei nomi specificato in. namespace

Interactive

interactive

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è true, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di stdin o tty. Questo parametro è mappato OpenStdin al comando docker create-container e --interactive all'opzione docker run.

Il valore predefinito è false.

Pseudoterminale

pseudoTerminal

Tipo: Booleano

Campo obbligatorio: no

Quando il parametro è true, è allocato un TTY. Questo parametro è mappato Tty al comando docker create-container e --tty all'opzione docker run.

Il valore predefinito è false.

Nome dell'acceleratore di inferenza elastica

Il requisito della risorsa dell'acceleratore di Elastic Inference per la definizione di attività.

Note

Amazon Elastic Inference (EI) non è più disponibile per i clienti.

I parametri seguenti sono permessi nella definizione di un processo:

`deviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del dispositivo dell'acceleratore di inferenza elastica. Il `deviceName` deve anche essere riferito in una definizione del container; consulta [Elastic Inference accelerator](#).

`deviceType`

Tipo: stringa

Campo obbligatorio: sì

L'acceleratore di inferenza elastica da utilizzare.

Vincoli di posizionamento delle attività

Quando registri una definizione di attività, puoi specificare vincoli di posizionamento dei processi che definiscono il modo in cui Amazon ECS posiziona i processi.

Puoi utilizzare i vincoli per posizionare le attività in base alla zona di disponibilità, al tipo di istanza o agli attributi personalizzati. Per ulteriori informazioni, consulta [Definisci quali istanze di container Amazon ECS utilizza per le attività](#).

I seguenti parametri sono consentiti in una definizione del container:

`expression`

Tipo: string

Campo obbligatorio: no

Un'espressione del linguaggio di query del cluster da applicare al vincolo. Per ulteriori informazioni, consulta [Crea espressioni per definire istanze di container per le attività di Amazon ECS](#).

type

Tipo: stringa

Campo obbligatorio: sì

Il tipo di vincolo. Utilizza `memberOf` per limitare la selezione a un gruppo di candidati validi.

Configurazione del proxy

proxyConfiguration

Tipo: oggetto [ProxyConfiguration](#)

Campo obbligatorio: no

I dettagli di configurazione del proxy App Mesh.

Per le attività che utilizzano il tipo di EC2 avvio, le istanze del contenitore richiedono almeno la versione 1.26.0 dell'agente contenitore e almeno la versione 1.26.0-1 del pacchetto per abilitare una configurazione proxy. `ecs-init` Se le istanze di container sono avviate dall'AMI ottimizzata per Amazon ECS versione 20190301 o successiva, contengono le versioni richieste dell'agente container e di `ecs-init`. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Note

Questo parametro non è supportato per i container Windows.

```
"proxyConfiguration": {  
  "type": "APPMESH",  
  "containerName": "string",  
  "properties": [  

```

```
{
  "name": "string",
  "value": "string"
}
```

type

Tipo: stringa

Valori del valore: APPMESH

Campo obbligatorio: no

Il tipo di proxy. L'unico valore supportato è APPMESH.

containerName

Tipo: stringa

Campo obbligatorio: sì

Il nome del container che agirà come proxy App Mesh.

properties

Tipo: matrice di oggetti [KeyValuePair](#)

Campo obbligatorio: no

Il set di parametri di configurazione di rete per il plug-in Container Network Interface (CNI), specificati come coppie chiave-valore.

- **IgnoredUID:** (obbligatorio) l'ID utente (UID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato `IgnoredGID`, questo campo può rimanere vuoto.
- **IgnoredGID:** (obbligatorio) l'ID gruppo (GID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato `IgnoredUID`, questo campo può rimanere vuoto.
- **AppPorts** (obbligatorio): l'elenco delle porte utilizzate dall'applicazione. Il traffico di rete per queste porte viene inoltrato alle porte `ProxyIngressPort` e `ProxyEgressPort`.

- `ProxyIngressPort` (obbligatorio): specifica la porta a cui è diretto il traffico in entrata per le `AppPorts`.
- `ProxyEgressPort` (obbligatorio): specifica la porta a cui è diretto il traffico in uscita da `AppPorts`.
- `EgressIgnoredPorts`: (obbligatorio) il traffico in uscita diretto verso le porte specificate viene ignorato e non reindirizzato alla `ProxyEgressPort`. Può essere un elenco vuoto.
- `EgressIgnoredIPs`: (obbligatorio) il traffico in uscita diretto verso gli indirizzi IP specificati viene ignorato e non reindirizzato alla `ProxyEgressPort`. Può essere un elenco vuoto.

`name`

Tipo: string

Campo obbligatorio: no

Nome della coppia chiave-valore.

`value`

Tipo: string

Campo obbligatorio: no

Valore della coppia chiave-valore.

Volumi

Quando si registra una definizione di attività, è possibile specificare facoltativamente un elenco di volumi da passare al Docker demone su un'istanza di contenitore, che diventa quindi disponibile per l'accesso da parte di altri contenitori sulla stessa istanza di contenitore.

Di seguito sono elencati i tipi di volumi di dati che è possibile utilizzare:

- **Volumi Amazon EBS:** fornisce storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Puoi collegare 1 volume Amazon EBS per attività Amazon ECS durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. I volumi Amazon EBS sono supportati per le attività Linux. Per ulteriori informazioni, consulta [Usa i volumi Amazon EBS con Amazon ECS](#).
- **Volumi Amazon EFS:** offre uno spazio di archiviazione di file semplice, scalabile e persistente da utilizzare con i processi Amazon ECS. Con Amazon EFS, la capacità di storage è elastica. La

capacità di storage aumenta e si riduce automaticamente quando si aggiungono e si rimuovono i file. Le tue applicazioni possono disporre dello storage di cui hanno bisogno e quando ne hanno bisogno. I volumi Amazon EFS sono supportati. Per ulteriori informazioni, consulta [Usa i volumi Amazon EFS con Amazon ECS](#).

- FSx per volumi Windows File Server: fornisce file server Microsoft Windows completamente gestiti. Questi server di file sono supportati da un file system Windows. Quando si utilizza FSx Windows File Server insieme ad Amazon ECS, è possibile effettuare il provisioning delle attività Windows con uno storage di file persistente, distribuito, condiviso e statico. Per ulteriori informazioni, consulta [Utilizzo FSx per volumi Windows File Server con Amazon ECS](#).

Questa opzione non è supportata per i container Windows su Fargate.

- Volumi Docker: un volume gestito da Docker creato nell'/var/lib/docker/volumes istanza Amazon host. EC2 I driver del volume Docker (detti anche plug-in) vengono utilizzati per integrare i volumi ai sistemi di archiviazione esterni, ad esempio Amazon EBS. È possibile utilizzare il driver del volume `local` integrato o un driver di volume di terza parte. I volumi Docker sono supportati solo quando si eseguono attività su EC2 istanze Amazon. I contenitori Windows supportano solo l'uso del `local` driver. Per usare i volumi Docker, specifica `dockerVolumeConfiguration` nella definizione di attività.
- Bind mounts: un file o una directory sul computer host montato in un contenitore. I volumi host Bind Mount sono supportati. Per usare i volumi host di montaggi vincolati, specifica un valore `host` e un valore opzionale `sourcePath` nella definizione di attività.

Per ulteriori informazioni, consulta [Opzioni di storage per le attività di Amazon ECS](#).

I seguenti parametri sono consentiti in una definizione del container.

`name`

Tipo: string

Campo obbligatorio: no

Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini (`-`) e caratteri di sottolineatura (`_`). A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume mountPoints`

`host`

Campo obbligatorio: no

Il `host` parametro viene utilizzato per collegare il ciclo di vita del `bind mount` all' EC2 istanza Amazon host, anziché all'attività e alla posizione in cui è archiviata. Se il parametro `host` è vuoto, il daemon Docker assegna un percorso host per il tuo volume di dati, ma non è garantito che i dati vengano mantenuti dopo che viene interrotta l'esecuzione del container a essi associato.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`.

 Note

Il `sourcePath` parametro è supportato solo quando si utilizzano attività ospitate su EC2 istanze Amazon.

`sourcePath`

Tipo: string

Campo obbligatorio: no

Quando viene utilizzato il `host` parametro, specifica `sourcePath` a per dichiarare il percorso sull' EC2 istanza Amazon host che viene presentata al contenitore. Se questo parametro è vuoto, il daemon Docker assegna automaticamente un percorso host. Se il `host` parametro contiene una posizione di `sourcePath` file, il volume di dati persiste nella posizione specificata sull' EC2 istanza Amazon host finché non lo elimini manualmente. Se il `sourcePath` valore non esiste sull' EC2 istanza Amazon host, il demone Docker lo crea. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

`configuredAtLaunch`

Tipo: Booleano

Campo obbligatorio: no

Specifica se un volume è configurabile all'avvio. Se impostato su `true`, è possibile configurare il volume durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. Se impostato su `true`, non sarà possibile fornire un'altra configurazione del volume nella definizione dell'attività. Questo parametro deve essere impostato per `true` configurare un volume Amazon EBS da allegare a un'attività. L'impostazione `configuredAtLaunch` `true` e il rinvio della configurazione del volume alla fase di avvio consentono di creare definizioni di attività che non sono limitate a un tipo di volume o a impostazioni di volume specifiche. In questo modo la

definizione delle attività è riutilizzabile in diversi ambienti di esecuzione. Per ulteriori informazioni consulta [Amazon EBS volumes](#).

`dockerVolumeConfiguration`

Tipo: oggetto [DockerVolumeConfiguration](#)

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker. I volumi Docker sono supportati solo quando si eseguono attività su EC2 istanze. I contenitori Windows supportano solo l'uso del `local` driver. Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

Tipo: stringa

Valori validi: `task` | `shared`

Campo obbligatorio: no

L'ambito del volume Docker che determina il suo ciclo di vita. I volumi Docker che rientrano nell'ambito `task` vengono automaticamente assegnati all'avvio del processo e distrutti quando il processo viene arrestato. I volumi Docker che vengono definiti come `shared` vengono mantenuti dopo l'arresto del processo.

`autoprovision`

Tipo: Booleano

Valore predefinito: `false`

Campo obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo viene utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omissso.

`driver`

Tipo: string

Campo obbligatorio: no

Il driver del volume Docker da utilizzare. Il valore del driver deve corrispondere al nome del driver fornito da Docker perché questo nome viene utilizzato per il posizionamento delle attività. Se il driver è stato installato utilizzando la CLI del plug-in Docker, `docker plugin ls` utilizzalo per recuperare il nome del driver dall'istanza del contenitore. Se il driver è stato installato utilizzando un altro metodo, utilizza Docker Plugin Discovery per recuperare il nome del driver.

`driverOpts`

Tipo: string

Campo obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da esaminare. Questo parametro viene mappato `DriverOpts` alla sezione Crea un volume di Docker.

`labels`

Tipo: string

Campo obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`efsVolumeConfiguration`

Tipo: oggetto [EFSVolumedi configurazione](#)

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

Tipo: string

Campo obbligatorio: no

La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omissso, verrà utilizzata la radice del volume Amazon EFS. La specifica di / avrà lo stesso effetto dell'omissione di questo parametro.

 Important

Se un punto di accesso EFS è specificato in `authorizationConfig`, il parametro della directory principale deve essere omissso o impostato su /, il che applicherà il percorso impostato sul punto di accesso EFS.

`transitEncryption`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se abilitare o meno la crittografia per i dati Amazon EFS in transito tra l'host Amazon ECS e il server Amazon EFS. Se si utilizza l'autorizzazione IAM di Amazon EFS, è necessario abilitare la crittografia di transito. Se questo parametro viene omissso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni, consulta [Crittografia dei dati in transito](#) nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: integer

Campo obbligatorio: no

La porta da utilizzare per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non specifichi una porta di crittografia di transito, l'attività utilizzerà la strategia di selezione delle porte utilizzata dall'helper di montaggio di Amazon EFS. Per ulteriori informazioni, consulta [Assistente per il montaggio di EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: oggetto [EFSAuthorizationConfig](#)

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

Tipo: string

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory principale in `efsVolumeConfiguration` deve essere omesso o impostato su `/`, il che applicherà il percorso impostato sul punto di accesso EFS. Se si utilizza un punto di accesso, la crittografia di transito deve essere abilitata in `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Amazon EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`iam`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Specifica se utilizzare il ruolo IAM dell'attività Amazon ECS definito in una definizione di attività durante il montaggio del file system Amazon EFS. Se abilitato, la crittografia di transito deve essere abilitata nella casella `EFSVolumeConfiguration`. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di DISABLED. Per ulteriori informazioni consulta [Ruoli IAM per le attività](#).

`FSxWindowsFileServerVolumeConfiguration`

Tipo: oggetto [FSxWindowsFileServerVolumeConfiguration](#)

Campo obbligatorio: sì

Questo parametro viene specificato quando utilizzi un file system [Amazon FSx for Windows File Server](#) per l'archiviazione delle attività.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

Campo obbligatorio: sì

La directory all'interno del FSx file system Windows File Server da montare come directory principale all'interno dell'host.

`authorizationConfig`

`credentialsParameter`

Tipo: stringa

Campo obbligatorio: sì

Le opzioni delle credenziali di autorizzazione.

opzioni:

- Amazon Resource Name (ARN) di un [AWS Secrets Manager](#)segreto.
- ARN di un [AWS Systems Manager](#)parametro.

`domain`

Tipo: stringa

Campo obbligatorio: sì

Un nome di dominio completo ospitato da una directory [AWS Directory Service for Microsoft Active Directory](#)(AWS Managed Microsoft AD) o da un Active Directory ospitato autonomamente EC2 .

Tag

Quando registri una definizione di attività, puoi facoltativamente specificare tag di metadati applicati alla definizione di attività. I tag consentono di suddividere in categorie e organizzare la definizione di attività. Ciascun tag è formato da una chiave e da un valore facoltativo. Li definisci entrambi. Per ulteriori informazioni, consulta [Taggare le risorse Amazon ECS](#).

⚠ Important

Non aggiungere Informazioni personali di identificazione o altre informazioni riservate o sensibili nei tag. I tag sono accessibili a molti AWS servizi, inclusa la fatturazione. I tag non sono destinati a essere utilizzati per dati privati o sensibili.

I seguenti parametri sono consentiti in un oggetto di tag.

key

Tipo: string

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

value

Tipo: string

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

Altri parametri di definizione di attività

I seguenti parametri di definizione di attività possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS con l'opzione Configure via JSON (Configura tramite JSON). Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Argomenti

- [Modalità IPC](#)
- [Modalità PID](#)
- [Iniezione in errore](#)

Modalità IPC

ipcMode

Tipo: string

Campo obbligatorio: no

Lo spazio dei nomi della risorsa IPC da utilizzare per i container nell'attività. I valori validi sono `host`, `task` o `none`. Se `host` specificato, tutti i contenitori che rientrano nelle attività che hanno specificato la modalità `host` IPC sulla stessa istanza di contenitore condividono le stesse risorse IPC con l'istanza Amazon EC2 host. Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono le stesse risorse IPC. Se è stato specificato `none`, le risorse IPC all'interno dei container di un'attività sono private e non condivise con altri container in un'attività o sull'istanza di container. Se non è stato specificato alcun valore, la condivisione dello spazio dei nomi della risorsa IPC dipende dalle impostazioni del daemon Docker sull'istanza del container.

Se viene utilizzata la modalità `host` IPC, tieni presente che esiste un maggiore rischio di esposizione a spazi dei nomi IPC indesiderati.

Se stai impostando parametri kernel associati a uno spazio dei nomi utilizzando `systemControls` per i container nell'attività, allo spazio dei nomi della risorsa IPC si applica quanto segue.

- Per le attività che utilizzano la modalità `host` IPC, spazi dei nomi IPC correlati a `systemControls` non sono supportati.
- Per le attività che utilizzano la modalità `task` IPC, `systemControls` correlati allo spazio dei nomi IPC vengono applicati a tutti i container all'interno di un'attività.

Modalità PID

pidMode

Tipo: stringa

Valori validi: `host` | `task`

Campo obbligatorio: no

Lo spazio dei nomi del processo da utilizzare per i container nell'attività. I valori validi sono `host` o `task`. Ad esempio, i sidecar di monitoraggio potrebbero aver bisogno di `pidMode` per accedere a informazioni su altri container in esecuzione nella stessa attività.

Se `host` specificato, tutti i contenitori all'interno delle attività che hanno specificato la modalità `host PID` sulla stessa istanza di contenitore condividono lo stesso spazio dei nomi di processo con l'istanza Amazon EC2 host.

Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono lo stesso spazio dei nomi del processo.

Se non è stato specificato alcun valore, l'impostazione predefinita è uno spazio dei nomi privato per ogni container.

Se viene utilizzata la modalità `host PID`, esiste un maggiore rischio di esposizione a spazi dei nomi di attività indesiderati.

Note

Questo parametro non è supportato per i container Windows.

Iniezione in errore

`enableFaultInjection`

Tipo: Booleano

Valori validi: `true` | `false`

Campo obbligatorio: no

Se questo parametro è impostato su `true`, nel payload di un'attività, Amazon ECS accetta richieste di iniezione di errori dai contenitori dell'attività. Questo parametro è impostato su `false` per impostazione predefinita.

Modello di definizione delle attività di Amazon ECS

Di seguito è riportato un modello di definizione di attività vuoto. Puoi utilizzare questo modello per creare la definizione dell'attività, che può quindi essere incollata nell'area di input JSON della console

o salvata in un file e utilizzata con l'opzione. AWS CLI `--cli-input-json` Per ulteriori informazioni, consulta [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#).

Modello del tipo di EC2 lancio di Amazon

```
{
  "family": "",
  "taskRoleArn": "",
  "executionRoleArn": "",
  "networkMode": "none",
  "containerDefinitions": [
    {
      "name": "",
      "image": "",
      "repositoryCredentials": {
        "credentialsParameter": ""
      },
      "cpu": 0,
      "memory": 0,
      "memoryReservation": 0,
      "links": [""],
      "portMappings": [
        {
          "containerPort": 0,
          "hostPort": 0,
          "protocol": "tcp"
        }
      ],
      "restartPolicy": {
        "enabled": true,
        "ignoredExitCodes": [0],
        "restartAttemptPeriod": 180
      },
      "essential": true,
      "entryPoint": [""],
      "command": [""],
      "environment": [
        {
          "name": "",
          "value": ""
        }
      ],
      "environmentFiles": [
        {
```



```
        "value": "",
        "type": "s3"
    }
],
"mountPoints": [
    {
        "sourceVolume": "",
        "containerPath": "",
        "readOnly": true
    }
],
"volumesFrom": [
    {
        "sourceContainer": "",
        "readOnly": true
    }
],
"linuxParameters": {
    "capabilities": {
        "add": [""],
        "drop": [""],
    },
    "devices": [
        {
            "hostPath": "",
            "containerPath": "",
            "permissions": ["read"]
        }
    ],
    "initProcessEnabled": true,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "",
            "size": 0,
            "mountOptions": [""],
        }
    ],
    "maxSwap": 0,
    "swappiness": 0
},
"secrets": [
    {
        "name": "",
```

```
        "valueFrom": ""
    },
],
"dependsOn": [
    {
        "containerName": "",
        "condition": "COMPLETE"
    }
],
"startTimeout": 0,
"stopTimeout": 0,
"hostname": "",
"user": "",
"workingDirectory": "",
"disableNetworking": true,
"privileged": true,
"readonlyRootFilesystem": true,
"dnsServers": [""],
"dnsSearchDomains": [""],
"extraHosts": [
    {
        "hostname": "",
        "ipAddress": ""
    }
],
"dockerSecurityOptions": [""],
"interactive": true,
"pseudoTerminal": true,
"dockerLabels": {
    "KeyName": ""
},
"ulimits": [
    {
        "name": "nofile",
        "softLimit": 0,
        "hardLimit": 0
    }
],
"logConfiguration": {
    "logDriver": "splunk",
    "options": {
        "KeyName": ""
    },
    "secretOptions": [
```

```

        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"healthCheck": {
    "command": [""],
    "interval": 0,
    "timeout": 0,
    "retries": 0,
    "startPeriod": 0
},
"systemControls": [
    {
        "namespace": "",
        "value": ""
    }
],
"resourceRequirements": [
    {
        "value": "",
        "type": "InferenceAccelerator"
    }
],
"firelensConfiguration": {
    "type": "fluentbit",
    "options": {
        "KeyName": ""
    }
}
},
"volumes": [
    {
        "name": "",
        "host": {
            "sourcePath": ""
        },
        "configuredAtLaunch": true,
        "dockerVolumeConfiguration": {
            "scope": "shared",
            "autoprovision": true,
            "driver": "",

```

```

    "driverOpts": {
      "KeyName": ""
    },
    "labels": {
      "KeyName": ""
    }
  },
  "efsVolumeConfiguration": {
    "fileSystemId": "",
    "rootDirectory": "",
    "transitEncryption": "DISABLED",
    "transitEncryptionPort": 0,
    "authorizationConfig": {
      "accessPointId": "",
      "iam": "ENABLED"
    }
  },
  "fsxWindowsFileServerVolumeConfiguration": {
    "fileSystemId": "",
    "rootDirectory": "",
    "authorizationConfig": {
      "credentialsParameter": "",
      "domain": ""
    }
  }
},
"placementConstraints": [
  {
    "type": "memberOf",
    "expression": ""
  }
],
"requiresCompatibilities": ["EC2"],
"cpu": "",
"memory": "",
"tags": [
  {
    "key": "",
    "value": ""
  }
],
"pidMode": "task",
"ipcMode": "task",

```

```
"proxyConfiguration": {
  "type": "APPMESH",
  "containerName": "",
  "properties": [
    {
      "name": "",
      "value": ""
    }
  ]
},
"inferenceAccelerators": [
  {
    "deviceName": "",
    "deviceType": ""
  }
],
"ephemeralStorage": {
  "sizeInGiB": 0
},
"runtimePlatform": {
  "cpuArchitecture": "X86_64",
  "operatingSystemFamily": "WINDOWS_SERVER_20H2_CORE"
}
}
```

Modello del tipo di lancio Fargate

Important

Per il tipo di lancio Fargate, è necessario includere il `operatingSystemFamily` parametro con uno dei seguenti valori:

- LINUX
- WINDOWS_SERVER_2019_FULL
- WINDOWS_SERVER_2019_CORE
- WINDOWS_SERVER_2022_FULL
- WINDOWS_SERVER_2022_CORE

```
{
  "family": "",
  "runtimePlatform": {"operatingSystemFamily": ""},
  "taskRoleArn": "",
  "executionRoleArn": "",
  "networkMode": "awsvpc",
  "platformFamily": "",
  "containerDefinitions": [
    {
      "name": "",
      "image": "",
      "repositoryCredentials": {"credentialsParameter": ""},
      "cpu": 0,
      "memory": 0,
      "memoryReservation": 0,
      "links": [""],
      "portMappings": [
        {
          "containerPort": 0,
          "hostPort": 0,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "entryPoint": [""],
      "command": [""],
      "environment": [
        {
          "name": "",
          "value": ""
        }
      ],
      "environmentFiles": [
        {
          "value": "",
          "type": "s3"
        }
      ],
      "mountPoints": [
        {
          "sourceVolume": "",
          "containerPath": "",
          "readOnly": true
        }
      ]
    }
  ]
}
```

```
    }
  ],
  "volumesFrom": [
    {
      "sourceContainer": "",
      "readOnly": true
    }
  ],
  "linuxParameters": {
    "capabilities": {
      "add": [""],
      "drop": [""],
    },
    "devices": [
      {
        "hostPath": "",
        "containerPath": "",
        "permissions": ["read"]
      }
    ],
    "initProcessEnabled": true,
    "sharedMemorySize": 0,
    "tmpfs": [
      {
        "containerPath": "",
        "size": 0,
        "mountOptions": [""],
      }
    ],
    "maxSwap": 0,
    "swappiness": 0
  },
  "secrets": [
    {
      "name": "",
      "valueFrom": ""
    }
  ],
  "dependsOn": [
    {
      "containerName": "",
      "condition": "HEALTHY"
    }
  ],
],
```

```
"startTimeout": 0,
"stopTimeout": 0,
"hostname": "",
"user": "",
"workingDirectory": "",
"disableNetworking": true,
"privileged": true,
"readonlyRootFilesystem": true,
"dnsServers": [""],
"dnsSearchDomains": [""],
"extraHosts": [
  {
    "hostname": "",
    "ipAddress": ""
  }
],
"dockerSecurityOptions": [""],
"interactive": true,
"pseudoTerminal": true,
"dockerLabels": {"KeyName": ""},
"ulimits": [
  {
    "name": "msgqueue",
    "softLimit": 0,
    "hardLimit": 0
  }
],
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {"KeyName": ""},
  "secretOptions": [
    {
      "name": "",
      "valueFrom": ""
    }
  ]
},
"healthCheck": {
  "command": [""],
  "interval": 0,
  "timeout": 0,
  "retries": 0,
  "startPeriod": 0
},
```



```

    "systemControls": [
      {
        "namespace": "",
        "value": ""
      }
    ],
    "resourceRequirements": [
      {
        "value": "",
        "type": "GPU"
      }
    ],
    "firelensConfiguration": {
      "type": "fluentd",
      "options": {"KeyName": ""}
    }
  },
  "volumes": [
    {
      "name": "",
      "host": {"sourcePath": ""},
      "configuredAtLaunch": true,
      "dockerVolumeConfiguration": {
        "scope": "task",
        "autoprovision": true,
        "driver": "",
        "driverOpts": {"KeyName": ""},
        "labels": {"KeyName": ""}
      },
      "efsVolumeConfiguration": {
        "fileSystemId": "",
        "rootDirectory": "",
        "transitEncryption": "ENABLED",
        "transitEncryptionPort": 0,
        "authorizationConfig": {
          "accessPointId": "",
          "iam": "ENABLED"
        }
      }
    }
  ],
  "requiresCompatibilities": ["FARGATE"],
  "cpu": "",

```

```
"memory": "",
"tags": [
  {
    "key": "",
    "value": ""
  }
],
"ephemeralStorage": {"sizeInGiB": 0},
"pidMode": "task",
"ipcMode": "none",
"proxyConfiguration": {
  "type": "APPMESH",
  "containerName": "",
  "properties": [
    {
      "name": "",
      "value": ""
    }
  ]
},
"inferenceAccelerators": [
  {
    "deviceName": "",
    "deviceType": ""
  }
]
}
```

Puoi generare questo modello di definizione di attività utilizzando il seguente comando AWS CLI .

```
aws ecs register-task-definition --generate-cli-skeleton
```

Esempi di definizioni di attività Amazon ECS

È possibile copiare gli esempi e gli snippet per iniziare a creare definizioni di attività personalizzate.

Puoi copiare gli esempi e incollarli quando utilizzi l'opzione Configura tramite JSON nella console. Assicurati di personalizzare gli esempi, ad esempio l'utilizzo del tuo ID account. Puoi includere i frammenti di codice nella definizione dell'attività JSON. Per ulteriori informazioni, consultare [Creazione di una definizione di attività Amazon ECS utilizzando la console](#) e [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#).

Per altri esempi di definizione delle attività, consulta [AWS Esempi di definizioni delle attività](#) su GitHub

Argomenti

- [Server Web](#)
- [splunkdriver di registro](#)
- [fluentddriver di registro](#)
- [gelfdriver di registro](#)
- [Carichi di lavoro su istanze esterne](#)
- [Immagine Amazon ECR e definizione delle attività, ruolo IAM](#)
- [Punto di ingresso con comando](#)
- [Dipendenze per i container](#)
- [Volumi nelle definizioni delle attività](#)
- [Definizioni di attività di esempio di Windows](#)

Server Web

Di seguito è riportato un esempio di definizione di attività usando container Linux sul tipo di avvio Fargate che configura un server Web:

```
{
  "containerDefinitions": [
    {
      "command": [
        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-foreground\""]
      ],
      "entryPoint": [
        "sh",
        "-c"
      ],
      "essential": true,
      "image": "public.ecr.aws/docker/library/httpd:2.4",
      "logConfiguration": {
```

```

        "logDriver": "awslogs",
        "options": {
            "awslogs-group" : "/ecs/fargate-task-definition",
            "awslogs-region": "us-east-1",
            "awslogs-stream-prefix": "ecs"
        }
    },
    "name": "sample-fargate-app",
    "portMappings": [
        {
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
        }
    ]
},
"cpu": "256",
"executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
"family": "fargate-task-definition",
"memory": "512",
"networkMode": "awsvpc",
"runtimePlatform": {
    "operatingSystemFamily": "LINUX"
},
"requiresCompatibilities": [
    "FARGATE"
]
}

```

Di seguito è riportato un esempio di definizione di attività usando container Windows sul tipo di avvio Fargate che configura un server Web:

```

{
    "containerDefinitions": [
        {
            "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html -Type file
-Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your
application is now running on a container in Amazon ECS.</p>'; C:\\ServiceMonitor.exe
w3svc"],
            "entryPoint": [

```

```

        "powershell",
        "-Command"
    ],
    "essential": true,
    "cpu": 2048,
    "memory": 4096,
    "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
    "name": "sample_windows_app",
    "portMappings": [
        {
            "hostPort": 80,
            "containerPort": 80,
            "protocol": "tcp"
        }
    ]
}
],
"memory": "4096",
"cpu": "2048",
"networkMode": "awsvpc",
"family": "windows-simple-iis-2019-core",
"executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
"runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
"requiresCompatibilities": ["FARGATE"]
}

```

splunkdriver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log splunk in una definizione di attività che invia i log a un servizio remoto. Il parametro di token Splunk è specificato come opzione segreta perché può essere trattato come dati sensibili. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

```

"containerDefinitions": [{
  "logConfiguration": {
    "logDriver": "splunk",
    "options": {
      "splunk-url": "https://cloud.splunk.com:8080",
      "tag": "tag_name",
    },
  },
  "secretOptions": [{
    "name": "splunk-token",
  }

```

```
"valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:splunk-token-KnrBkD"
}],
```

fluentddriver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log `fluentd` in una definizione di attività che invia i log a un servizio remoto. Il valore `fluentd-address` è specificato come opzione segreta perché può essere trattato come dati sensibili. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

```
"containerDefinitions": [{
  "logConfiguration": {
    "logDriver": "fluentd",
    "options": {
      "tag": "fluentd demo"
    },
    "secretOptions": [{
      "name": "fluentd-address",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:fluentd-address-KnrBkD"
    }]
  },
  "entryPoint": [],
  "portMappings": [{
    "hostPort": 80,
    "protocol": "tcp",
    "containerPort": 80
  },
  {
    "hostPort": 24224,
    "protocol": "tcp",
    "containerPort": 24224
  }]
}],
```

gelfdriver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log `gelf` in una definizione di attività che invia i log a un host remoto. Tale host esegue Logstash e accetta i log di Gelf come input. Per ulteriori informazioni, consulta [logConfiguration](#).

```
"containerDefinitions": [{
  "logConfiguration": {
    "logDriver": "gelf",
    "options": {
      "gelf-address": "udp://logstash-service-address:5000",
      "tag": "gelf task demo"
    }
  },
  "entryPoint": [],
  "portMappings": [{
    "hostPort": 5000,
    "protocol": "udp",
    "containerPort": 5000
  },
  {
    "hostPort": 5000,
    "protocol": "tcp",
    "containerPort": 5000
  }
]
}],
```

Carichi di lavoro su istanze esterne

Durante la registrazione di una definizione di attività di Amazon ECS, utilizza il parametro `requiresCompatibilities` e specifica `EXTERNAL` che verifica che la definizione di attività è compatibile per l'uso durante l'esecuzione di carichi di lavoro Amazon ECS sulle istanze esterne. Se registri una definizione di attività tramite la console, devi utilizzare l'editor JSON. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Important

Se i processi richiedono un ruolo IAM di esecuzione del processo, assicurati che sia specificato nella definizione di attività.

Quando distribuisce il tuo carico di lavoro, utilizza il tipo di avvio `EXTERNAL` durante la creazione del servizio o l'esecuzione del processo autonomo.

Di seguito è riportata una definizione di tabella di esempio.

Linux

```
{
  "requiresCompatibilities": [
    "EXTERNAL"
  ],
  "containerDefinitions": [{
    "name": "nginx",
    "image": "public.ecr.aws/nginx/nginx:latest",
    "memory": 256,
    "cpu": 256,
    "essential": true,
    "portMappings": [{
      "containerPort": 80,
      "hostPort": 8080,
      "protocol": "tcp"
    }]
  }],
  "networkMode": "bridge",
  "family": "nginx"
}
```

Windows

```
{
  "requiresCompatibilities": [
    "EXTERNAL"
  ],
  "containerDefinitions": [{
    "name": "windows-container",
    "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2019",
    "memory": 256,
    "cpu": 512,
    "essential": true,
    "portMappings": [{
      "containerPort": 80,
      "hostPort": 8080,
      "protocol": "tcp"
    }]
  }],
  "networkMode": "bridge",
  "family": "windows-container"
}
```


Immagine Amazon ECR e definizione delle attività, ruolo IAM

Il frammento di codice seguente utilizza un'immagine Amazon ECR denominata `aws-nodejs-sample` con il tag `v1` dal registro `123456789012.dkr.ecr.us-west-2.amazonaws.com`. Il container in questo processo eredita le autorizzazioni IAM dal ruolo `arn:aws:iam::123456789012:role/AmazonECSTaskS3BucketRole`. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

```
{
  "containerDefinitions": [
    {
      "name": "sample-app",
      "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/aws-nodejs-sample:v1",
      "memory": 200,
      "cpu": 10,
      "essential": true
    }
  ],
  "family": "example_task_3",
  "taskRoleArn": "arn:aws:iam::123456789012:role/AmazonECSTaskS3BucketRole"
}
```

Punto di ingresso con comando

Il frammento di codice seguente mostra la sintassi per un container Docker che utilizza un punto di ingresso e un argomento del comando. Questo container esegue il ping di `example.com` quattro volte e quindi si chiude.

```
{
  "containerDefinitions": [
    {
      "memory": 32,
      "essential": true,
      "entryPoint": ["ping"],
      "name": "alpine_ping",
      "readonlyRootFilesystem": true,
      "image": "alpine:3.4",
      "command": [
        "-c",
        "4",

```

```
        "example.com"
      ],
      "cpu": 16
    }
  ],
  "family": "example_task_2"
}
```

Dipendenze per i container

Questo frammento di codice mostra la sintassi di una definizione di attività con più container in cui è specificata la dipendenza per il container. Nella seguente definizione di attività, il container envoy deve raggiungere un stato di integrità, determinato dai parametri richiesti per il controllo dell'integrità, prima che il container app venga avviato. Per ulteriori informazioni, consulta [Dipendenze per i container](#).

```
{
  "family": "appmesh-gateway",
  "runtimePlatform": {
    "operatingSystemFamily": "LINUX"
  },
  "proxyConfiguration": {
    "type": "APPMESH",
    "containerName": "envoy",
    "properties": [
      {
        "name": "IgnoredUID",
        "value": "1337"
      },
      {
        "name": "ProxyIngressPort",
        "value": "15000"
      },
      {
        "name": "ProxyEgressPort",
        "value": "15001"
      },
      {
        "name": "AppPorts",
        "value": "9080"
      },
      {
```

```

        "name": "EgressIgnoredIPs",
        "value": "169.254.170.2,169.254.169.254"
    }
]
},
"containerDefinitions": [
{
    "name": "app",
    "image": "application_image",
    "portMappings": [
        {
            "containerPort": 9080,
            "hostPort": 9080,
            "protocol": "tcp"
        }
    ],
    "essential": true,
    "dependsOn": [
        {
            "containerName": "envoy",
            "condition": "HEALTHY"
        }
    ]
},
{
    "name": "envoy",
    "image": "840364872350.dkr.ecr.region-code.amazonaws.com/aws-appmesh-
envoy:v1.15.1.0-prod",
    "essential": true,
    "environment": [
        {
            "name": "APPMESH_VIRTUAL_NODE_NAME",
            "value": "mesh/meshName/virtualNode/virtualNodeName"
        },
        {
            "name": "ENVOY_LOG_LEVEL",
            "value": "info"
        }
    ],
    "healthCheck": {
        "command": [
            "CMD-SHELL",
            "echo hello"
        ]
    }
},

```

```
        "interval": 5,
        "timeout": 2,
        "retries": 3
    }
},
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
"networkMode": "awsvpc"
}
```

Volumi nelle definizioni delle attività

Usa quanto segue per capire come specificare i volumi nelle attività.

- Per informazioni su come configurare un volume Amazon EBS, consulta [Specificare la configurazione del volume Amazon EBS nella distribuzione di Amazon ECS](#).
- Per informazioni su come configurare un volume Amazon EFS, consulta [Configurazione dei file system Amazon EFS per Amazon ECS tramite la console](#).
- Per informazioni su come configurare un volume FSx per Windows File Server, consulta [Scopri come configurare FSx i file system Windows File Server per Amazon ECS](#).
- Per informazioni su come configurare un volume Docker, vedere [Esempi di volumi Docker per Amazon ECS](#).
- Per informazioni su come configurare un bind mount, vedere [Esempi di montaggio Bind per Amazon ECS](#).

Definizioni di attività di esempio di Windows

Di seguito è riportata una definizione di attività di esempio che consente di iniziare a utilizzare i container Windows su Amazon ECS.

Example Applicazione di esempio della console Amazon ECS per Windows

La seguente definizione di attività è l'applicazione di esempio della console Amazon ECS che viene prodotta nella procedura guidata per la prima esecuzione di Amazon ECS; è stata esportata per utilizzare l'immagine del container Windows microsoft/iis.

```
{
  "family": "windows-simple-iis",
```

```
"containerDefinitions": [  
  {  
    "name": "windows_sample_app",  
    "image": "mcr.microsoft.com/windows/servercore/iis",  
    "cpu": 1024,  
    "entryPoint":["powershell", "-Command"],  
    "command":["New-Item -Path C:\\\\inetpub\\\\wwwroot\\\\index.html -Type file -  
Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:  
40px; background-color: #333;} </style> </head><body> <div style=color:white;text-  
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your  
application is now running on a container in Amazon ECS.</p>'; C:\\\\ServiceMonitor.exe  
w3svc"],  
    "portMappings": [  
      {  
        "protocol": "tcp",  
        "containerPort": 80  
      }  
    ],  
    "memory": 1024,  
    "essential": true  
  }  
],  
  "networkMode": "awsvpc",  
  "memory": "1024",  
  "cpu": "1024"  
}
```

Cluster Amazon ECS

Un cluster Amazon ECS è un raggruppamento logico di processi o servizi. Oltre alle attività e ai servizi, un cluster è composto dalle seguenti risorse:

- La capacità dell'infrastruttura, che può essere una combinazione delle seguenti:
 - EC2 Istanze Amazon nel cloud AWS
 - Serverless (AWS Fargate) nel cloud AWS
 - Macchine virtuali (VM) o server on-premise
- La rete (VPC e sottorete) su cui vengono eseguite le attività e i servizi

Quando utilizzi EC2 istanze Amazon per la capacità, la sottorete può trovarsi in Availability Zones, Local Zones, Wavelength Zones o. AWS Outposts

- Uno spazio dei nomi facoltativo

Il namespace viene utilizzato per la service-to-service comunicazione con Service Connect.

- Un'opzione di monitoraggio

CloudWatch Container Insights ha un costo aggiuntivo ed è un servizio completamente gestito. Raccoglie, aggrega e riepiloga automaticamente parametri e log di Amazon ECS.

Di seguito sono elencati i concetti generali sui cluster Amazon ECS.

- I cluster vengono creati per separare le risorse.
- I cluster sono Regione AWS specifici.
- I cluster possono trovarsi in uno dei seguenti stati.

ACTIVE

Il cluster è pronto per accettare i processi e, se applicabile, puoi registrare le istanze del container con il cluster.

PROVISIONING

Al cluster sono associati provider di capacità e vengono create le risorse necessarie per il provider di capacità.

DEPROVISIONING

Al cluster sono associati provider di capacità e le risorse necessarie per il provider di capacità vengono eliminate.

Non riuscito

Al cluster sono associati provider di capacità e le risorse necessarie per il provider di capacità non sono state create.

INACTIVE

Il cluster è stato eliminato. I cluster con stato INACTIVE potrebbero rimanere individuabili nel tuo account per un periodo di tempo. Questo comportamento è soggetto a modifiche future, quindi assicurati di non fare affidamento sulla persistenza INACTIVE dei cluster.

- È possibile utilizzare diversi tipi di istanza per il tipo di EC2 avvio o per i provider di capacità del gruppo Auto Scaling. Un'istanza può essere registrata solo su un cluster alla volta.
- Puoi limitare l'accesso ai cluster creando policy IAM personalizzate. Per informazioni, consulta la [Esempi di cluster Amazon ECS](#) sezione in [Esempi di policy basate su identità per Amazon Elastic Container Service](#).
- È possibile utilizzare Service Auto Scaling per scalare le attività di Fargate. Per ulteriori informazioni, consulta [Ridimensiona automaticamente il tuo servizio Amazon ECS](#).
- È possibile configurare uno spazio dei nomi Service Connect predefinito per un cluster. Dopo aver impostato uno spazio dei nomi predefinito di Service Connect, tutti i nuovi servizi con Service Connect attivato creati nel cluster saranno aggiunti come servizi client nello spazio dei nomi. e non sono necessarie ulteriori configurazioni. Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

Provider di capacità

I provider di capacità Amazon ECS gestiscono il dimensionamento dell'infrastruttura per le attività nei cluster. Ogni cluster ha uno o più provider di capacità e una strategia di provider di capacità facoltativa. È possibile assegnare una strategia di provider di capacità predefinita al cluster. La strategia del provider di capacità determina il modo in cui le attività vengono distribuite tra i provider di capacità del cluster. Quando si esegue un'attività autonoma o si crea un servizio, si utilizza la strategia predefinita del provider di capacità del cluster o una strategia del fornitore di capacità che sostituisce quella predefinita. La strategia predefinita del provider di capacità del cluster si applica

solo quando non si specifica un tipo di avvio o una strategia del fornitore di capacità per l'attività o il servizio. Se si fornisce uno di questi parametri, la strategia predefinita non viene utilizzata.

Per Fargate, non è necessario creare o gestire la capacità. È sufficiente associare uno dei seguenti fornitori di capacità predefiniti al cluster:

- Fargate
- Fargate Spot

Quando utilizzi EC2 le istanze per la tua capacità, utilizzi il gruppo Auto Scaling per gestire le EC2 istanze. L'Auto Scaling aiuta a garantire il numero corretto di EC2 istanze disponibili per gestire il carico dell'applicazione.

Un cluster può contenere una combinazione di attività ospitate su AWS Fargate istanze Amazon o EC2 istanze esterne. Le attività possono essere eseguite su Fargate o sull' EC2 infrastruttura come tipo di lancio o strategia di capacity provider. Se utilizzi EC2 come tipo di lancio, Amazon ECS non monitora e non ridimensiona la capacità dei gruppi Amazon EC2 Auto Scaling. Per ulteriori informazioni sui tipi di avvio, consulta [Tipi di avvio di Amazon ECS](#).

Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Una strategia per i fornitori di capacità può contenere solo i fornitori di capacità del gruppo Auto Scaling o i fornitori di capacità Fargate.

Cluster Amazon ECS per Fargate

I provider di capacità Amazon ECS gestiscono il dimensionamento dell'infrastruttura per le attività nei cluster. Ogni cluster ha uno o più provider di capacità e una strategia di provider di capacità facoltativa. La strategia del provider di capacità determina il modo in cui le attività vengono distribuite tra i provider di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del provider di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster.

Quando si eseguono le attività AWS Fargate, non è necessario creare o gestire la capacità. È sufficiente associare uno dei seguenti provider di capacità predefiniti al cluster:

- Fargate
- Fargate Spot

Con Amazon ECS on AWS Fargate capacity provider, puoi utilizzare sia la capacità di Fargate che quella di Fargate Spot per le tue attività Amazon ECS.

Con Fargate Spot è possibile eseguire le attività di Amazon ECS con tolleranza alle interruzioni a una tariffa scontata rispetto al prezzo di Fargate. Fargate Spot esegue le attività nella capacità di elaborazione di riserva. Quando è AWS necessario ripristinare la capacità, le attività vengono interrotte con un avviso di due minuti.

Quando le attività che utilizzano i provider di capacità Fargate e Fargate Spot vengono interrotte, l'evento di modifica dello stato dell'attività viene inviato ad Amazon. EventBridge Il motivo dell'interruzione descrive la causa. Per ulteriori informazioni, consulta [Eventi di modifica dello stato delle attività di Amazon ECS](#).

Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e di Fargate. Tuttavia, una strategia di provider di capacità può contenere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi. Per ulteriori informazioni, consulta [Auto Scaling Group Capacity Provider](#).

Quando si utilizzano provider di capacità:

- È necessario associare un provider di capacità a un cluster prima di associarlo alla strategia del fornitore di capacità.
- È possibile specificare un massimo di 20 provider di capacità per una strategia di provider di capacità.
- Non è possibile aggiornare un servizio che utilizza un provider di capacità di un gruppo con scalabilità automatica per utilizzare un provider di capacità Fargate. È vero anche il contrario.
- In una strategia del provider di capacità, se non viene specificato alcun valore di `weight` per un provider di capacità nella console, allora viene utilizzato il valore predefinito 1. Se si utilizza l'API o AWS CLI, 0 viene utilizzato il valore predefinito di.
- Quando più provider di capacità sono specificati nell'ambito di una strategia di provider di capacità, almeno uno dei provider deve avere un valore di peso maggiore di zero. I provider di capacità con un peso pari a zero non vengono utilizzati per collocare le attività. Se specifichi più provider di capacità in una strategia tutti con un peso pari a zero, allora qualsiasi operazione `RunTask` o `CreateService` che utilizza la strategia del provider di capacità avrà esito negativo.
- In una strategia di provider di capacità, solo un provider di capacità può avere un valore di base definito. Se non viene specificato alcun valore, viene utilizzato il valore predefinito zero.

- Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Tuttavia, una strategia di provider di capacità può includere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi.
- Un cluster può contenere una combinazione di servizi e attività autonome che utilizzano sia i provider di capacità che i tipi di avvio. Un servizio può essere aggiornato per utilizzare una strategia del provider di capacità anziché un tipo di avvio. Tuttavia, quando si esegue questa operazione è necessario forzare una nuova implementazione.

Avvisi di risoluzione di Fargate Spot

Nei periodi con domanda estremamente elevata, la capacità Fargate spot potrebbe non essere disponibile. Ciò può causare ritardi nelle attività di Fargate Spot. Quando ciò accade, i servizi Amazon ECS riprovano ad avviare le attività finché non diventa disponibile la capacità richiesta. Fargate non sostituisce la capacità spot con quella on demand.

Quando le attività che utilizzano la capacità Fargate Spot vengono interrotte a causa di un'interruzione Spot, viene inviato un avviso due minuti prima dell'arresto del processo. L'avviso viene inviato come evento di modifica dello stato dell'attività ad Amazon EventBridge e come segnale SIGTERM all'attività in esecuzione. Quando si utilizza Fargate Spot come parte di un servizio, lo scheduler del servizio riceve il segnale di interruzione e prova ad avviare altre attività su Fargate Spot se c'è capacità disponibile. Un servizio con una sola attività verrà interrotto fino a quando la capacità non sarà disponibile. Per ulteriori informazioni su un arresto corretto, consulta [Arresto regolare con ECS](#).

Per assicurarti che i container si chiudano correttamente prima che l'attività si interrompa, puoi configurare le seguenti impostazioni:

- È possibile specificare un valore `stopTimeout` di 120 secondi al massimo nella definizione del container utilizzata dall'attività. Il valore di `stopTimeout` predefinito è 30 secondi. È possibile specificare un valore `stopTimeout` più lungo per avere più tempo tra il momento in cui viene ricevuto l'evento di modifica dello stato delle attività e il momento in cui viene forzato l'arresto del container. Per ulteriori informazioni, consulta [Timeout del container](#).
- Il segnale SIGTERM deve essere ricevuto dall'interno del container per eseguire qualsiasi operazione di pulizia. La mancata elaborazione di questo segnale comporterà la ricezione di un segnale SIGKILL dopo la configurazione di `stopTimeout` e può comportare la perdita o il danneggiamento dei dati.

Di seguito è riportato un frammento di un evento di modifica dello stato dell'attività. Questo frammento visualizza il motivo dell'arresto e il codice di arresto per un'interruzione di Fargate Spot.

```
{
  "version": "0",
  "id": "9bcdac79-b31f-4d3d-9410-fbd727c29fab",
  "detail-type": "ECS Task State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "resources": [
    "arn:aws:ecs:us-east-1:111122223333:task/b99d40b3-5176-4f71-9a52-9dbd6f1cebef"
  ],
  "detail": {
    "clusterArn": "arn:aws:ecs:us-east-1:111122223333:cluster/default",
    "createdAt": "2016-12-06T16:41:05.702Z",
    "desiredStatus": "STOPPED",
    "lastStatus": "RUNNING",
    "stoppedReason": "Your Spot Task was interrupted.",
    "stopCode": "SpotInterruption",
    "taskArn": "arn:aws:ecs:us-east-1:111122223333:task/
b99d40b3-5176-4f71-9a52-9dbd6fEXAMPLE",
    ...
  }
}
```

Di seguito è riportato uno schema di eventi utilizzato per creare una EventBridge regola per gli eventi di modifica dello stato delle attività di Amazon ECS. Facoltativamente, puoi specificare un cluster nel campo `detail`. Ciò significa che per quel cluster riceverai eventi di modifica dello stato delle attività. Per ulteriori informazioni sulla creazione di una EventBridge regola, consulta la sezione Guida [introduttiva ad Amazon EventBridge](#) nella Amazon EventBridge User Guide.

```
{
  "source": [
    "aws.ecs"
  ],
  "detail-type": [
    "ECS Task State Change"
  ],
  "detail": {
    "clusterArn": [
      "arn:aws:ecs:us-west-2:111122223333:cluster/default"
    ]
  }
}
```

```
}  
}
```

Creazione di un cluster Amazon ECS per il tipo di lancio Fargate

Crei un cluster per definire l'infrastruttura su cui vengono eseguiti i tuoi compiti e i tuoi servizi.

Prima di iniziare, accertati di aver completato le fasi in [Configurazione per l'uso di Amazon ECS](#) e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#). La console Amazon ECS crea le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

La console associa automaticamente i provider di capacità Fargate e Fargate Spot al cluster.

È possibile modificare le seguenti opzioni:

- Aggiungi uno spazio dei nomi al cluster.

Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnetti i servizi Amazon ECS](#).

- Attiva Container Insights con una migliore osservabilità o Container Insights.

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta [the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata”](#).

Il 2 dicembre 2024, AWS ha rilasciato Container Insights con osservabilità migliorata per Amazon ECS. Questa versione supporta l'osservabilità migliorata per i cluster Amazon ECS utilizzando i tipi di lancio Amazon e EC2 Fargate. Dopo aver configurato Container Insights con una migliore osservabilità su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura dal livello del cluster fino al livello del contenitore nel tuo ambiente e visualizza i tuoi dati in dashboard che mostrano una varietà di metriche e dimensioni. Puoi quindi utilizzare questi out-of-the-box dashboard sulla console Container Insights per comprendere meglio lo stato e le prestazioni dei container e mitigare i problemi più rapidamente identificando le anomalie.

Ti consigliamo di utilizzare Container Insights con una migliore osservabilità anziché Container Insights perché offre una visibilità dettagliata nell'ambiente dei container, riducendo il tempo medio di risoluzione.

- Aggiungi tag per facilitare l'identificazione del cluster.
- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Creare una chiave KMS nella Guida per l'AWS Key Management Service](#) utente.
- Assegna una AWS KMS chiave per la tua memoria effimera Fargate. Per informazioni su come creare una chiave, consulta [Creare una chiave KMS](#) nella Guida per l'utente AWS Key Management Service

Procedura

Creazione di un nuovo cluster (console Amazon ECS)

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Configurazione del cluster, configura gli elementi seguenti:
 - In Nome cluster, inserisci un nome univoco.

Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.
 - (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. (Facoltativo) Usa Container Insights, espandi Monitoring, quindi scegli una delle seguenti opzioni:
 - Per utilizzare la soluzione consigliata di Container Insights con osservabilità migliorata, scegli Container Insights con osservabilità migliorata.
 - Per utilizzare Container Insights, scegli Container Insights.
7. (Facoltativo), in Encryption, puoi configurare quanto segue:

- Crittografa i tuoi dati sullo storage temporaneo Fargate. In Crittografia, per l'archiviazione effimera Fargate, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati di archiviazione effimera Fargate.
 - Crittografa i dati sullo storage gestito. In Crittografia, per Storage gestito, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati di storage gestito.
8. (Facoltativo) Per identificare il tuo cluster, espandi la sezione Tags (Tag), quindi configura i tag.
- [Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:
- In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
- [Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.
9. Scegli Create (Crea).

Passaggi successivi

Dopo aver creato il cluster, è possibile creare le definizioni delle attività per le applicazioni e quindi eseguirle come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Definizioni dei processi di Amazon ECS](#)
- [Esecuzione di un'applicazione come attività Amazon ECS](#)
- [Creazione di un servizio Amazon ECS utilizzando la console](#)

Provider di capacità Amazon ECS per il tipo di EC2 lancio

Quando utilizzi EC2 le istanze Amazon per la tua capacità, utilizzi i gruppi di Auto Scaling per gestire le istanze EC2 Amazon registrate nei rispettivi cluster. Auto Scaling aiuta a garantire il numero corretto di EC2 istanze Amazon disponibili per gestire il carico dell'applicazione.

Puoi utilizzare la funzionalità di scalabilità gestita per consentire ad Amazon ECS di gestire le azioni di scalabilità in e orizzontale del gruppo Auto Scaling oppure puoi gestire tu stesso le azioni di scalabilità. Per ulteriori informazioni, consulta [Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster](#).

Ti consigliamo di creare un nuovo gruppo Auto Scaling vuoto. Se utilizzi un gruppo Auto Scaling esistente, tutte EC2 le istanze Amazon associate al gruppo che erano già in esecuzione e registrate in un cluster Amazon ECS prima che il gruppo Auto Scaling venisse utilizzato per creare un fornitore di capacità potrebbero non essere registrate correttamente presso il provider di capacità. Ciò può causare problemi quando si utilizza il provider di capacità in una strategia di provider di capacità. Utilizza `DescribeContainerInstances` per verificare se un'istanza di container è associata o meno a un provider di capacità.

Note

Per creare un gruppo Auto Scaling vuoto, imposta il conteggio desiderato su zero. Dopo aver creato il provider di capacità e averlo associato a un cluster, potrai aumentarlo.

Quando usi la console Amazon ECS, Amazon ECS crea un modello di EC2 lancio Amazon e un gruppo Auto Scaling per tuo conto come parte dello stack. AWS CloudFormation Sono preceduti da. `EC2ContainerService-<ClusterName>` Puoi utilizzare il gruppo con dimensionamento automatico come provider di capacità per tale cluster.

Ti consigliamo di utilizzare il drenaggio gestito delle istanze per consentire la chiusura graduale delle istanze EC2 Amazon senza interrompere i carichi di lavoro. Questa funzionalità è attiva per impostazione predefinita. Per ulteriori informazioni, consulta [Blocca in sicurezza i carichi di lavoro Amazon ECS in esecuzione sulle istanze EC2](#)

Quando utilizzi i provider di capacità del gruppo con dimensionamento automatico nella console, è opportuno considerare quanto segue:

- Un gruppo Auto Scaling deve avere un valore `MaxSize` maggiore di zero per l'aumento orizzontale.
- Il gruppo Auto Scaling non può avere impostazioni di ponderazione delle istanze.
- Se il gruppo con dimensionamento automatico non è in grado di impiegare la scalabilità orizzontale per adattarsi al numero di esecuzioni di attività, le attività non riusciranno ad andare oltre lo stato `PROVISIONING`.
- Non modificare la risorsa della policy di scalabilità associata ai gruppi con scalabilità automatica gestiti dai provider di capacità.
- Se il dimensionamento gestito è attivato quando crei un provider di capacità, puoi impostare il conteggio per il gruppo con scalabilità automatica desiderato su `0`. Quando il dimensionamento

gestito è attivato, Amazon ECS gestisce le operazioni di riduzione e aumento del gruppo con scalabilità automatica.

- È necessario associare il provider di capacità a un cluster prima di associarlo alla strategia del fornitore di capacità.
- È possibile specificare un massimo di 20 provider di capacità per una strategia di provider di capacità.
- Non è possibile aggiornare un servizio che utilizza un provider di capacità di un gruppo con scalabilità automatica per utilizzare un provider di capacità Fargate. È vero anche il contrario.
- In una strategia del provider di capacità, se non viene specificato alcun valore di `weight` per un provider di capacità nella console, allora viene utilizzato il valore predefinito 1. Se si utilizza l'API o AWS CLI, 0 viene utilizzato il valore predefinito di.
- Quando più provider di capacità sono specificati nell'ambito di una strategia di provider di capacità, almeno uno dei provider deve avere un valore di peso maggiore di zero. I provider di capacità con un peso pari a zero non vengono utilizzati per collocare le attività. Se specifichi più provider di capacità in una strategia tutti con un peso pari a zero, allora qualsiasi operazione `RunTask` o `CreateService` che utilizza la strategia del provider di capacità avrà esito negativo.
- In una strategia di provider di capacità, solo un provider di capacità può avere un valore di base definito. Se non viene specificato alcun valore, viene utilizzato il valore predefinito zero.
- Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Tuttavia, una strategia di provider di capacità può includere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi.
- Un cluster può contenere una combinazione di servizi e attività autonome che utilizzano sia i provider di capacità che i tipi di avvio. Un servizio può essere aggiornato per utilizzare una strategia del provider di capacità anziché un tipo di avvio. Tuttavia, quando si esegue questa operazione è necessario forzare una nuova implementazione.
- Amazon ECS supporta i pool caldi di Amazon EC2 Auto Scaling. Un pool caldo è un gruppo di EC2 istanze Amazon preinizializzate pronte per essere messe in servizio. Ogni volta che l'applicazione deve essere scalata orizzontalmente, Amazon EC2 Auto Scaling utilizza le istanze preinizializzate del pool caldo anziché lanciare istanze fredde. Ciò consente l'esecuzione di qualsiasi processo di inizializzazione finale prima che l'istanza venga messa in servizio. Per ulteriori informazioni, consulta [Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling](#).

Per ulteriori informazioni sulla creazione di un modello di lancio di Amazon EC2 Auto Scaling, consulta i modelli di lancio [di Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide. Per ulteriori informazioni sulla creazione di un gruppo Amazon EC2 Auto Scaling, consulta i gruppi di Auto [Scaling nella Amazon Auto Scaling](#) User EC2 Guide.

Considerazioni sulla sicurezza delle istanze di EC2 container Amazon per Amazon ECS

È consigliabile prendere in considerazione una singola istanza di container e il relativo accesso all'interno del modello di minaccia. Ad esempio, una singola attività interessata potrebbe essere in grado di sfruttare le autorizzazioni IAM di un'attività non infetta sulla stessa istanza.

Per impedirlo, consigliamo di utilizzare la procedura seguente:

- Non utilizzare i privilegi di amministratore quando esegui le attività.
- Assegna un ruolo di attività con accesso meno privilegiato alle attività.

L'agente di container crea in automatico un token con un ID di credenziali univoco che viene utilizzato per accedere alle risorse Amazon ECS.

- Per impedire ai contenitori eseguiti da attività che utilizzano la modalità di aws vpc rete di accedere alle informazioni sulle credenziali fornite al profilo dell' EC2 istanza Amazon, pur consentendo le autorizzazioni fornite dal ruolo dell'attività, imposta la variabile di configurazione dell'ECS_AWSVPC_BLOCK_IMDS agente su true nel file di configurazione dell'agente e riavvia l'agente.
- Usa Amazon GuardDuty Runtime Monitoring per rilevare le minacce per cluster e contenitori all'interno del tuo AWS ambiente. Runtime Monitoring utilizza un agente di GuardDuty sicurezza che aggiunge visibilità di runtime ai singoli carichi di lavoro Amazon ECS, ad esempio accesso ai file, esecuzione dei processi e connessioni di rete. Per ulteriori informazioni, consulta [GuardDutyRuntime Monitoring](#) nella Guida per l'GuardDuty utente.

Creazione di un cluster Amazon ECS per il tipo di EC2 lancio Amazon

Crei un cluster per definire l'infrastruttura su cui vengono eseguiti i tuoi compiti e i tuoi servizi.

Prima di iniziare, accertati di aver completato le fasi in [Configurazione per l'uso di Amazon ECS](#) e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#). La console Amazon ECS offre un modo semplice per creare le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

Per rendere il processo di creazione del cluster il più semplice possibile, la console dispone di selezioni predefinite per molte scelte che descriviamo di seguito. Ci sono anche pannelli di aiuto disponibili per la maggior parte delle sezioni della console che forniscono ulteriore contesto.

Puoi registrare EC2 istanze Amazon quando crei il cluster o registrare istanze aggiuntive con il cluster dopo che è stato creato.

È possibile modificare le seguenti opzioni di default:

- Modifica le sottoreti in cui vengono avviate le istanze
- Modifica i gruppi di sicurezza utilizzati per controllare il traffico verso le istanze di container
- Aggiungi uno spazio dei nomi al cluster.

Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnetti i servizi Amazon ECS](#).

- Attiva Container Insights con una migliore osservabilità o Container Insights.

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta [the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata”](#).

Il 2 dicembre 2024, AWS ha rilasciato Container Insights con osservabilità migliorata per Amazon ECS. Questa versione supporta l'osservabilità migliorata per i cluster Amazon ECS utilizzando i tipi di lancio Amazon e EC2 Fargate. Dopo aver configurato Container Insights con una migliore osservabilità su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura dal livello del cluster fino al livello del contenitore nel tuo ambiente e visualizza i tuoi dati in dashboard che mostrano una varietà di metriche e dimensioni. Puoi quindi utilizzare questi out-of-the-box dashboard sulla console Container Insights per comprendere meglio lo stato e le prestazioni dei container e mitigare i problemi più rapidamente identificando le anomalie.

Ti consigliamo di utilizzare Container Insights con una migliore osservabilità anziché Container Insights perché offre una visibilità dettagliata nell'ambiente dei container, riducendo il tempo medio di risoluzione.

- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Creare una chiave KMS nella Guida per l'AWS Key Management Service](#) utente.
- Assegna una AWS KMS chiave per la tua memoria effimera Fargate. Per informazioni su come creare una chiave, consulta [Creare una chiave KMS](#) nella Guida per l'utente AWS Key Management Service
- Aggiungi tag per facilitare l'identificazione del cluster.

Opzioni del gruppo Auto Scaling

Quando utilizzi EC2 istanze Amazon, devi specificare un gruppo Auto Scaling per gestire l'infrastruttura su cui vengono eseguiti i tuoi compiti e i tuoi servizi.

Quando scegli di creare un nuovo gruppo Auto Scaling, questo viene configurato automaticamente per il seguente comportamento:

- Amazon ECS gestisce le operazioni di riduzione e incremento orizzontale del gruppo Auto Scaling.
- Amazon ECS non impedirà che EC2 le istanze Amazon che contengono attività e che fanno parte di un gruppo Auto Scaling vengano terminate durante un'azione di scale-in. Per ulteriori informazioni, consulta [Protezione delle istanze](#) nella Guida per l'utente di AWS Auto Scaling .

Puoi configurare le seguenti proprietà del gruppo Auto Scaling che determinano il tipo e il numero di istanze da avviare per il gruppo:

- Le AMI ottimizzate per Amazon ECS.
- Il tipo di istanza.
- La coppia di chiavi SSH che dimostra la tua identità quando ti connetti all'istanza. Per informazioni su come creare chiavi SSH, consulta le [coppie di EC2 chiavi Amazon e le istanze Linux](#) nella Amazon EC2 User Guide.
- Il numero minimo di istanze da avviare per il gruppo Auto Scaling.
- Il numero massimo di istanze avviate per il gruppo Auto Scaling.

Affinché il gruppo venga aumentato orizzontalmente, il massimo deve essere superiore a 0.

Amazon ECS crea un modello di avvio di Amazon EC2 Auto Scaling e un gruppo Auto Scaling per tuo conto come parte dello stack. AWS CloudFormation I valori specificati per l'AMI, i tipi di istanza e la coppia di chiavi SSH fanno parte del modello di avvio. Viene utilizzato il prefisso

EC2ContainerService-<*ClusterName*> per identificare i modelli con facilità. I gruppi Auto Scaling hanno il prefisso <*ClusterName*>-ECS-Infra-ECSAutoScalingGroup.

Le istanze avviate per il gruppo Auto Scaling utilizzano il modello di avvio.

Opzioni di rete

Per impostazione predefinita, le istanze vengono avviate nelle sottoreti predefinite per la regione. Vengono utilizzati i gruppi di sicurezza, che controllano il traffico verso le istanze dei container, attualmente associati alle sottoreti. Puoi modificare le sottoreti e i gruppi di sicurezza per le istanze.

È possibile scegliere una sottorete esistente. Puoi utilizzare un gruppo di sicurezza esistente o crearne uno nuovo. Se si crea un nuovo gruppo di sicurezza, è necessario specificare almeno una regola in entrata.

Le regole in entrata determinano il traffico che può raggiungere le istanze di container e includono le seguenti proprietà:

- Il protocollo da consentire
- L'intervallo di porte da autorizzare
- Il traffico in entrata (fonte)

Per consentire il traffico in entrata da un indirizzo o da un blocco CIDR specifico, usa Personalizzato per Origine con il CIDR consentito.

Per consentire il traffico in entrata da tutte le destinazioni, specifica Ovunque per Origine. Questo aggiunge automaticamente il blocco CIDR 0.0.0.0/0 e il blocco IPv4 CIDR: :/0. IPv6

Per consentire il traffico in entrata dal computer locale, usa Gruppo di origine per Origine. Questa operazione aggiunge automaticamente l'indirizzo IP attuale del computer locale come origine consentita.

Creazione di un nuovo cluster (console Amazon ECS)

Prima di iniziare, assegna l'autorizzazione IAM appropriata. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#).

1. Apri <https://console.aws.amazon.com/ecs/la console> alla v2.
2. Seleziona la Regione da utilizzare nella barra di navigazione.

3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Configurazione del cluster, configura gli elementi seguenti:

- In Nome cluster, inserisci un nome univoco.

Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.

- (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. Aggiungi EC2 istanze Amazon al tuo cluster, espandi l'infrastruttura e quindi seleziona le EC2 istanze Amazon.

Successivamente, configura il gruppo Auto Scaling che funge da provider di capacità:

- a. Per utilizzare un gruppo Auto Scaling esistente, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona il gruppo.
- b. Per creare un gruppo Auto Scaling, da Auto Scaling group (ASG) (Gruppo di Auto Scaling (ASG)), seleziona Create new group (Crea nuovo gruppo) e quindi fornisci i seguenti dettagli sul gruppo:
 - Per il modello di provisioning, scegli se utilizzare istanze On-demand o istanze Spot.
 - Se scegli di utilizzare le istanze Spot, per la strategia di allocazione scegli quali pool di capacità Spot (tipi di istanze e zone di disponibilità) vengono utilizzati per le istanze.

Per la maggior parte dei carichi di lavoro, puoi scegliere Price capacity optimized.

Per ulteriori informazioni, consulta [Strategie di allocazione per istanze Spot](#) nella Amazon EC2 User Guide.

- Per l'istanza Container Amazon Machine Image (AMI), scegli l'AMI ottimizzata per Amazon ECS per le istanze di gruppo Auto Scaling.
- Ad EC2 esempio, scegli il tipo di istanza per i tuoi carichi di lavoro.

La scalabilità gestita funziona meglio se il gruppo Auto Scaling utilizza tipi di istanza uguali o simili.

- Ad EC2 esempio, scegli un ruolo di istanza del contenitore esistente oppure puoi crearne uno nuovo.

Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

- In Capacity (Capacità), inserisci il numero minimo e massimo di istanze da avviare nel gruppo Auto Scaling.
- In SSH key pair (Coppia di chiavi SSH), scegli la coppia che dimostra la tua identità quando ti connetti all'istanza.
- Per consentire immagini e spazio di archiviazione più grandi, per la dimensione del volume Root EBS, inserisci il valore in GiB.

7. (Facoltativo) Per modificare il VPC e le sottoreti, in Networking for Amazon EC2 instances, esegui una delle seguenti operazioni:

- Per rimuovere una sottorete, in Subnets (Sottoreti), scegli X per ogni sottorete da rimuovere.
- Per passare a un VPC diverso da quello di default, in VPC, scegli un VPC esistente, poi in Sottoreti, seleziona le sottoreti.
- Scegli i gruppi di sicurezza. In Gruppo di sicurezza, scegli una delle seguenti opzioni:
 - Per utilizzare un gruppo di sicurezza esistente, scegli Usa un gruppo di sicurezza esistente, quindi selezionalo.
 - Per creare un nuovo gruppo di sicurezza, scegli Crea un nuovo gruppo di sicurezza. Scegli quindi Aggiungi regola per ogni regola in entrata.

Per informazioni sulle regole in entrata, consulta [Opzioni di rete](#).

- Per assegnare automaticamente indirizzi IP pubblici alle istanze di EC2 container Amazon, per Assegnare automaticamente un IP pubblico, scegli una delle seguenti opzioni:
 - Usa impostazione della sottorete: assegna un indirizzo IP pubblico alle istanze quando la sottorete in cui le istanze vengono avviate è una sottorete pubblica.
 - Attiva: assegna un indirizzo IP pubblico alle istanze.

8. (Facoltativo) Usa Container Insights, espandi Monitoring, quindi scegli una delle seguenti opzioni:

- Per utilizzare la soluzione consigliata di Container Insights con osservabilità migliorata, scegli Container Insights con osservabilità migliorata.
- Per utilizzare Container Insights, scegli Container Insights.

9. (Facoltativo)

Se utilizzi Runtime Monitoring con l'opzione manuale e desideri che questo cluster venga monitorato da GuardDuty, scegli Aggiungi tag ed esegui le seguenti operazioni:

- Per Key, inserisci **guardDutyRuntimeMonitoringManaged**
 - In Valore, specifica **true**.
10. (Facoltativo) Crittografa i dati sullo storage gestito. In Crittografia, per Storage gestito, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati di storage gestito.
11. (Facoltativo) Per gestire i tag cluster, espandi Tags (Tag), quindi esegui una delle seguenti operazioni:

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:

- In Chiave, immetti il nome della chiave.
- In Valore, immetti il valore della chiave.

[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.

12. Scegli Create (Crea).

Passaggi successivi

Dopo aver creato il cluster, è possibile creare le definizioni delle attività per le applicazioni e quindi eseguirle come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Definizioni dei processi di Amazon ECS](#)
- [Esecuzione di un'applicazione come attività Amazon ECS](#)
- [Creazione di un servizio Amazon ECS utilizzando la console](#)

Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster

Amazon ECS può gestire la scalabilità delle EC2 istanze Amazon registrate nel tuo cluster. Questa operazione viene definita dimensionamento automatico del cluster Amazon ECS. Attivi la scalabilità gestita quando crei il provider di capacità di gruppo Amazon ECS Auto Scaling. Quindi, si imposta una percentuale target (`targetCapacity`) per l'utilizzo dell'istanza in questo gruppo Auto Scaling. Amazon ECS crea due CloudWatch metriche personalizzate e una politica di scalabilità

di tracciamento mirata per il tuo gruppo Auto Scaling. Amazon ECS gestisce quindi le azioni di scalabilità verticale e orizzontale in base all'utilizzo delle risorse utilizzato dalle attività.

Per ogni provider di capacità del gruppo con scalabilità automatica associato a un cluster, Amazon ECS crea e gestisce le seguenti risorse.

- Un allarme con valore metrico basso CloudWatch
- Un allarme con valore metrico elevato CloudWatch
- Una policy di scalabilità di monitoraggio dei target

Note

Amazon ECS crea la policy di scalabilità di monitoraggio dei target e la collega al gruppo con scalabilità automatica. Per aggiornare la policy di scalabilità di monitoraggio delle destinazioni, aggiornare le impostazioni di scalabilità gestita della policy e non aggiornare direttamente la policy di scalabilità.

Quando disattivi la scalabilità gestita o dissocii il fornitore di capacità da un cluster, Amazon ECS rimuove sia i CloudWatch parametri che le risorse della policy di scalabilità di tracciamento di destinazione.

Amazon ECS utilizza i seguenti parametri per determinare le azioni da intraprendere:

CapacityProviderReservation

La percentuale di istanze di container utilizzate per uno specifico fornitore di capacità. Questo parametro è generato da Amazon ECS.

Amazon ECS imposta il valore CapacityProviderReservation su un numero compreso tra 0 e 100. Amazon ECS utilizza la seguente formula per rappresentare il rapporto della capacità rimanente nel gruppo con dimensionamento automatico. Quindi, Amazon ECS pubblica la metrica su CloudWatch. Per ulteriori informazioni su come viene calcolata la metrica, consulta [Deep Dive on Amazon ECS Cluster Auto Scaling](#).

```
CapacityProviderReservation = (number of instances needed) / (number of running instances) x 100
```


DesiredCapacity

La quantità di capacità del gruppo con dimensionamento automatico. Questa metrica non è pubblicata su CloudWatch.

Amazon ECS pubblica la `CapacityProviderReservation` metrica CloudWatch nel namespace `AWS/ECS/ManagedScaling`. Il parametro `CapacityProviderReservation` provoca una delle seguenti operazioni:

Il valore **`CapacityProviderReservation`** è uguale a **`targetCapacity`**

Il gruppo con dimensionamento automatico non ha bisogno di essere ridimensionato o di impiegare la scalabilità orizzontale. La percentuale di utilizzo prevista è stata raggiunta.

Il valore **`CapacityProviderReservation`** è maggiore di **`targetCapacity`**

Ci sono più attività che utilizzano una percentuale di capacità più elevata rispetto alla percentuale `targetCapacity`. L'aumento del valore della `CapacityProviderReservation` metrica provoca l'attivazione dell'allarme associato. CloudWatch. Questo allarme aggiorna il valore di `DesiredCapacity` per il gruppo Auto Scaling. Il gruppo Auto Scaling utilizza questo valore per avviare EC2 le istanze e quindi registrarle nel cluster.

Se `targetCapacity` è il valore predefinito del 100%, le nuove attività rimangono nello stato `PENDING` durante l'impiego della scalabilità orizzontale poiché non c'è capacità disponibile sulle istanze per eseguire le attività. Dopo la registrazione delle nuove istanze con ECS, queste attività verranno avviate sulle nuove istanze.

Il valore **`CapacityProviderReservation`** è inferiore a **`targetCapacity`**

Ci sono meno attività che utilizzano una percentuale di capacità inferiore rispetto alla percentuale `targetCapacity` ed è presente almeno un'istanza che può essere terminata. La diminuzione del valore della `CapacityProviderReservation` metrica provoca l'attivazione dell'allarme associato. Questo allarme aggiorna il valore di `DesiredCapacity` per il gruppo Auto Scaling. Il gruppo Auto Scaling utilizza questo valore per terminare le istanze dei EC2 container e quindi annullarle dal cluster.

Il gruppo con dimensionamento automatico utilizza le policy di terminazione, per determinare quali istanze terminare per prime durante gli eventi di ridimensionamento. Inoltre, evita le istanze con l'impostazione di protezione da ridimensionamento delle istanze attivata. Il dimensionamento automatico del cluster può gestire le istanze con l'impostazione di protezione

per il ridimensionamento se la protezione da terminazione gestita è attivata. Per ulteriori informazioni sulla protezione da terminazione gestita, consulta [Controlla le istanze terminate da Amazon ECS](#). Per ulteriori informazioni su come i gruppi di Auto Scaling terminano le istanze, consulta [Controlla quali istanze di Auto Scaling terminano durante la scalabilità nella Amazon Auto Scaling User Guide](#). EC2

Quando utilizzi il dimensionamento automatico del cluster, tieni in considerazione i seguenti aspetti:

- Non modificare o gestire la capacità desiderata per il gruppo con scalabilità automatica associato a un provider di capacità con policy di dimensionamento diverse da quelle gestite da Amazon ECS.
- Quando Amazon ECS passa da 0 istanze, ne avvia automaticamente 2.
- Amazon ECS utilizza il ruolo IAM `AWSServiceRoleForECS` collegato al servizio per le autorizzazioni necessarie per richiamare AWS Auto Scaling per tuo conto. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Quando utilizzi provider di capacità con gruppi con scalabilità automatica, l'utente, il gruppo e il ruolo che crea i provider di capacità richiedono l'autorizzazione `autoscaling:CreateOrUpdateTags`. Questo perché Amazon ECS aggiunge un tag al gruppo Auto Scaling quando lo associ al provider di capacità.

 Important

Assicurati che qualsiasi strumento utilizzato non rimuova il tag `AmazonECSManaged` dal gruppo con scalabilità automatica. Se questo tag viene rimosso, Amazon ECS non è in grado di gestire la scalabilità.

- La scalabilità automatica del cluster non modifica l'`MinimumCapacity` o `MaximumCapacity` per il gruppo. Affinché il gruppo venga scalato orizzontalmente, il valore di `MaximumCapacity` deve essere maggiore di zero.
- Quando è attivata la scalabilità automatica (dimensionamento gestito), un provider di capacità può essere connesso solo a un cluster contemporaneamente. Se per il provider di capacità la scalabilità gestita è disattivata, è possibile associarlo a più cluster.
- Quando il dimensionamento gestito è disattivato, il provider di capacità non esegue operazioni di dimensionamento con riduzione o aumento orizzontale. In questo caso, è possibile utilizzare una strategia del provider di capacità per bilanciare le attività tra i provider di capacità.
- La `binpack` strategia è la strategia più efficiente in termini di capacità.

- Quando la capacità target è inferiore al 100%, la strategia di collocamento deve utilizzare la binpack strategia deve avere un ordine superiore rispetto alla spread strategia. Ciò impedisce al provider di capacità di espandersi fino al raggiungimento del limite o fino al raggiungimento di un'istanza dedicata per ciascuna attività.

Attivazione del dimensionamento automatico del cluster

È possibile attivare la scalabilità automatica del cluster utilizzando la Console o il AWS CLI.

Quando crei un cluster per il tipo di EC2 avvio utilizzando la console, Amazon ECS crea un gruppo Auto Scaling per tuo conto e imposta la capacità target. Per ulteriori informazioni, consulta [Creazione di un cluster Amazon ECS per il tipo di EC2 lancio Amazon](#).

È inoltre possibile creare un gruppo Auto Scaling e quindi assegnarlo a un cluster. Per ulteriori informazioni, consulta [Aggiornamento di un provider di capacità Amazon ECS](#).

Quando si utilizza il AWS CLI, dopo aver creato il cluster

1. Prima di creare il provider di capacità, è necessario creare un gruppo con scalabilità automatica. Per ulteriori informazioni, consulta i [gruppi Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.
2. Utilizzare `put-cluster-capacity-providers` per modificare il provider di capacità del cluster. Per ulteriori informazioni, consulta [Attivazione della scalabilità automatica del cluster Amazon ECS](#).

Ottimizza la scalabilità automatica dei cluster Amazon ECS

I clienti che eseguono Amazon ECS su Amazon EC2 possono sfruttare la scalabilità automatica del cluster per gestire la scalabilità dei gruppi Amazon Auto Scaling EC2 . Con Cluster Auto Scaling, puoi configurare Amazon ECS per scalare automaticamente il tuo gruppo Auto Scaling e concentrarti solo sull'esecuzione delle tue attività. Amazon ECS garantisce la scalabilità interna e orizzontale del gruppo Auto Scaling secondo necessità senza ulteriori interventi. I provider di capacità di Amazon ECS vengono utilizzati per gestire l'infrastruttura del cluster assicurando che ci siano abbastanza istanze di container per soddisfare le esigenze dell'applicazione. Per scoprire come funziona la scalabilità automatica dei cluster, consulta [Deep Dive on Amazon ECS Cluster Auto Scaling](#).

La scalabilità automatica del cluster si basa su un'integrazione CloudWatch basata con il gruppo Auto Scaling per la regolazione della capacità del cluster. Pertanto ha una latenza intrinseca associata a

- Pubblicazione delle metriche, CloudWatch
- Il tempo impiegato dalla metrica per CapacityProviderReservation violare gli CloudWatch allarmi (sia alti che bassi)
- Il tempo impiegato da un' EC2 istanza Amazon appena lanciata per il riscaldamento. È possibile intraprendere le seguenti azioni per rendere la scalabilità automatica del cluster più reattiva per implementazioni più rapide:

Scalabilità graduale delle dimensioni del provider di capacità

I provider di capacità di Amazon ECS aumenteranno o ridurranno le istanze di container per soddisfare le esigenze della tua applicazione. Il numero minimo di istanze che Amazon ECS avvierà è impostato su 1 per impostazione predefinita. Ciò può aggiungere ulteriore tempo alle distribuzioni, se sono necessarie più istanze per collocare le attività in sospeso. Puoi aumentarlo [minimumScalingStepSize](#) tramite l'API Amazon ECS per aumentare il numero minimo di istanze scalabili da Amazon ECS alla volta. Un [maximumScalingStepSize](#) valore troppo basso può limitare il numero di istanze di container scalabili contemporaneamente, il che può rallentare le implementazioni.

Note

Questa configurazione è attualmente disponibile solo tramite o.
[CreateCapacityProviderUpdateCapacityProvider](#) APIs

Periodo di riscaldamento dell'istanza

Il periodo di riscaldamento dell'istanza è il periodo di tempo dopo il quale un' EC2 istanza Amazon appena lanciata può contribuire ai CloudWatch parametri per il gruppo Auto Scaling. Dopo la scadenza del periodo di riscaldamento specificato, l'istanza viene conteggiata nelle metriche aggregate del gruppo Auto Scaling e l'autoscaling del cluster procede con la successiva iterazione di calcoli per stimare il numero di istanze richieste.

Il valore predefinito per [instanceWarmupPeriod](#) è 300 secondi, che è possibile configurare su un valore inferiore tramite o per un ridimensionamento più reattivo.
[CreateCapacityProviderUpdateCapacityProvider](#) APIs Si consiglia di impostare il valore su un valore superiore a 60 secondi in modo da evitare un eccesso di provisioning.

Capacità inutilizzata

Se il tuo provider di capacità non dispone di istanze di container per l'inserimento delle attività, deve aumentare (scalare) la capacità del cluster avviando EC2 immediatamente le istanze Amazon e attendere che si avviino prima di poter avviare contenitori su di esse. Ciò può ridurre in modo significativo la frequenza di avvio delle attività. Hai due opzioni qui.

In questo caso, disporre di EC2 capacità Amazon di riserva già avviata e pronta per l'esecuzione delle attività aumenterà la percentuale effettiva di avvio delle attività. Puoi utilizzare la `Target Capacity` configurazione per indicare che desideri mantenere la capacità inutilizzata nei tuoi cluster. Ad esempio, `Target Capacity` impostando l'80%, si indica che il cluster necessita sempre del 20% di capacità di riserva. Questa capacità di riserva può consentire l'avvio immediato di qualsiasi attività autonoma, assicurando che l'avvio delle attività non venga limitato. Il compromesso di questo approccio è il potenziale aumento dei costi legati al mantenimento della capacità di riserva del cluster.

Un approccio alternativo che puoi prendere in considerazione è quello di aggiungere spazio di crescita al tuo servizio, non al fornitore di capacità. Ciò significa che invece di ridurre la `Target Capacity` configurazione per avviare la capacità di riserva, è possibile aumentare il numero di repliche nel servizio modificando la metrica di tracciamento della scalabilità di destinazione o le soglie di scalabilità graduale del servizio auto scaling. Tieni presente che questo approccio sarà utile solo per i carichi di lavoro con picchi di lavoro, ma non avrà alcun effetto quando distribuisce nuovi servizi e passi da 0 a N attività per la prima volta. Per ulteriori informazioni sulle politiche di scalabilità correlate, consulta [Target Tracking Scaling Policies o Step Scaling Policies](#) nella Amazon Elastic Container Service Developer Guide.

Comportamento di scalabilità gestito da Amazon ECS

Se disponi di fornitori di capacità di gruppo Auto Scaling che utilizzano la scalabilità gestita, Amazon ECS stima il numero ottimale di istanze da aggiungere al cluster e utilizza il valore per determinare quante istanze richiedere o rilasciare.

Comportamento dell'aumento orizzontale gestito

Amazon ECS seleziona un fornitore di capacità per ogni attività seguendo la strategia del fornitore di capacità dal servizio, dall'attività autonoma o dall'impostazione predefinita del cluster. Amazon ECS segue il resto di questi passaggi per un singolo provider di capacità.

Le attività senza una strategia di fornitore di capacità vengono ignorate dai fornitori di capacità. Un'attività in sospeso che non prevede una strategia del provider di capacità non comporterà

l'impiego della scalabilità orizzontale di alcun provider di capacità. Le attività o i servizi non possono configurare una strategia del provider di capacità se tale attività o servizio imposta un tipo di avvio.

Di seguito viene descritto il comportamento dell'aumento orizzontale in modo più dettagliato.

- Raggruppa tutti i processi di provisioning per questo provider di capacità in modo che ogni gruppo abbia gli stessi requisiti di risorse esatti.
- Quando utilizzi più istanze in un gruppo con dimensionamento automatico, le istanze all'interno di tale gruppo vengono ordinate in base ai relativi parametri. Questi parametri includono vCPU, memoria, interfacce di rete elastiche (ENIs), porte e GPUs. Vengono selezionati i tipi di istanza più piccoli e più grandi per ciascun parametro. Per ulteriori informazioni su come scegliere il tipo di istanza, consulta [Istanze di EC2 container Amazon per Amazon ECS](#)

Important

Se un gruppo di attività ha requisiti di risorse superiori al tipo di istanza più piccolo del gruppo con dimensionamento automatico, quel gruppo di attività non può essere eseguito con questo provider di capacità. Il provider di capacità non dimensiona il gruppo con dimensionamento automatico. Le attività rimangono nello stato PROVISIONING.

Per evitare che le attività rimangano nello stato PROVISIONING, consigliamo di creare gruppi con dimensionamento automatico e provider di capacità separati per diversi requisiti minimi di risorse. Quando esegui attività o crei servizi, aggiungi solo provider di capacità alla strategia del provider di capacità in grado di eseguire l'attività sul tipo di istanza più piccolo del gruppo con dimensionamento automatico. Per altri parametri, puoi utilizzare i vincoli di posizionamento

- Per ogni gruppo di attività, Amazon ECS calcola il numero di istanze necessarie per eseguire le attività non posizionate. Questo calcolo utilizza una strategia binpack. Questa strategia tiene conto della vCPU, della memoria, delle interfacce di rete elastiche (ENI), delle porte e dei GPUs requisiti delle attività. Inoltre, tiene conto della disponibilità delle risorse delle EC2 istanze Amazon. I valori per i tipi di istanza più grandi sono considerati quale numero massimo di istanze calcolato. I valori per il tipo di istanza più piccolo vengono utilizzati come protezione. Se il tipo di istanza più piccolo non può eseguire almeno un'istanza dell'attività, il calcolo considera l'attività come non compatibile. Di conseguenza, l'attività viene esclusa dal calcolo dell'aumento orizzontale. Quando tutte le attività non sono compatibili con il tipo di istanza più piccolo, il dimensionamento automatico del cluster si interrompe e il valore CapacityProviderReservation rimane targetCapacity.

- Amazon ECS pubblica la `CapacityProviderReservation` metrica in CloudWatch relazione al `minimumScalingStepSize` caso in cui si verifichi una delle seguenti condizioni.
 - Il numero massimo di istanze calcolato è inferiore alla dimensione minima del passaggio di scalabilità.
 - Il valore più basso del numero massimo di istanze calcolato `maximumScalingStepSize` o quello massimo.
- CloudWatch gli allarmi utilizzano la `CapacityProviderReservation` metrica per i fornitori di capacità. Quando il parametro `CapacityProviderReservation` è maggiore del valore di `targetCapacity`, gli allarmi aumentano anche la `DesiredCapacity` del gruppo con scalabilità automatica. Il `targetCapacity` valore è un'impostazione del provider di capacità che viene inviata all' CloudWatch allarme durante la fase di attivazione dell'auto scaling del cluster.

L'impostazione predefinita `targetCapacity` è 100%.

- Il gruppo Auto Scaling avvia istanze aggiuntive. EC2 Per evitare l'over-provisioning, Auto Scaling assicura che la capacità delle EC2 istanze lanciate di recente sia stabilizzata prima del lancio di nuove istanze. La scalabilità automatica verifica se tutte le istanze esistenti hanno superato il `instanceWarmupPeriod` (ora meno il tempo di avvio dell'istanza). La scalabilità orizzontale è bloccata per le istanze che si trovano all'interno di `instanceWarmupPeriod`

Il tempo di default per il riscaldamento di un'istanza appena avviata è di 300 secondi.

Per maggiori informazioni, consulta [Approfondimento sulla scalabilità automatica del cluster Amazon ECS](#).

Considerazioni sull'aumento orizzontale

Considera quanto segue per il processo di aumento orizzontale:

- Sebbene esistano più vincoli di collocamento, è consigliabile utilizzare solo il vincolo `distinctInstance` per la collocazione delle attività. Ciò impedisce l'arresto del processo di aumento orizzontale in seguito all'utilizzo di un vincolo di posizionamento non compatibile con le istanze campionate.
- La scalabilità gestita funziona meglio se il gruppo Auto Scaling utilizza tipi di istanza uguali o simili.
- Quando è necessario un processo di scalabilità orizzontale e non sono presenti istanze di container attualmente in esecuzione, inizialmente Amazon ECS impiega sempre la scalabilità orizzontale fino a due istanze, quindi esegue processi di scalabilità orizzontale o ridimensionamento aggiuntivi. Qualsiasi ulteriore impiego della scalabilità orizzontale attende il

periodo di preparazione dell'istanza. Per i processi di ridimensionamento, Amazon ECS attende 15 minuti dopo un processo di scalabilità orizzontale prima di avviare in qualsiasi momento i processi di ridimensionamento.

- La seconda fase di aumento orizzontale deve attendere fino allo scadere del `instanceWarmupPeriod`, che potrebbe influire sul limite di scalabilità complessivo. Se devi ridurre questo tempo, assicurati che `instanceWarmupPeriod` sia sufficientemente grande da consentire all' EC2istanza di avviare e avviare l'agente Amazon ECS (che impedisce l'overprovisioning).
- Il dimensionamento automatico del cluster supporta la configurazione di avvio, i modelli di avvio e più tipi di istanze nel gruppo con dimensionamento automatico del provider di capacità. Puoi inoltre utilizzare la selezione del tipo di istanza basata su attributi senza molteplici tipi di istanze.
- Quando utilizzi un gruppo Auto Scaling con istanze on demand e più tipi di istanza o istanze Spot, posiziona i tipi di istanza più grandi più in alto nell'elenco di priorità e non specificare un peso. Al momento la specifica di un peso non è supportata. Per ulteriori informazioni, consulta [Gruppi Auto Scaling con più tipi di istanze](#) nella Guida per l'utente di AWS Auto Scaling .
- Amazon ECS avvierà quindi `minimumScalingStepSize`, se il conteggio massimo delle istanze calcolate è inferiore alla dimensione minima del passo di dimensionamento o il valore inferiore di `maximumScalingStepSize` o del valore massimo del conteggio delle istanze calcolato.
- Se un servizio Amazon ECS o `run-task` avvia un'attività e le istanze del container del Capacity Provider non dispongono di risorse sufficienti per avviare l'attività, Amazon ECS limita il numero di attività con questo stato per ogni cluster e impedisce che qualsiasi attività superi questo limite. Per ulteriori informazioni, consulta [Quote del servizio](#).

Comportamento di riduzione orizzontale gestita

Amazon ECS monitora le istanze di container per ciascun provider di capacità all'interno di un cluster. Quando un'istanza di container non esegue alcuna attività, viene considerata vuota e Amazon ECS avvia il processo di ridimensionamento.

CloudWatch gli allarmi scale-in richiedono 15 punti dati (15 minuti) prima dell'avvio del processo di scalabilità per il gruppo Auto Scaling. Dopo l'avvio del processo di riduzione orizzontale, fino a quando Amazon ECS non ha bisogno di ridurre il numero di istanze di container registrate, il gruppo con scalabilità automatica imposta il valore `DesireCapacity` di modo che sia superiore a un'istanza e inferiore al 50% ogni minuto.

Quando Amazon ECS richiede un aumento orizzontale (quando `CapacityProviderReservation` è maggiore di 100) mentre è in corso un processo di riduzione orizzontale, il processo di riduzione orizzontale viene interrotto e ricomincerà daccapo, se necessario.

Di seguito viene descritto il comportamento del ridimensionamento in modo più dettagliato:

1. Amazon ECS calcola il numero di istanze di container vuote. Un'istanza di container è considerata vuota quando non sono in esecuzione attività daemon.
2. Amazon ECS imposta il valore `CapacityProviderReservation` su un numero compreso tra 0 e 100 che utilizza la seguente formula per rappresentare il rapporto tra la dimensione prevista del gruppo con dimensionamento automatico e la sua dimensione effettiva, espresso in percentuale. Quindi, Amazon ECS pubblica la metrica su CloudWatch. Per maggiori informazioni su come viene calcolato il parametro, consulta [Approfondimento sul dimensionamento automatico del cluster Amazon ECS](#)

$$\text{CapacityProviderReservation} = (\text{number of instances needed}) / (\text{number of running instances}) \times 100$$

3. La `CapacityProviderReservation` metrica genera un allarme. CloudWatch. Questo allarme aggiorna il valore di `DesiredCapacity` per il gruppo Auto Scaling. Quindi, si verifica una delle seguenti operazioni:
 - Se non si utilizza la terminazione gestita dal provider di capacità, il gruppo Auto Scaling EC2 seleziona le istanze utilizzando la politica di terminazione del gruppo Auto Scaling e termina le istanze finché il numero di istanze non raggiunge il `EC2 DesiredCapacity`. Viene quindi annullata la registrazione delle istanze di container dal cluster.
 - Se tutte le istanze di container utilizzano la protezione da terminazione gestita, Amazon ECS rimuove la protezione da ridimensionamento sulle istanze di container vuote. Il gruppo Auto Scaling sarà quindi in grado di terminare le istanze. Viene quindi annullata la registrazione delle istanze di container dal cluster.

Controlla le istanze terminate da Amazon ECS

Important

Devi attivare la protezione da ridimensionamento dell'istanza del dimensionamento automatico nel gruppo con dimensionamento automatico per utilizzare la funzionalità di protezione da terminazione gestita del dimensionamento automatico del cluster.

La protezione gestita dalla terminazione consente la scalabilità automatica del cluster per controllare quali istanze vengono terminate. Quando hai utilizzato la protezione gestita dalle terminazioni, Amazon ECS termina solo EC2 le istanze in cui non è in esecuzione alcuna attività Amazon ECS. Le attività eseguite da un servizio che utilizza la strategia di pianificazione DAEMON vengono ignorate e un'istanza può essere terminata mediante il dimensionamento automatico del cluster anche quando l'istanza esegue queste attività. Ciò avviene perché tutte le istanze del cluster eseguono queste attività.

Amazon ECS attiva innanzitutto l'opzione di protezione scale-in delle istanze per EC2 le istanze del gruppo Auto Scaling. Quindi, Amazon ECS inserisce le attività sulle istanze. Quando tutte le attività non daemon vengono interrotte su un'istanza, Amazon ECS avvia il processo di scalabilità in e disattiva la protezione scalabile per l'istanza. EC2 Il gruppo Auto Scaling può quindi terminare l'istanza.

La protezione scalabile in base alle istanze di Auto Scaling controlla quali EC2 istanze possono essere terminate mediante Auto Scaling. Le istanze con la funzione di riduzione orizzontale attivata non possono essere terminate durante il processo di riduzione orizzontale. Per ulteriori informazioni sulla protezione scalabile in base alle istanze di Auto Scaling, consulta [Using Instance Scale-In protection nella Amazon Auto EC2 Scaling User Guide](#).

Puoi impostare la `targetCapacity` percentuale in modo da disporre di capacità inutilizzata. Ciò consente di avviare più rapidamente le attività future perché il gruppo Auto Scaling non deve avviare più istanze. Amazon ECS utilizza il valore della capacità target per gestire la CloudWatch metrica creata dal servizio. Amazon ECS gestisce la CloudWatch metrica. Il gruppo Auto Scaling viene trattato come uno stato stazionario in modo che non sia richiesta alcuna azione di ridimensionamento. I valori possono essere compresi tra 0 e 100%. Ad esempio, per configurare Amazon ECS per mantenere una capacità libera del 10% oltre a quella utilizzata dalle attività Amazon ECS, imposta il valore della capacità target al 90%. Quando imposti il valore di `targetCapacity` su un provider di capacità, tieni presenti le considerazioni seguenti.

- Un `targetCapacity` valore inferiore al 100% rappresenta la quantità di capacità libera (EC2 istanze Amazon) che deve essere presente nel cluster. Capacità libera significa che non ci sono attività in esecuzione.
- I vincoli di posizionamento come le zone di disponibilità, senza ulteriori binpack, costringono Amazon ECS a eseguire infine un'attività per ogni istanza, comportamento che potrebbe non essere quello desiderato.

Devi attivare la protezione da ridimensionamento dell'istanza del dimensionamento automatico nel gruppo con dimensionamento automatico per utilizzare la protezione da terminazione gestita. Se non attivi la protezione da ridimensionamento, l'attivazione della protezione da terminazione gestita può portare a comportamenti indesiderati. Ad esempio, è possibile che le istanze siano bloccate in uno stato di svuotamento. Per ulteriori informazioni, consulta [Using Instance Scale-In Protection nella Amazon Auto EC2 Scaling User Guide](#).

Quando utilizzi la protezione da terminazione con un provider di capacità, non eseguire operazioni manuali, come lo scollegamento dell'istanza, sul gruppo con dimensionamento automatico associato al provider di capacità. Le azioni manuali possono interrompere il processo di ridimensionamento del provider di capacità. Se scolleghi un'istanza dal gruppo con dimensionamento automatico, devi anche [annullare la registrazione dell'istanza scollegata](#) dal cluster Amazon ECS.

Aggiornamento della protezione gestita dalle terminazioni per i provider di capacità Amazon ECS

Quando utilizzi la protezione dalla terminazione gestita, devi aggiornare l'impostazione per i provider di capacità esistenti.

Console

1. Apri la console Amazon ECS all'indirizzo <https://console.aws.amazon.com/ecs/>
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Seleziona il cluster che contiene il provider di capacità che desideri aggiornare.
4. Nella pagina del cluster, seleziona la scheda Capacity Providers.
5. Seleziona il provider di capacità che desideri aggiornare.
6. Scegli Aggiorna per modificare le impostazioni del provider di capacità.
7. Nelle impostazioni del gruppo Auto Scaling, attiva la protezione da terminazione gestita per abilitare o disabilitare la funzione.
8. Scegli Update (Aggiorna) per salvare le modifiche.

AWS CLI

È possibile aggiornare l'impostazione di protezione dalla terminazione gestita di un provider di capacità utilizzando il comando: `update-capacity-provider`

Per abilitare la protezione gestita dalla terminazione:

```
aws ecs update-capacity-provider \
```

```
--capacity-provider CapacityProviderName \
--auto-scaling-group-provider
"managedScaling={status=ENABLED,targetCapacity=70,minimumScalingStepSize=1,maximumScalingStepS
```

Per disabilitare la protezione gestita dalla terminazione:

```
aws ecs update-capacity-provider \
--capacity-provider CapacityProviderName \
--auto-scaling-group-provider
"managedScaling={status=ENABLED,targetCapacity=70,minimumScalingStepSize=1,maximumScalingStepS
```

Note

Potrebbero essere necessari alcuni minuti prima che le modifiche abbiano effetto nel cluster. Quando si abilita la protezione gestita dalla terminazione, le istanze che stanno già eseguendo attività verranno protette dagli eventi di scalabilità. Quando si disabilita la protezione dalla terminazione gestita, il flag di protezione verrà rimosso dalle istanze durante il successivo ciclo di gestione del provider di capacità ECS.

Attivazione della scalabilità automatica del cluster Amazon ECS

Attivi la scalabilità automatica del cluster in modo che Amazon ECS gestisca la scalabilità delle EC2 istanze Amazon registrate nel tuo cluster.

Se desideri utilizzare la console per attivare la scalabilità automatica del cluster, [Creazione di un provider di capacità per Amazon ECS](#) consulta.

Prima di iniziare, crea un gruppo con scalabilità automatica e un provider di capacità. Per ulteriori informazioni, consulta [the section called “Fornitori di capacità per il tipo di EC2 lancio”](#).

Per attivare la scalabilità automatica del cluster, si associa il provider di capacità al cluster, quindi si attiva la scalabilità automatica del cluster.

1. Esegui il comando `put-cluster-capacity-providers` per associare uno o più provider di capacità al cluster.

Per aggiungere i provider AWS Fargate di capacità, includi `FARGATE` i fornitori `FARGATE_SPOT` di capacità nella richiesta. Per ulteriori informazioni, consulta la sezione [put-cluster-capacity-providers](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \  
  --cluster ClusterName \  
  --capacity-providers CapacityProviderName FARGATE FARGATE_SPOT \  
  --default-capacity-provider-strategy capacityProvider=CapacityProvider,weight=1
```

Per aggiungere un gruppo Auto Scaling per il tipo di EC2 avvio, includi il nome del gruppo Auto Scaling nella richiesta. Per ulteriori informazioni, consulta la sezione [put-cluster-capacity-providers](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \  
  --cluster ClusterName \  
  --capacity-providers CapacityProviderName \  
  --default-capacity-provider-strategy capacityProvider=CapacityProvider,weight=1
```

2. Utilizza il comando `describe-clusters` per verificare che l'associazione abbia avuto successo. Per ulteriori informazioni, consulta la sezione [describe-clusters](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs describe-clusters \  
  --cluster ClusterName \  
  --include ATTACHMENTS
```

3. Esegui il comando `update-capacity-provider` per attivare la scalabilità automatica gestita per il provider di capacità. Per ulteriori informazioni, consulta la sezione [update-capacity-provider](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs update-capacity-provider \  
  --capacity-providers CapacityProviderName \  
  --auto-scaling-group-provider managedScaling=ENABLED
```

Disattivazione della scalabilità automatica del cluster Amazon ECS

Disattivi la scalabilità automatica del cluster quando hai bisogno di un controllo più granulare delle EC2 istanze registrate nel cluster,

Per disattivare la scalabilità automatica del cluster per un cluster, è possibile disassociare il provider di capacità con scalabilità gestita attivato dal cluster o aggiornare il provider di capacità per disattivare la scalabilità gestita.

Dissociare il provider di capacità

Utilizza la procedura seguente per annullare l'associazione di un provider di capacità con un cluster.

1. Utilizza il comando `put-cluster-capacity-providers` per annullare l'associazione del provider di capacità del gruppo con scalabilità automatica con il cluster. Il cluster può mantenere l'associazione con i fornitori AWS Fargate di capacità. Per ulteriori informazioni, consulta la sezione [put-cluster-capacity-providers](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \  
  --cluster ClusterName \  
  --capacity-providers FARGATE FARGATE_SPOT \  
  --default-capacity-provider-strategy '[]'
```

Utilizza il comando `put-cluster-capacity-providers` per annullare l'associazione del provider di capacità del gruppo con scalabilità automatica con il cluster. Per ulteriori informazioni, consulta la sezione [put-cluster-capacity-providers](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \  
  --cluster ClusterName \  
  --capacity-providers [] \  
  --default-capacity-provider-strategy '[]'
```

2. Utilizza il comando `describe-clusters` per verificare che l'annullamento dell'associazione abbia avuto successo. Per ulteriori informazioni, consulta la sezione [describe-clusters](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs describe-clusters \  
  --cluster ClusterName \  
  --include ATTACHMENTS
```

Disattivare la scalabilità gestita per il provider di capacità

Utilizza la procedura seguente per disattivare la scalabilità gestita per il provider di capacità.

- Utilizza il comando `update-capacity-provider` per attivare la scalabilità automatica gestita per il provider di capacità. Per ulteriori informazioni, consulta la sezione [update-capacity-provider](#) nella Documentazione di riferimento della AWS CLI .

```
aws ecs update-capacity-provider \  
  --capacity-providers CapacityProviderName \  
  --auto-scaling-group-provider managedScaling=DISABLED
```

Creazione di un provider di capacità per Amazon ECS

Una volta completata la creazione del cluster, è possibile creare un nuovo provider di capacità (gruppo Auto Scaling) per EC2 il tipo di avvio. I provider di capacità aiutano a gestire e scalare l'infrastruttura per le applicazioni.

Prima di creare il provider di capacità, è necessario creare un gruppo con scalabilità automatica. Per ulteriori informazioni, consulta i [gruppi Auto Scaling](#) nella Amazon Auto EC2 Scaling User Guide.

Creazione di un provider di capacità per il cluster (console Amazon ECS)

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella *name* pagina Cluster:, scegli Infrastruttura, quindi scegli Crea.
5. Nella pagina Create capacity providers (Crea provider di capacità), configura le seguenti opzioni.
 - a. In Basic details (Dettagli di base), per Capacity provider name (Nome provider di capacità), immetti un nome univoco per il provider di capacità.
 - b. In Auto Scaling group (Gruppo con scalabilità automatica), per Use an existing Auto Scaling group (Utilizza un gruppo con scalabilità automatica esistente), scegli il gruppo con scalabilità automatica.
 - c. (Facoltativo) Per configurare una policy di scalabilità, in Scaling policies (Policy di scalabilità), configura le seguenti opzioni.
 - Per consentire ad Amazon ECS di gestire le operazioni di riduzione e aumento orizzontale, seleziona Turn on managed scaling (Attiva scalabilità gestita).
 - Per evitare che l' EC2 istanza con attività in esecuzione di Amazon ECS venga interrotta, seleziona Turn on Scaling Protection.

- Per Set target capacity, inserisci il valore target per la CloudWatch metrica utilizzata nella policy di scalabilità di tracciamento degli obiettivi gestita da Amazon ECS.

6. Scegli Create (Crea).

Aggiornamento di un provider di capacità Amazon ECS

Quando si utilizza un gruppo con scalabilità automatica come provider di capacità, è possibile modificare la policy di scalabilità del gruppo.

Per aggiornare un provider di capacità per il cluster (console Amazon ECS)

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella *name* pagina Cluster:, scegli Infrastruttura, quindi scegli Aggiorna.
5. Nella pagina Create capacity providers (Crea provider di capacità), configura le seguenti opzioni.
 - Nella sezione Gruppo con scalabilità automatica, in Policy di dimensionamento, configura le seguenti opzioni.
 - Per consentire ad Amazon ECS di gestire le operazioni di riduzione e aumento orizzontale, seleziona Turn on managed scaling (Attiva scalabilità gestita).
 - Per evitare che EC2 le istanze con attività in esecuzione di Amazon ECS vengano terminate, seleziona Attiva la protezione della scalabilità.
 - Per Set target capacity, inserisci il valore target per la CloudWatch metrica utilizzata nella policy di scalabilità di tracciamento degli obiettivi gestita da Amazon ECS.
6. Scegli Aggiorna.

Eliminazione di un provider di capacità Amazon ECS

Al termine dell'utilizzo di un provider di capacità del gruppo con dimensionamento automatico, puoi eliminarlo. Dopo l'eliminazione del gruppo, il provider di capacità del gruppo Auto Scaling passa allo stato. INACTIVE I provider di capacità con stato INACTIVE potrebbero rimanere rilevabili nell'account per un periodo di tempo. Tuttavia, questo comportamento è soggetto a modifiche in futuro, quindi non dovresti fare affidamento sulla persistenza dei provider di capacità INACTIVE. Prima di eliminare un provider di capacità del gruppo con scalabilità automatica, il provider di capacità

deve essere rimosso dalla strategia del provider di capacità per tutti i servizi. L'API UpdateService o il flusso di lavoro del servizio di aggiornamento nella console Amazon ECS possono essere utilizzati per rimuovere un provider di capacità dalla strategia del provider di capacità di un servizio. Utilizza l'opzione Force new deployment per assicurarti che tutte le attività che utilizzano la capacità delle EC2 istanze Amazon fornita dal fornitore di capacità vengano trasferite all'utilizzo della capacità dei fornitori di capacità rimanenti.

Per eliminare un provider di capacità per il cluster (console Amazon ECS)

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella **name** pagina Cluster:, scegli Infrastruttura, il gruppo Auto Scaling, quindi scegli Elimina.
5. Nella casella di conferma, inserisci delete **Auto Scaling group name**
6. Scegliere Delete (Elimina).

Blocca in sicurezza i carichi di lavoro Amazon ECS in esecuzione sulle istanze EC2

Il drenaggio gestito delle istanze facilita la chiusura graduale delle istanze Amazon. EC2 Ciò consente ai carichi di lavoro di interrompersi in sicurezza e di essere riprogrammati su istanze non terminanti. La manutenzione e gli aggiornamenti dell'infrastruttura vengono eseguiti senza preoccuparsi dell'interruzione dei carichi di lavoro. Utilizzando il drenaggio gestito delle istanze, semplifichi i flussi di lavoro di gestione dell'infrastruttura che richiedono la sostituzione delle EC2 istanze Amazon, garantendo al contempo la resilienza e la disponibilità delle applicazioni.

Il drenaggio gestito delle istanze di Amazon ECS funziona con le istanze di gruppo Auto Scaling sostitutive. In base all'aggiornamento dell'istanza e alla durata massima dell'istanza, i clienti possono garantire la conformità ai più recenti requisiti di sistema operativo e di sicurezza relativi alla loro capacità.

Il drenaggio gestito delle istanze può essere utilizzato solo con i provider di capacità Amazon ECS. Puoi attivare il drenaggio gestito delle istanze quando crei o aggiorni i fornitori di capacità del gruppo Auto Scaling utilizzando la console AWS CLI Amazon ECS o l'SDK.

I seguenti eventi sono coperti dal drenaggio delle istanze gestite di Amazon ECS.

- Aggiornamento delle [istanze del gruppo Auto Scaling - Utilizza l'aggiornamento](#) delle istanze per eseguire la sostituzione progressiva delle EC2 istanze Amazon nel gruppo Auto Scaling

anziché eseguirla manualmente in batch. Ciò è utile quando è necessario sostituire un gran numero di istanze. L'aggiornamento dell'istanza viene avviato tramite la EC2 console Amazon o l'`StartInstanceRefreshAPI`. Assicurati di selezionare `Replace` la protezione Scale-in quando chiami `StartInstanceRefresh` se utilizzi la protezione gestita dalla terminazione.

- [Durata massima dell'istanza](#) - È possibile definire una durata massima quando si tratta di sostituire le istanze del gruppo Auto Scaling. Ciò è utile per pianificare le istanze sostitutive in base a politiche di sicurezza interne o alla conformità.
- Scale-in del gruppo Auto Scaling - Basato su politiche di scalabilità e azioni di scalabilità pianificate, il gruppo Auto Scaling supporta il ridimensionamento automatico delle istanze. Utilizzando un gruppo Auto Scaling come fornitore di capacità Amazon ECS, puoi scalare le istanze di gruppo Auto Scaling quando non sono in esecuzione alcuna attività.
- [Controlli di integrità del gruppo Auto Scaling](#) - Il gruppo Auto Scaling supporta molti controlli di integrità per gestire la chiusura delle istanze non integre.
- [AWS CloudFormation aggiornamenti dello stack](#) - È possibile aggiungere un `UpdatePolicy` attributo allo AWS CloudFormation stack per eseguire aggiornamenti continui quando il gruppo cambia.
- [Ribilanciamento della capacità Spot](#) - Il gruppo Auto Scaling cerca di sostituire in modo proattivo le istanze Spot che presentano un rischio di interruzione più elevato in base all'avviso di ribilanciamento della capacità di Amazon. EC2 Il gruppo Auto Scaling termina la vecchia istanza quando la variante sostitutiva viene lanciata ed è funzionante. Il drenaggio dell'istanza gestita di Amazon ECS prosciuga l'istanza Spot nello stesso modo in cui drena un'istanza non Spot.
- [Interruzione in loco](#) - Le istanze Spot vengono chiuse con un preavviso di due minuti. Il drenaggio delle istanze gestito da Amazon ECS mette l'istanza in stato di drenaggio in risposta.

Il ciclo di vita di Amazon EC2 Auto Scaling si aggancia al drenaggio gestito delle istanze

Gli hook del ciclo di vita del gruppo Auto Scaling consentono al cliente di creare soluzioni attivate da determinati eventi nel ciclo di vita dell'istanza ed eseguire un'azione personalizzata quando si verifica quel determinato evento. Un gruppo Auto Scaling consente fino a 50 hook. Possono esistere più hook di terminazione che vengono eseguiti in parallelo e il gruppo Auto Scaling attende il completamento di tutti gli hook prima di terminare un'istanza.

Oltre alla terminazione degli hook gestita da Amazon ECS, puoi anche configurare i tuoi hook di terminazione del ciclo di vita. Gli hook del ciclo di vita dispongono di un `default action hook` e consigliamo di impostarlo `continue` come predefinito per garantire che altri hook, come l'hook gestito di Amazon ECS, non siano influenzati da errori causati dagli hook personalizzati.

Se hai già configurato un hook del ciclo di vita di terminazione del gruppo Auto Scaling e hai anche abilitato il drenaggio delle istanze gestite di Amazon ECS, vengono eseguiti entrambi gli hook del ciclo di vita. Le tempistiche relative, tuttavia, non sono garantite. I Lifecycle Hook hanno un default action impostazione per specificare l'azione da intraprendere allo scadere del timeout. In caso di errori, consigliamo di utilizzare `continue` come risultato predefinito l'hook personalizzato. Ciò garantisce che altri hook, in particolare gli hook gestiti di Amazon ECS, non siano influenzati da errori nel tuo hook personalizzato del ciclo di vita. Il risultato alternativo di `abandon` fa sì che tutti gli altri hook vengano ignorati e dovrebbe essere evitato. Per ulteriori informazioni sugli hook del ciclo di vita del gruppo Auto Scaling, consulta gli hook del ciclo di vita di Amazon Auto [Scaling nella Amazon EC2 Auto Scaling User Guide](#). EC2

Attività e drenaggio gestito delle istanze

Il drenaggio gestito delle istanze di Amazon ECS utilizza la funzionalità di drenaggio esistente presente nelle istanze di container. La funzionalità di [drenaggio delle istanze del contenitore](#) esegue la sostituzione e l'interruzione delle attività di replica che appartengono a un servizio Amazon ECS. Un'attività autonoma, come quella richiamata da `RunTask`, che si trova nello stato `or` rimane inalterata `PENDING`. `RUNNING` È necessario attendere che vengano completate o interromperle manualmente. L'istanza del contenitore rimane nello `DRAINING` stato finché tutte le attività non vengono interrotte o non sono trascorse 48 ore. Le attività daemon sono le ultime a interrompersi dopo l'interruzione di tutte le attività di replica.

Drenaggio gestito delle istanze e protezione gestita dalla terminazione

Il drenaggio gestito delle istanze funziona anche se la terminazione gestita è disabilitata. Per informazioni sulla protezione gestita dalla terminazione, vedere. [Controlla le istanze terminate da Amazon ECS](#)

La tabella seguente riassume il comportamento delle diverse combinazioni di terminazione gestita e drenaggio gestito.

Terminazione gestita	Drenaggio gestito	Outcome
Abilitato	Abilitato	Amazon ECS protegge EC2 le istanze Amazon

Terminazione gestita	Drenaggio gestito	Outcome
		che eseguono attività dall'interruzione a causa di eventi di scalabilità. Tutte le istanze in fase di terminazione, ad esempio quelle che non hanno impostato la protezione e dalla terminazione, che hanno subito un'interruzione Spot o sono state forzate dall'aggiornamento dell'istanza, vengono esaurite

Terminazione gestita	Drenaggio gestito	Outcome
		senza problemi.
Disabilitato	Abilitato	Amazon ECS non protegge EC2 le istanze Amazon che eseguono attività dalla scalabilità. Tuttavia, tutte le istanze che vengono terminate vengono completamente svuotate.

Terminazione gestita	Drenaggio gestito	Outcome
Abilitato	Disabilitato	Amazon ECS protegge EC2 le istanze Amazon che eseguono attività dall'interruzione a causa di eventi di scalabilità. Tuttavia, le istanze possono comunque essere interrotte e dall'interruzione di Spot o dall'aggiornamento forzato dell'istanza o se non eseguono alcuna attività. Amazon ECS non esegue

Terminazione gestita	Drenaggio gestito	Outcome
		operazion i di drenaggio regolari per queste istanze e avvia attività di assisten a sostituti ve dopo che queste si sono interrotte.

Terminazione gestita	Drenaggio gestito	Outcome
Disabilitato	Disabilitato	EC2 Le istanze Amazon possono essere scalate o terminate in qualsiasi momento, anche se eseguono attività di Amazon ECS. Amazon ECS avvierà attività di assistenza sostitutive una volta interrotte.

Drenaggio gestito delle istanze e drenaggio delle istanze Spot

Con il drenaggio delle istanze Spot, puoi impostare una variabile di ambiente `ECS_ENABLE_SPOT_INSTANCE_DRAINING` sull'agente Amazon ECS che consente ad Amazon ECS di mettere un'istanza nello stato di drenaggio in risposta all'interruzione Spot di due minuti. Il drenaggio gestito delle istanze di Amazon ECS facilita l'arresto graduale delle istanze EC2 Amazon in fase di chiusura per molte ragioni, non solo per un'interruzione Spot. Ad esempio, puoi utilizzare il ribilanciamento della capacità di Amazon EC2 Auto Scaling per sostituire in modo proattivo l'istanza Spot con un rischio elevato di interruzione, mentre il drenaggio gestito delle istanze esegue l'arresto graduale dell'istanza Spot in fase di sostituzione. Quando si utilizza il drenaggio gestito delle istanze, non è necessario abilitare il drenaggio delle istanze Spot separatamente,

quindi `ECS_ENABLE_SPOT_INSTANCE_DRAINING` nel gruppo Auto Scaling i dati degli utenti sono ridondanti. Per ulteriori informazioni sul drenaggio delle istanze Spot, consulta [Spot Instances](#)

Come funziona il drenaggio gestito delle istanze con EventBridge

Gli eventi di drenaggio delle istanze gestite di Amazon ECS vengono pubblicati su Amazon EventBridge e Amazon ECS crea una regola EventBridge gestita nel bus predefinito del tuo account per supportare il drenaggio gestito delle istanze. Puoi filtrare questi eventi su altri AWS servizi come Lambda, Amazon SNS e Amazon SQS per monitorare e risolvere i problemi.

- Amazon EC2 Auto Scaling invia un evento EventBridge quando viene richiamato un hook del ciclo di vita.
- Gli avvisi di interruzione Spot vengono pubblicati su EventBridge
- Amazon ECS genera messaggi di errore che puoi recuperare tramite la console Amazon ECS e APIs
- EventBridge dispone di meccanismi di riprova integrati per mitigare i guasti temporanei.

Configurazione dei provider di capacità Amazon ECS per chiudere le istanze in modo sicuro

Puoi attivare il drenaggio delle istanze gestite quando crei o aggiorni i tuoi fornitori di capacità di gruppo Auto Scaling utilizzando la console Amazon ECS e AWS CLI

Note

Il drenaggio gestito delle istanze è attivo per impostazione predefinita quando crei un provider di capacità.

Di seguito sono riportati alcuni esempi di utilizzo di AWS CLI per creare un provider di capacità con il drenaggio gestito delle istanze abilitate e di abilitazione del drenaggio gestito delle istanze per il provider di capacità esistente di un cluster.

Crea un provider di capacità con il drenaggio gestito delle istanze abilitato

Per creare un provider di capacità con il drenaggio gestito delle istanze abilitate, usa il `create-capacity-provider` comando. Imposta il parametro `managedDraining` su `ENABLED`.

```
aws ecs create-capacity-provider \  
--name capacity-provider \  
--managed-draining ENABLED
```

```
--auto-scaling-group-provider '{
  "autoScalingGroupArn": "asg-arn",
  "managedScaling": {
    "status": "ENABLED",
    "targetCapacity": 100,
    "minimumScalingStepSize": 1,
    "maximumScalingStepSize": 1
  },
  "managedDraining": "ENABLED",
  "managedTerminationProtection": "ENABLED",
}'
```

Risposta:

```
{
  "capacityProvider": {
    "capacityProviderArn": "capacity-provider-arn",
    "name": "capacity-provider",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
      "autoScalingGroupArn": "asg-arn",
      "managedScaling": {
        "status": "ENABLED",
        "targetCapacity": 100,
        "minimumScalingStepSize": 1,
        "maximumScalingStepSize": 1
      },
      "managedTerminationProtection": "ENABLED",
      "managedDraining": "ENABLED"
    },
  },
}
```

Abilita il drenaggio gestito delle istanze per il provider di capacità esistente di un cluster

Abilita il drenaggio gestito delle istanze per il provider di capacità esistente di un cluster utilizza il `update-capacity-provider` comando. Vedete che `managedDraining` attualmente dice `DISABLED` e `updateStatus` dice `UPDATE_IN_PROGRESS`.

```
aws ecs update-capacity-provider \
--name cp-draining \
--auto-scaling-group-provider '{
  "managedDraining": "ENABLED"
```

```
}

```

Risposta:

```
{
  "capacityProvider": {
    "capacityProviderArn": "cp-draining-arn",
    "name": "cp-draining",
    "status": "ACTIVE",
    "autoScalingGroupProvider": {
      "autoScalingGroupArn": "asg-draining-arn",
      "managedScaling": {
        "status": "ENABLED",
        "targetCapacity": 100,
        "minimumScalingStepSize": 1,
        "maximumScalingStepSize": 1,
        "instanceWarmupPeriod": 300
      },
      "managedTerminationProtection": "DISABLED",
      "managedDraining": "DISABLED" // before update
    },
    "updateStatus": "UPDATE_IN_PROGRESS", // in progress and need describe again to
    find out the result
    "tags": [
    ]
  }
}
```

Usa il `describe-clusters` comando e includi `ATTACHMENTS`. L'allegato status di drenaggio dell'istanza gestita è `PRECREATED`, e il complesso `attachmentsStatus` è `UPDATING`.

```
aws ecs describe-clusters --clusters cluster-name --include ATTACHMENTS

```

Risposta:

```
{
  "clusters": [
    {
      ...

      "capacityProviders": [
        "cp-draining"
      ]
    }
  ]
}
```

```

    ],
    "defaultCapacityProviderStrategy": [],
    "attachments": [
      # new precreated managed draining attachment
      {
        "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "type": "managed_draining",
        "status": "PRECREATED",
        "details": [
          {
            "name": "capacityProviderName",
            "value": "cp-draining"
          },
          {
            "name": "autoScalingLifecycleHookName",
            "value": "ecs-managed-draining-termination-hook"
          }
        ]
      },
      ...
    ],
    "attachmentsStatus": "UPDATING"
  },
  "failures": []
}

```

Quando l'aggiornamento è terminato `describe-capacity-providers`, usa e lo vedi `managedDraining` ora `ENABLED`.

```
aws ecs describe-capacity-providers --capacity-providers cp-draining
```

Risposta:

```

{
  "capacityProviders": [
    {
      "capacityProviderArn": "cp-draining-arn",
      "name": "cp-draining",
      "status": "ACTIVE",
      "autoScalingGroupProvider": {

```

```
        "autoScalingGroupArn": "asg-draning-arn",
        "managedScaling": {
            "status": "ENABLED",
            "targetCapacity": 100,
            "minimumScalingStepSize": 1,
            "maximumScalingStepSize": 1,
            "instanceWarmupPeriod": 300
        },
        "managedTerminationProtection": "DISABLED",
        "managedDraining": "ENABLED" // successfully update
    },
    "updateStatus": "UPDATE_COMPLETE",
    "tags": []
}
]
```

Risoluzione dei problemi relativi al drenaggio delle istanze di Amazon ECS Managed

Potrebbe essere necessario risolvere i problemi relativi al drenaggio gestito delle istanze. Di seguito è riportato un esempio di un problema e di una risoluzione che potresti riscontrare durante l'utilizzo.

Le istanze non terminano dopo aver superato la durata massima dell'istanza quando si utilizza la scalabilità automatica.

Se le istanze non terminano nemmeno dopo aver raggiunto e superato la durata massima dell'istanza durante l'utilizzo di un gruppo con scalabilità automatica, è possibile che siano protette dalla scalabilità in ingresso. Puoi disattivare la terminazione gestita e consentire il drenaggio gestito per gestire il riciclo delle istanze.

Creazione di risorse per la scalabilità automatica del cluster Amazon ECS utilizzando AWS Management Console

Scopri come creare le risorse per la scalabilità automatica del cluster Amazon ECS utilizzando l'AWS Management Console. Laddove le risorse richiedono un nome, utilizziamo il prefisso `ConsoleTutorial` per assicurarci che tutte abbiano un nome univoco e per renderle facili da individuare.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creazione di un cluster Amazon ECS](#)
- [Fase 2: Registrazione di una definizione di attività](#)

- [Fase 3: esecuzione di un'attività](#)
- [Fase 4: verifica](#)
- [Fase 5: rimozione](#)

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy [Amazon ECS_FullAccess](#) IAM.
- Viene creato il ruolo IAM dell'istanza di container Amazon ECS. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).
- Viene creato il ruolo IAM collegato ai servizi Amazon ECS. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Viene creato il ruolo IAM collegato al servizio Auto Scaling. Per ulteriori informazioni, consulta [Service-Linked Roles for Amazon Auto EC2 Scaling nella Amazon Auto EC2](#) Scaling User Guide.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).

Fase 1: Creazione di un cluster Amazon ECS

Utilizza la procedura seguente per creare un cluster Amazon ECS.

Amazon ECS crea un modello di avvio di Amazon EC2 Auto Scaling e un gruppo Auto Scaling per tuo conto come parte dello stack. AWS CloudFormation

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel riquadro di navigazione scegli Cluster, quindi Crea cluster.
3. In Configurazione del cluster, per Nome cluster, inserisci ConsoleTutorial-cluster.
4. In Infrastruttura, deseleziona AWS Fargate (serverless), quindi seleziona Istanze Amazon EC2. Successivamente, configura il gruppo con scalabilità automatica che funge da provider di capacità.
 - In Gruppo Auto Scaling (ASG). Seleziona Crea nuovo ASG, quindi fornisci i seguenti dettagli relativi al gruppo:

- Per Sistema operativo/architettura, seleziona Amazon Linux 2.
 - EC2 Ad esempio, scegli t3.nano.
 - In Capacity (Capacità), inserisci il numero minimo e massimo di istanze da avviare nel gruppo Auto Scaling.
5. (Facoltativo) Per gestire i tag cluster, espandi Tags (Tag), quindi esegui una delle seguenti operazioni:

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:

- In Chiave, immetti il nome della chiave.
- In Valore, immetti il valore della chiave.

[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.

6. Scegli Create (Crea).

Fase 2: Registrazione di una definizione di attività

Prima di eseguire un'attività nel cluster, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente illustra una semplice definizione di attività che utilizza un'immagine `amazonlinux` da Docker Hub ed è in sospensione. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta [Definizioni dei processi di Amazon ECS](#).

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).
4. Nella casella Editor JSON, incolla i seguenti contenuti.

```
{
  "family": "ConsoleTutorial-taskdef",
  "containerDefinitions": [
    {
      "name": "sleep",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
      "memory": 20,
```

```
        "essential": true,
        "command": [
            "sh",
            "-c",
            "sleep infinity"
        ]
    },
    "requiresCompatibilities": [
        "EC2"
    ]
}
```

5. Scegli Create (Crea).

Fase 3: esecuzione di un'attività

Dopo aver registrato una definizione di attività per l'account, puoi eseguire un'attività nel cluster. Per questo tutorial, esegui cinque istanze della definizione di attività `ConsoleTutorial-taskdef` nel cluster `ConsoleTutorial-cluster`.

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Clusters, scegli `ConsoleTutorial -cluster`.
3. In Attività, scegli Esegui nuova attività.
4. Nella sezione Ambiente, in Opzioni di calcolo, scegli Strategia del provider di capacità.
5. In Configurazione dell'implementazione, per Tipo di applicazione, scegli Attività.
6. Scegli `ConsoleTutorial-taskdef` dall'elenco a discesa Family.
7. In Attività desiderate, digita 5.
8. Scegli Create (Crea).

Fase 4: verifica

A questo punto del tutorial dovresti disporre di un cluster con cinque attività in esecuzione e un gruppo con dimensionamento automatico con un provider di capacità. Il provider di capacità ha il dimensionamento gestito di Amazon ECS abilitato.

Possiamo verificare che tutto funzioni correttamente visualizzando le CloudWatch metriche, le impostazioni del gruppo Auto Scaling e infine il conteggio delle attività del cluster Amazon ECS.

Per visualizzare le CloudWatch metriche per il tuo cluster

1. Apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
3. Nel pannello di navigazione, in Parametri, scegli Tutti i parametri.
4. Nella pagina Tutti i parametri, all'interno della scheda Sfoglia, scegli AWS/ECS/ManagedScaling.
5. Scegli CapacityProviderName, ClusterName.
6. Seleziona la casella di controllo corrispondente a ConsoleTutorial-cluster ClusterName.
7. Nella scheda Parametri definiti, modifica Periodo in 30 secondi e Statistica in Massimo.

Il valore visualizzato nel grafico mostra il valore di capacità target per il provider di capacità. Inizia da 100, la percentuale di capacità target impostata. Osservare l'incremento fino a 200, che attiva un allarme per la policy di dimensionamento del monitoraggio dei target. L'allarme attiverà il dimensionamento orizzontale del gruppo Auto Scaling.

Utilizza la procedura seguente per visualizzare i dettagli del gruppo Auto Scaling per verificare che l'operazione di dimensionamento orizzontale sia stata eseguita.

Come verificare il gruppo Auto Scaling dimensionato orizzontalmente

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
3. Nel pannello di navigazione, nella sezione Dimensionamento automatico, seleziona Gruppi con dimensionamento automatico.
4. Scegli il gruppo con dimensionamento automatico ConsoleTutorial-cluster creato in questo tutorial. Visualizza il valore in Capacità desiderata e visualizza le istanze nella scheda Gestione delle istanze per confermare che il gruppo è stato dimensionato a due istanze.

Utilizza i seguenti passaggi per visualizzare il tuo cluster Amazon ECS e confermare che le EC2 istanze Amazon sono state registrate nel cluster e che le tue attività sono passate a uno stato. RUNNING

Per verificare le istanze nel gruppo Auto Scaling

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)

2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster `ConsoleTutorial-cluster`.
4. Nella scheda Attività verifica che siano visualizzate cinque attività nello stato `RUNNING`.

Fase 5: rimozione

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi. L'eliminazione dei provider di capacità e delle definizioni di attività non è supportata, ma non sono associati costi a queste risorse.

Per eliminare le risorse del tutorial

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters, scegli `ConsoleTutorial-cluster`.
4. Nella pagina `ConsoleTutorial-cluster`, scegli la scheda Attività, quindi scegli Stop, Arresta tutto.
5. Nel pannello di navigazione scegliere Clusters (Cluster).
6. Nella pagina Clusters, scegli `ConsoleTutorial-cluster`.
7. In alto a destra della pagina, scegli Elimina cluster.
8. Nella casella di conferma, inserisci `delete ConsoleTutorial-cluster` e scegli Elimina.
9. Elimina i gruppi Auto Scaling completando la seguente procedura.
 - a. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
 - b. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
 - c. Nel pannello di navigazione, nella sezione Dimensionamento automatico, seleziona Gruppi con dimensionamento automatico.
 - d. Seleziona il gruppo con dimensionamento automatico `ConsoleTutorial-cluster`, quindi scegli Operazioni.
 - e. Nel menu Actions (Operazioni) selezionare Delete (Elimina). Nella casella di conferma, immetti elimina e scegli Elimina.

Istanze di EC2 container Amazon per Amazon ECS

Un'istanza di container Amazon ECS è un' EC2 istanza Amazon che esegue l'agente container Amazon ECS ed è registrata in un cluster. Quando esegui attività con Amazon ECS utilizzando il

tipo di EC2 avvio, il tipo di avvio esterno o un fornitore di capacità di gruppo Auto Scaling, le tue attività vengono assegnate alle istanze di container attive. Sei responsabile della gestione e della manutenzione delle istanze di container.

Sebbene tu possa creare la tua AMI di EC2 istanza Amazon che soddisfi le specifiche di base necessarie per eseguire i carichi di lavoro containerizzati su Amazon ECS, le AMI ottimizzate per Amazon ECS sono AMIs preconfigurate e testate su Amazon ECS da ingegneri. AWS È il modo più semplice per iniziare ed eseguire i container su AWS rapidamente.

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando lo utilizzi AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di EC2 lancio di Amazon per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisce lo stack, verifichi se sono disponibili aggiornamenti da applicare alle istanze. EC2 Per un esempio di come utilizzare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per AWS CloudFormation l'utente.

- [Recupero di metadati AMI Linux ottimizzati per Amazon ECS](#)
- [Recupero ottimizzato per Amazon ECS Bottlerocket Metadati AMI](#)
- [Recupero di metadati AMI Windows ottimizzati per Amazon ECS](#)

Puoi scegliere tra i tipi di istanza compatibili con la tua applicazione. Con istanze più grandi, puoi avviare più attività contemporaneamente. Con istanze più piccole, puoi scalare orizzontalmente in modo più dettagliato per risparmiare sui costi. Non è necessario scegliere un singolo tipo di EC2 istanza Amazon adatto a tutte le applicazioni del cluster. È invece possibile creare più gruppi di Auto Scaling in cui ogni gruppo ha un tipo di istanza diverso. Quindi, puoi creare un provider EC2 di capacità Amazon per ognuno di questi gruppi.

Utilizza le seguenti linee guida per determinare i tipi di famiglie di istanze e il tipo di istanza da utilizzare:

- Elimina i tipi o le famiglie di istanze che non soddisfano i requisiti specifici della tua applicazione. Ad esempio, se l'applicazione richiede una GPU, puoi escludere tutti i tipi di istanza che non dispongono di una GPU.
- Prendi in considerazione i requisiti, tra cui la velocità di trasmissione e lo storage di rete.

- Considerate la CPU e la memoria. Come regola generale, la CPU e la memoria devono essere sufficientemente grandi da contenere almeno una replica dell'attività che si desidera eseguire.

Spot Instances

La capacità spot può offrire risparmi significativi sui costi rispetto alle istanze on demand. La capacità spot è una capacità in eccesso a un prezzo notevolmente inferiore rispetto a quella on demand o riservata. La capacità spot è adatta per carichi di lavoro di elaborazione in batch e machine learning ambienti di sviluppo e staging. Più in generale, è adatto a qualsiasi carico di lavoro che tollera tempi di inattività temporanei.

Tieni presente le seguenti conseguenze, poiché la capacità spot potrebbe non essere sempre disponibile.

- Nei periodi con domanda estremamente elevata, la capacità spot potrebbe non essere disponibile. Ciò può causare ritardi nell'avvio delle istanze Amazon EC2 Spot. In questi eventi, i servizi Amazon ECS riprovano a lanciare le attività e anche i gruppi Amazon EC2 Auto Scaling riprovano a lanciare le istanze, finché la capacità richiesta non diventa disponibile. Amazon EC2 non sostituisce la capacità Spot con la capacità on demand.
- Quando la domanda complessiva di capacità aumenta, le istanze e le attività Spot potrebbero essere chiuse con un preavviso di soli due minuti. Dopo l'invio dell'avviso, le attività dovrebbero iniziare a chiudersi regolarmente, se necessario, prima che l'istanza venga terminata completamente. Questo aiuta a ridurre al minimo la possibilità di errori. Per ulteriori informazioni su un arresto corretto, consulta [Arresto regolare con ECS](#).

Per ridurre al minimo le carenze di capacità spot, considera i seguenti consigli:

- Utilizza più regioni e zone di disponibilità: la capacità spot varia in base alla regione e alla zona di disponibilità. Puoi migliorare la disponibilità spot eseguendo i carichi di lavoro in più regioni e zone di disponibilità. Se possibile, specifica le sottoreti in tutte le zone di disponibilità nelle regioni in cui esegui le attività e le istanze.
- Usa più tipi di EC2 istanze Amazon: quando utilizzi le politiche di istanze miste con Amazon EC2 Auto Scaling, nel tuo gruppo Auto Scaling vengono lanciati più tipi di istanze. Ciò garantisce che una richiesta di capacità spot possa essere soddisfatta quando necessario. Per massimizzare l'affidabilità e ridurre al minimo la complessità, nella tua policy sulle istanze miste utilizza tipi di istanze con all'incirca la stessa quantità di CPU e memoria. Queste istanze possono appartenere a una generazione diversa o varianti dello stesso tipo di istanza di base. Tieni presente che

potrebbero essere dotate di funzionalità aggiuntive di cui non hai bisogno. Un esempio di tale elenco potrebbe includere m4.large, m5.large, m5a.large, m5d.large, m5n.large, m5dn.large e m5ad.large. Per ulteriori informazioni, consulta i [gruppi di Auto Scaling con più tipi di istanze e opzioni di acquisto](#) nella Amazon Auto EC2 Scaling User Guide.

- Utilizza la strategia di allocazione Spot ottimizzata per la capacità: con Amazon EC2 Spot, puoi scegliere tra strategie di allocazione ottimizzate in termini di capacità e costi. Se scegli la strategia di ottimizzazione della capacità al momento del lancio di una nuova istanza, Amazon EC2 Spot seleziona il tipo di istanza con la massima disponibilità nella zona di disponibilità selezionata. Ciò aiuta a ridurre la possibilità che l'istanza venga terminata subito dopo l'avvio.

Per informazioni su come configurare gli avvisi di terminazione spot sulle istanze di container, consulta:

- [Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot](#)
- [Configurazione delle istanze di container Amazon ECS Windows per ricevere avvisi sulle istanze Spot](#)

Linux ottimizzato per Amazon ECS AMIs

Amazon ECS fornisce soluzioni ottimizzate per Amazon ECS AMIs preconfigurate con i requisiti e i consigli per eseguire i carichi di lavoro dei container. Ti consigliamo di utilizzare l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS per le tue EC2 istanze Amazon, a meno che l'applicazione non richieda istanze EC2 basate su Amazon GPU, un sistema operativo specifico o una versione Docker non ancora disponibile in quell'AMI. Per informazioni sulle istanze Amazon Linux 2 e Amazon Linux 2023, consulta la sezione [Confronto tra Amazon Linux 2 e Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023. L'avvio delle istanze di container dall'AMI ottimizzata per Amazon ECS più recente ti assicura di ricevere gli aggiornamenti di sicurezza e la versione dell'agente del container corrente. Per ulteriori informazioni sull'avvio di un'istanza, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando lo utilizzi AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di EC2 lancio di Amazon per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisce lo stack, verifichi

se sono disponibili aggiornamenti da applicare alle istanze. EC2 Per un esempio di come utilizzare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per AWS CloudFormation l'utente.

Se devi personalizzare l'AMI ottimizzata per Amazon ECS, consulta [Amazon ECS Optimized AMI Build Recipes](#) su. GitHub

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 come base. Sono disponibili anche le note di rilascio di AMI di Amazon Linux 2. Per ulteriori informazioni, consulta [Note di rilascio di Amazon Linux 2](#).

Si consiglia di utilizzare un'AMI con Linux kernel 5.10 perché il kernel Linux 4.14 è stato raggiunto il 10 end-of-life gennaio 2024.

Le seguenti varianti dell'AMI ottimizzata per Amazon ECS sono disponibili per le tue EC2 istanze Amazon con il sistema operativo Amazon Linux 2023.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2023	AMI Amazon Linux 2023 ottimizzata per Amazon ECS	Amazon Linux 2023 è la nuova generazione di Amazon Linux di AWS. Nella maggior parte dei casi, consigliato per avviare le EC2 istanze Amazon per i carichi di lavoro Amazon ECS. Per ulteriori informazioni, consulta la sezione Che cos'è Amazon Linux 2023 nella Guida per l'utente di Amazon Linux 2023.	Di default, l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è fornita con un singolo volume root di 30 GiB. Puoi modificar e le dimensioni del volume root di 30 GiB al momento dell'avvi o per aumentare lo storage disponibili nell'istanza di container. Questo storage è utilizzat o per il sistema operativo e per le

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2023 (arm64)	AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS	Basata su Amazon Linux 2023, questa AMI è consigliata per l'avvio di EC2 istanze Amazon, alimentate da AWS Graviton/Graviton 2/Graviton 3/Graviton 4 processor i basati su ARM, per i carichi di lavoro Amazon ECS. Per ulteriori informazioni, consulta le specifiche per le istanze Amazon EC2 generiche nella guida Amazon EC2 Instance Types .	Di default, l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è fornita con un singolo volume root di 30 GiB. Puoi modificarne le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2023 (Neuron)	AMI Amazon Linux 2023 ottimizzata per Amazon ECS	Basato su Amazon Linux 2023, questo AMI è per istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfigurato con i driver AWS Inferentia e AWS Trainium e il runtime AWS Neuron per Docker, che semplifica l'esecuzione di carichi di lavoro di inferenza di machine learning su Amazon ECS. Per ulteriori informazioni, consulta Definizione delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron. L'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	Di default, l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è fornita con un singolo volume root di 30 GiB. Puoi modificarne le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
GPU Amazon Linux 2023	AMI GPU Amazon Linux 2023 ottimizzata per Amazon ECS	Basata su Amazon Linux 2023, questa AMI è consigliata per l'avvio di istanze EC2 basate su Amazon GPU per i carichi di lavoro Amazon ECS. È preconfigurato con i driver del kernel NVIDIA e un runtime GPU Docker che consente l'esecuzione di carichi di lavoro che sfruttano su Amazon ECS. GPUs Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU .	Di default, l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è fornita con un singolo volume root di 30 GiB. Puoi modificarne le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Le seguenti varianti dell'AMI ottimizzata per Amazon ECS sono disponibili per le tue EC2 istanze Amazon con il sistema operativo Amazon Linux 2.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2	AMI Amazon Linux 2 kernel 5.10 ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è destinata all'avvio EC2 delle istanze Amazon e desideri utilizzare il kernel Linux 5.10 anziché il kernel 4.14 per i tuoi carichi di lavoro Amazon ECS. L'AMI Amazon Linux 2 kernel 5.10 ottimizzata per Amazon ECS non viene fornita con la AWS CLI preinstallata.	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2	AMI Amazon Linux 2 ottimizzata per Amazon ECS	Questo è per i tuoi carichi di lavoro Amazon ECS. L'AMI Amazon Linux 2 ottimizzata per Amazon ECS non viene fornita con AWS CLI preinstallato.	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (arm64)	AMI Amazon Linux 2 (arm64) kernel 5.10 ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è per le tue EC2 istanze Amazon, che sono alimentate da AWS Graviton/Graviton 2/Graviton 3/Graviton 4 processor i basati su ARM, e desideri utilizzare il kernel Linux 5.10 anziché il kernel Linux 4.14 per i tuoi carichi di lavoro Amazon ECS. Per ulteriori informazioni, consulta la sezione Specifiche e per le istanze EC2 generiche di Amazon nella guida Amazon EC2 Instance Types. L'AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (arm64)	AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è destinata all'avvio di EC2 istanze Amazon, alimentate da AWS Graviton/Graviton 2/Graviton 3/Graviton 4 processori basati su ARM, per i carichi di lavoro Amazon ECS. L'AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (GPU)	AMI kernel 5.10 ottimizzata per GPU Amazon ECS	Basata su Amazon Linux 2, questa AMI è consigliata per l'avvio di istanze EC2 basate su Amazon GPU con Linux kernel 5.10 per i carichi di lavoro Amazon ECS. È preconfigurato con i driver del kernel NVIDIA e un runtime GPU Docker che consente l'esecuzione di carichi di lavoro che sfruttano su Amazon ECS. GPUs Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU .	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (GPU)	AMI Amazon ECS ottimizzata per GPU	Basata su Amazon Linux 2, questa AMI è consigliata per l'avvio di istanze EC2 basate su Amazon GPU con kernel Linux 4.14 per i carichi di lavoro Amazon ECS. È preconfigurato con i driver del kernel NVIDIA e un runtime GPU Docker che consente l'esecuzione di carichi di lavoro che sfruttano su Amazon ECS. GPUs Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU .	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (Neuron)	AMI Amazon ECS per il kernel 5.10 di Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è per istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfigurato con AWS Inferentia con kernel Linux 5.10 e driver AWS Trainium e il runtime AWS Neuron per Docker, che semplifica l'esecuzione di carichi di lavoro di inferenza di machine learning su Amazon ECS. Per ulteriori informazioni, consulta Definizioni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron . L'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (Neuron)	AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è per istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfigurato con i driver AWS Inference e AWS Trainium e il runtime AWS Neuron per Docker, che semplifica l'esecuzione di carichi di lavoro di inferenza di machine learning su Amazon ECS. Per ulteriori informazioni, consulta Definizione delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron . L'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	Per impostazione predefinita, l'AMI Amazon ECS ottimizzata per Amazon Linux 2 (AMI AMIs Amazon Linux 2 ottimizzata per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS e AMI ottimizzata per GPU Amazon ECS) viene fornita con un singolo volume root da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio o per aumentare lo storage disponibile nell'istanza di container. Questo storage è utilizzato per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS è

Sistema operativo	AMI	Descrizione	Configurazione dello storage
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Amazon ECS fornisce un changelog per la variante Linux dell'AMI ottimizzata per Amazon ECS su GitHub. Per ulteriori informazioni, consulta [Changelog](#).

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 o Amazon Linux 2023 come base. Puoi recuperare il nome dell'AMI di origine di Amazon Linux 2 o Amazon Linux 2023 per ogni variante eseguendo query sull'API Archivio dei parametri Systems Manager. Per ulteriori informazioni, consulta [Recupero di metadati AMI Linux ottimizzati per Amazon ECS](#). Sono disponibili anche le note di rilascio di AMI di Amazon Linux 2. Per ulteriori informazioni, consulta [Note di rilascio di Amazon Linux 2](#). Sono disponibili anche le note di rilascio di Amazon Linux 2023. Per ulteriori informazioni, consulta [Note di rilascio di Amazon Linux 2023](#).

Nelle pagine seguenti vengono fornite ulteriori informazioni sulle modifiche:

- Note di [rilascio di Source AMI](#) su GitHub
- [Note di rilascio di Docker Engine](#) nella documentazione Docker
- [Documentazione dei driver NVIDIA](#) nella documentazione NVIDIA
- Registro delle [modifiche dell'agente Amazon ECS attivo](#) GitHub

Il codice sorgente dell'applicazione `ecs-init`, gli script e la configurazione per il pacchetto dell'agente fanno ora parte del repository dell'agente. Per le versioni `ecs-init` e i pacchetti precedenti di Amazon `ecs-init`, consulta il changelog di [Amazon ecs-init](#) su GitHub

Applicazione degli aggiornamenti di sicurezza all'AMI ottimizzata per Amazon ECS

I programmi ottimizzati per Amazon ECS AMIs basati su Amazon Linux contengono una versione personalizzata di cloud-init. Cloud-init è un pacchetto che viene utilizzato per avviare le immagini Linux in un ambiente di cloud computing ed eseguire le azioni desiderate all'avvio di un'istanza. Per impostazione predefinita, tutti gli aggiornamenti di sicurezza «critici» e «importanti» applicati all'avvio dell'istanza sono applicati per impostazione predefinita su tutti gli aggiornamenti di sicurezza ottimizzati per Amazon ECS AMIs basati su Amazon Linux e rilasciati prima del 12 giugno 2024.

A partire dalle versioni del 12 giugno 2024 di Amazon ECS ottimizzato AMIs sulla base di Amazon Linux 2, il comportamento predefinito non includerà più l'aggiornamento dei pacchetti al momento del lancio. Ti consigliamo invece di eseguire l'aggiornamento a una nuova AMI ottimizzata per Amazon ECS non appena le versioni vengono rese disponibili. Le versioni ottimizzate per Amazon ECS AMIs vengono rilasciate quando sono disponibili aggiornamenti di sicurezza o modifiche alle AMI di base. In questo modo avrai la certezza di ricevere le versioni più recenti dei pacchetti e gli aggiornamenti di sicurezza e che le versioni dei pacchetti siano immutabili al momento del lancio dell'istanza. Per ulteriori informazioni sul recupero dell'ultima AMI ottimizzata per Amazon ECS, consulta [Recupero di metadati AMI Linux ottimizzati per Amazon ECS](#)

Ti consigliamo di automatizzare l'ambiente per l'aggiornamento a una nuova AMI non appena vengono rese disponibili. Per informazioni sulle opzioni disponibili, consulta [Amazon ECS consente una gestione più semplice EC2 della capacità, con il drenaggio gestito delle istanze](#).

Per continuare ad applicare manualmente gli aggiornamenti di sicurezza «Critici» e «Importanti» su una versione AMI, puoi eseguire il seguente comando sulla tua EC2 istanza Amazon.

```
yum update --security
```

Se desideri riattivare gli aggiornamenti di sicurezza all'avvio, puoi aggiungere la seguente riga alla `#cloud-config` sezione dei dati utente cloud-init all'avvio dell'istanza Amazon. EC2 Per ulteriori informazioni, consulta [Using cloud-init su Amazon Linux 2 nella Amazon Linux User Guide](#).

```
#cloud-config
repo_upgrade: security
```

Recupero di metadati AMI Linux ottimizzati per Amazon ECS

Puoi recuperare a livello di codice i metadati AMI ottimizzati per Amazon ECS. I metadati includono il nome AMI, la versione dell'agente container Amazon ECS e la versione runtime di Amazon ECS che include la versione Docker.

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando lo utilizzi AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di EC2 lancio di Amazon per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisce lo stack, verifichi se sono disponibili aggiornamenti da applicare alle istanze. EC2 Per un esempio di come utilizzare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per AWS CloudFormation l'utente.

L'ID AMI, il nome dell'immagine, il sistema operativo, la versione dell'agente contenitore, il nome dell'immagine di origine e la versione di runtime per ogni variante di Amazon ECS Optimized AMIs possono essere recuperati a livello di codice interrogando l'API Systems Manager Parameter Store. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere [GetParameterse GetParametersByPath](#).

Note

Per recuperare i metadati dell'AMI ottimizzata per Amazon ECS, l'utente di amministrazione deve disporre delle seguenti autorizzazioni IAM. Queste autorizzazioni sono state aggiunte alla policy IAM AmazonECS_FullAccess.

- ssm: GetParameters
- ssm: GetParameter
- ssm: GetParametersByPath

Formato del parametro dell'archivio parametri di Systems Manager

Di seguito è riportato il formato del nome del parametro per ogni variante AMI ottimizzata per Amazon ECS.

Ottimizzato per Linux Amazon ECS AMIs

- Metadati dell'AMI Amazon Linux 2023:

```
/aws/service/ecs/optimized-ami/amazon-linux-2023/<version>
```

- Metadati dell'AMI Amazon Linux 2023 (arm64):

```
/aws/service/ecs/optimized-ami/amazon-linux-2023/arm64/<version>
```

- Metadati dell'AMI Amazon Linux 2023 (Neuron):

```
/aws/service/ecs/optimized-ami/amazon-linux-2023/neuron/<version>
```

- Metadati AMI Amazon Linux 2023 (GPU):

```
/aws/service/ecs/optimized-ami/amazon-linux-2023/gpu/<version>
```

Metadati AMI Amazon Linux 2:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/<version>
```

- Metadati AMI Amazon Linux 2 kernel 5.10:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/<version>
```

- Metadati AMI Amazon Linux 2 (arm64):

```
/aws/service/ecs/optimized-ami/amazon-linux-2/arm64/<version>
```

- Metadati AMI Amazon Linux 2 (arm64) kernel 5.10:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/arm64/<version>
```

- Metadati AMI del kernel 5.10 ottimizzati per GPU Amazon ECS:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/gpu/<version>
```

- Metadati AMI Amazon Linux 2 (GPU):

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/<version>
```

- Metadati AMI Amazon Linux 2 (Neuron) del kernel 5.10 ottimizzati per Amazon ECS:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/inf/<version>
```

- Metadati dell'AMI Amazon Linux 2 (Neuron):

```
/aws/service/ecs/optimized-ami/amazon-linux-2/inf/<version>
```

Il seguente formato del nome del parametro recupera l'ID immagine dell'ultima AMI Amazon Linux 2 ottimizzata per Amazon ECS consigliata utilizzando il sottoparametro. `image_id`

```
/aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id
```

Il formato dei nomi di parametro seguente recupera i metadati di una versione di AMI ottimizzata per Amazon ECS specifica indicando il nome dell'AMI.

- Metadati dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/amzn2-ami-ecs-hvm-2.0.20181112-x86_64-ebs
```

Note

Tutte le versioni dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS sono disponibili per il recupero. Possono essere recuperate solo le versioni `amzn-ami-2017.09.1-amazon-ecs-optimized` (Linux) dell'AMI ottimizzata per Amazon ECS e successive.

Esempi

Negli esempi seguenti vengono illustrati i modi in cui è possibile recuperare i metadati per ogni variante dell'AMI ottimizzata per Amazon ECS.

Recupero dei metadati dell'ultima AMI ottimizzata per Amazon ECS consigliata

Puoi recuperare l'ultima AMI ottimizzata per Amazon ECS consigliata utilizzando AWS CLI i AWS CLI seguenti comandi.

Ottimizzato per Linux Amazon ECS AMIs

- Per Amazon Linux 2023 ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/  
recommended --region us-east-1
```

- Per Amazon Linux 2023 (arm64) ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/  
arm64/recommended --region us-east-1
```

- Per Amazon Linux 2023 (Neuron) ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/  
neuron/recommended --region us-east-1
```

- Per la GPU Amazon Linux 2023 ottimizzata per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/gpu/  
recommended --region us-east-1
```

- Per il kernel Amazon Linux 2 5.10 ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/  
kernel-5.10/recommended --region us-east-1
```

- Per Amazon Linux 2 ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/  
recommended --region us-east-1
```

- Per il kernel Amazon Linux 2 5.10 (arm64) ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/  
kernel-5.10/arm64/recommended --region us-east-1
```

- Per Amazon Linux 2 (arm64) ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/arm64/  
recommended --region us-east-1
```

- Per il kernel 5.10 ottimizzato per GPU Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/gpu/recommended --region us-east-1
```

- Per Amazon ECS ottimizzato per AMIs GPU:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended --region us-east-1
```

- Per il kernel Amazon Linux 2 (Neuron) 5.10 ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/inf/recommended --region us-east-1
```

- Per Amazon Linux 2 (Neuron) ottimizzato per Amazon ECS: AMIs

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/inf/
recommended --region us-east-1
```

Recupero dell'ID immagine dell'AMI Amazon Linux 2023 ottimizzata per Amazon ECS consigliata più recente

Puoi recuperare l'ID immagine dell'ID dell'AMI Amazon Linux 2023 ottimizzata per Amazon ECS consigliata più recente utilizzando il parametro secondario `image_id`.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-
linux-2023/recommended/image_id --region us-east-1
```

Per recuperare solo il valore `image_id`, è possibile eseguire query sul valore di parametro specifico, ad esempio:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/
recommended/image_id --region us-east-1 --query "Parameters[0].Value"
```

Recupero dei metadati di una versione specifica dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS

Recupera i metadati di una versione specifica dell'AMI Amazon Linux ottimizzata per Amazon ECS utilizzando AWS CLI il comando seguente. AWS CLI Sostituisci il nome dell'AMI con il nome dell'AMI ottimizzata per Amazon ECS da recuperare.


```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/amzn2-ami-ecs-hvm-2.0.20200928-x86_64-ebs --region us-east-1
```

Recupero dei metadati AMI Amazon Linux 2 kernel 5.10 ottimizzati per Amazon ECS utilizzando l'API Systems Manager GetParametersByPath

Recupera i metadati AMI Amazon Linux 2 ottimizzati per Amazon ECS con l'API Systems Manager GetParametersByPath utilizzando AWS CLI il comando seguente.

```
aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/ --region us-east-1
```

Recupero dell'ID immagine dell'ultima AMI Amazon Linux 2 kernel 5.10 consigliata e ottimizzata per Amazon ECS

Puoi recuperare l'ID immagine dell'ultimo ID AMI Amazon Linux Linux 2 kernel 5.10 consigliato per Amazon ECS utilizzando il sottoparametro. `image_id`

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/recommended/image_id --region us-east-1
```

Per recuperare solo il valore `image_id`, è possibile eseguire query sul valore di parametro specifico, ad esempio:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/recommended/image_id --region us-east-1 --query "Parameters[0].Value"
```

Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello AWS CloudFormation facendo riferimento al nome dell'archivio parametri di Systems Manager.

Esempio per Linux

```
Parameters:kernel-5.10
LatestECSOptimizedAMI:
  Description: AMI ID
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/recommended/image_id
```

Script di build per AMI Linux ottimizzata per Amazon ECS

Amazon ECS ha reso open-source gli script della build che vengono utilizzati per creare le varianti Linux dell'AMI ottimizzata per Amazon ECS. Questi script di build sono ora disponibili su GitHub. Per ulteriori informazioni, vedere [amazon-ecs-ami](#) su GitHub.

Se devi personalizzare l'AMI ottimizzata per Amazon ECS, consulta Amazon [ECS Optimized AMI Build Recipes](#) on GitHub.

L'archivio degli script di compilazione include un modello di [HashiCorp packer](#) e script di compilazione per generare ciascuna delle varianti Linux dell'AMI ottimizzata per Amazon ECS. Questi script sono la fonte di verità per le build AMI ottimizzate per Amazon ECS, quindi puoi seguire GitHub il repository per monitorare le modifiche al nostro. Ad esempio, magari vuoi che l'AMI utilizzi la stessa versione di Docker utilizzata dal team Amazon ECS per l'AMI ufficiale.

Per ulteriori informazioni, consulta il repository AMI Amazon ECS all'indirizzo [amazon-ecs-amiaws/](#) on GitHub.

Come creare un'AMI Linux ottimizzata per Amazon ECS

1. Clona il repository. `aws/amazon-ecs-ami` GitHub

```
git clone https://github.com/aws/amazon-ecs-ami.git
```

2. Aggiungi una variabile di ambiente per la AWS regione da utilizzare durante la creazione dell'AMI. Sostituisci il valore `us-west-2` con la regione da utilizzare.

```
export REGION=us-west-2
```

3. Viene fornito un Makefile per creare l'AMI. Dalla directory principale del repository clonato, utilizza uno dei seguenti comandi, corrispondente alla variante Linux dell'AMI ottimizzata per Amazon ECS che desideri creare.

- AMI Amazon Linux 2 ottimizzata per Amazon ECS

```
make al2
```

- AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS

```
make al2arm
```

- AMI Amazon ECS ottimizzata per GPU

```
make a12gpu
```

- AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS

```
make a12inf
```

- AMI Amazon Linux 2023 ottimizzata per Amazon ECS

```
make a12023
```

- AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS

```
make a12023arm
```

- AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS

```
make a12023neu
```

Ottimizzato per Amazon ECS Bottlerocket AMIs

Bottlerocket è un Linux sistema operativo open source basato sullo scopo di AWS eseguire container su macchine virtuali o host bare metal. Ottimizzato per Amazon ECS Bottlerocket L'AMI è sicuro e include solo il numero minimo di pacchetti necessari per eseguire i container. Ciò migliora l'utilizzo delle risorse, riduce la superficie di attacco della sicurezza e aiuta a ridurre il sovraccarico di gestione. Il Bottlerocket L'AMI è inoltre integrato con Amazon ECS per contribuire a ridurre il sovraccarico operativo legato all'aggiornamento delle istanze di container in un cluster.

Bottlerocket si differenzia da Amazon Linux nei seguenti modi:

- Bottlerocket non include un gestore di pacchetti e il suo software può essere eseguito solo come contenitore. Aggiornamenti a Bottlerocket sono entrambi applicati e possono essere ripristinati in un unico passaggio, il che riduce la probabilità di errori di aggiornamento.
- Il meccanismo principale da gestire Bottlerocket hosts è dotato di un container scheduler. A differenza di Amazon Linux, l'accesso a un utente Bottlerocket le istanze sono pensate per essere un'operazione poco frequente solo per scopi avanzati di debug e risoluzione dei problemi.

Per ulteriori informazioni sull' Bottlerocket, [consulta la documentazione e le versioni su GitHub](#).

Esistono varianti di Amazon ECS ottimizzate Bottlerocket AMI per kernel 6.1 e kernel 5.10.

Le seguenti varianti utilizzano il kernel 6.1:

- `aws-ecs-2`
- `aws-ecs-2-nvidia`

Le seguenti varianti utilizzano il kernel 5.10:

- `aws-ecs-1`
- `aws-ecs-1-nvidia`

Per ulteriori informazioni sulla `aws-ecs-1-nvidia` variante, consulta [Annuncio del supporto delle GPU NVIDIA per Bottlerocket su Amazon ECS](#).

Considerazioni

Considera quanto segue quando usi un Bottlerocket AMI con Amazon ECS.

- Bottlerocket supporta EC2 istanze Amazon con x86_64 arm64 processori. Il Bottlerocket L'uso dell'AMI non è consigliato con EC2 istanze Amazon con chip Inferentia.
- Bottlerocket le immagini non includono un server SSH o una shell. Tuttavia, è possibile utilizzare strumenti di out-of-band gestione per ottenere l'accesso come amministratore SSH ed eseguire il bootstrap. [Per ulteriori informazioni, consulta queste sezioni nel file README.md di bottlerocket su GitHub](#):
 - [Esplorazione](#)
 - [Container amministratore](#)
- Per impostazione predefinita, Bottlerocket ha un [contenitore di controllo](#) abilitato. Questo contenitore esegue l'[AWS Systems Manager agente](#) che puoi utilizzare per eseguire comandi o avviare sessioni di shell su Amazon EC2 Bottlerocket istanze. Per ulteriori informazioni, consulta [Configurazione di Session Manager](#) nella Guida per l'utente di AWS Systems Manager .
- Bottlerocket è ottimizzato per i carichi di lavoro dei container e si concentra sulla sicurezza. Bottlerocket non include un gestore di pacchetti ed è immutabile. Per informazioni sulle funzionalità

e sulle linee guida di sicurezza, vedere [Funzionalità di sicurezza](#) e [linee guida sulla sicurezza](#) su GitHub.

- La modalità di rete `aws_vpc` è supportata qper l'AMI Bottlerocket versione `1.1.0` o successiva.
- L'App Mesh in una definizione di attività è supportata per la versione `1.15.0` dell'AMI Bottlerocket o versioni successive.
- Il parametro di definizione dell'`initProcessEnabled`attività è supportato per la versione AMI Bottlerocket `1.19.0` o successiva.
- Il Bottlerocket AMIs inoltre non supportano i seguenti servizi e funzionalità:
 - ECS Anywhere
 - Service Connect
 - Amazon EFS in modalità crittografata
 - Amazon EFS in modalità `aws_vpc` di rete
 - Acceleratore di inferenze elastiche

Recupero ottimizzato per Amazon ECS Bottlerocket Metadati AMI

Puoi recuperare l'ID Amazon Machine Image (AMI) per Amazon ECS ottimizzato per Amazon ECS AMIs interrogando l' AWS Systems Manager API Parameter Store. Utilizzando questo parametro, non è necessario cercare manualmente le AMI ottimizzate per Amazon ECS. IDs Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere [GetParameter](#). Il principale IAM che utilizzi deve disporre dell'autorizzazione IAM `ssm:GetParameter` per recuperare i metadati dell'AMI ottimizzata per Amazon ECS.

aws-ecs-2 Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2` Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima versione ottimizzata per Amazon ECS consigliata Bottlerocket AMI con il seguente AWS CLI comando utilizzando il sottoparametro `image_id`. Sostituisci la *region* con il codice della regione per la quale desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub. Per recuperare una versione diversa da quella più recente, sostituisci `latest` con il numero della versione desiderata.
 - Per un'architettura a 64 bit (`x86_64`):

```
aws ssm get-parameter --region us-east-2 --name "/aws/service/bottlerocket/aws-ecs-2/x86_64/latest/image_id" --query Parameter.Value --output text
```

- Per l'architettura a 64 bit Arm (arm64):

```
aws ssm get-parameter --region us-east-2 --name "/aws/service/bottlerocket/aws-ecs-2/arm64/latest/image_id" --query Parameter.Value --output text
```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro. Nel seguente URL, sostituisci *region* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub.
- Per un'architettura a 64 bit (x86_64):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-2/x86_64/latest/image_id/description?region=region#
```

- Per l'architettura a 64 bit Arm (arm64):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-2/arm64/latest/image_id/description?region=region#
```

aws-ecs-2-nvidia Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2-nvidia` Bottlerocket per regione e architettura con AWS CLI o. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima versione ottimizzata per Amazon ECS consigliata Bottlerocket AMI con il seguente AWS CLI comando utilizzando il sottoparametro `image_id`. Sostituisci la *region* con il codice della regione per la quale desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub. Per recuperare una versione diversa da quella più recente, sostituisci `latest` con il numero della versione desiderata.
- Per un'architettura a 64 bit (x86_64):

```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-2-nvidia/x86_64/latest/image_id" --query Parameter.Value --output text
```

- Per l'architettura a 64 bit Arm (arm64):

```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-2-nvidia/arm64/latest/image_id" --query Parameter.Value --output text
```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro. Nel seguente URL, sostituisci *region* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub.

- Per l'architettura a 64 bit (x86_64):

```
https://regionconsole.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-2-nvidia/x86_64/latest/image_id/description?region=region#
```

- Per l'architettura a 64 bit Arm (arm64):

```
https://regionconsole.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-2-nvidia/arm64/latest/image_id/description?region=region#
```

aws-ecs-1 Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI aws-ecs-1 Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima versione ottimizzata per Amazon ECS consigliata Bottlerocket AMI con il seguente AWS CLI comando utilizzando il sottoparametro `image_id`. Sostituisci la *region* con il codice della regione per la quale desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub. Per recuperare una versione diversa da quella più recente, sostituisci `latest` con il numero della versione desiderata.
- Per un'architettura a 64 bit (x86_64):

```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-1/x86_64/latest/image_id" --query Parameter.Value --output text
```

- Per l'architettura a 64 bit Arm (arm64):

```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-1/arm64/latest/image_id" --query Parameter.Value --output text
```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro. Nel seguente URL, sostituisci *region* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub.

- Per un'architettura a 64 bit (x86_64):

```
https://region.console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-1/x86_64/latest/image_id/description
```

- Per l'architettura a 64 bit Arm (arm64):

```
https://region.console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-1/arm64/latest/image_id/description
```

aws-ecs-1-nvidia Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-1-nvidia` Bottlerocket per regione e architettura con AWS CLI o. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima versione ottimizzata per Amazon ECS consigliata Bottlerocket AMI con il seguente AWS CLI comando utilizzando il sottoparametro `image_id`. Sostituisci la *region* con il codice della regione per la quale desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub. Per recuperare una versione diversa da quella più recente, sostituisci `latest` con il numero della versione desiderata.
- Per un'architettura a 64 bit (x86_64):


```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-1-nvidia/x86_64/latest/image_id" --query Parameter.Value --output text
```

- Per l'architettura a 64 bit Arm (arm64):

```
aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-ecs-1-nvidia/arm64/latest/image_id" --query Parameter.Value --output text
```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro. Nel seguente URL, sostituisci *region* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta [Ricerca di un'AMI](#) attiva GitHub.
- Per l'architettura a 64 bit (x86_64):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-1-nvidia/x86_64/latest/image_id/description?region=region#
```

- Per l'architettura a 64 bit Arm (arm64):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-1-nvidia/arm64/latest/image_id/description?region=region#
```

Passaggi successivi

Per un tutorial dettagliato su come iniziare a usare Bottlerocket sistema operativo su Amazon ECS, consulta [Utilizzo di un'AMI Bottlerocket con Amazon ECS](#) attivo e Guida GitHub introduttiva [Bottlerocket e Amazon ECS](#) sul sito del AWS blog.

Per informazioni su come avviare un'istanza Bottlerocket, consulta [Avvio di un Bottlerocket istanza per Amazon ECS](#)

Avvio di un Bottlerocket istanza per Amazon ECS

Puoi lanciare un Bottlerocket istanza in modo da poter eseguire i carichi di lavoro dei container.

È possibile utilizzare il AWS CLI per avviare il Bottlerocket istanza.

1. Crea un file denominato `userdata.toml`. Questo file viene utilizzato per i dati utente dell'istanza. Sostituisci *cluster-name* con il nome del cluster.

```
[settings.ecs]
cluster = "cluster-name"
```

2. Usa uno dei comandi inclusi in [the section called “Recupero ottimizzato per Amazon ECS Bottlerocket Metadati AMI”](#) per ottenere il Bottlerocket ID AMI. Ti servirà per la fase successiva.
3. Esegui il seguente comando per lanciare Bottlerocket istanza. Ricordati di sostituire i seguenti parametri:
 - Sostituisci *subnet* con l'ID della sottorete privata o pubblica in cui verrà avviata l'istanza.
 - Sostituisci *bottlerocket_ami* con l'ID AMI del passaggio precedente.
 - Sostituisci *t3.large* con il tipo di istanza che desideri utilizzare.
 - Sostituisci *region* con il codice regionale.

```
aws ec2 run-instances --key-name ecs-bottlerocket-example \
  --subnet-id subnet \
  --image-id bottlerocket_ami \
  --instance-type t3.large \
  --region region \
  --tag-specifications
  'ResourceType=instance,Tags=[{Key=bottlerocket,Value=example}]' \
  --user-data file:///userdata.toml \
  --iam-instance-profile Name=ecsInstanceRole
```

4. Esegui il comando seguente per verificare che l'istanza di container sia registrata nel cluster. Quando esegui questo comando, ricordati di sostituire i parametri seguenti:
 - Sostituisci *cluster* con il nome del cluster.
 - Sostituisci *region* con il tuo codice regionale.

```
aws ecs list-container-instances --cluster cluster-name --region region
```

Per una guida dettagliata su come iniziare a usare il Bottlerocket sistema operativo su Amazon ECS, consulta [Utilizzo di un'AMI Bottlerocket con Amazon ECS](#) attivo e Guida GitHub introduttiva [Bottlerocket e Amazon ECS](#) sul sito del AWS blog.

Gestione delle istanze di container Amazon ECS Linux

Quando utilizzi EC2 istanze per i tuoi carichi di lavoro Amazon ECS, sei responsabile della manutenzione delle istanze.

Procedure di gestione

- [Avvio di un'istanza di container Linux di Amazon ECS](#)
- [Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati](#)
- [Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot](#)
- [Esecuzione di uno script all'avvio di un'istanza di container Amazon ECS Linux](#)
- [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#)
- [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#)
- [Gestione remota delle istanze di container Amazon ECS tramite AWS Systems Manager](#)
- [Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux](#)
- [Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling](#)
- [Aggiornamento dell'agente del container Amazon ECS](#)

Ogni versione dell'agente del container di Amazon ECS supporta una serie di funzioni differenti e assicura le correzioni dei bug delle versioni precedenti. Quando possibile, consigliamo sempre di utilizzare la versione più recente dell'agente del container di Amazon ECS. Per passare all'ultima versione dell'agente del container, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

[Per vedere quali funzionalità e miglioramenti sono inclusi in ogni versione dell'agente, consulta
https://github.com/aws/amazon-ecs-agent/releases.](#)

Important

La versione Docker minima per parametri affidabili è v20.10.13 e successive, inclusa nell'AMI 20220607 ottimizzata per Amazon ECS e versioni successive.

Le versioni dell'agente Amazon ECS 1.20.0 e successive non supportano più le versioni di Docker precedenti alla 18.01.0.

Avvio di un'istanza di container Linux di Amazon ECS

Puoi creare istanze di container Amazon ECS utilizzando la console Amazon EC2 .

Puoi avviare un'istanza con vari metodi, tra cui la EC2 console Amazon e AWS CLI l'SDK. Per informazioni sugli altri metodi per avviare un'istanza, consulta [Launch your instance](#) nella Amazon EC2 User Guide.

Per ulteriori informazioni sulla procedura guidata di avvio, consulta [Launch an instance using the new launch instance wizard](#) nella Amazon EC2 User Guide.

Prima di iniziare, completa i passaggi descritti in [Configurazione per l'uso di Amazon ECS](#).

Puoi utilizzare la nuova EC2 procedura guidata di Amazon per avviare un'istanza. La procedura guidata di avvio specifica tutti i parametri di avvio necessari per l'avvio di un'istanza.

Parametri per la configurazione di un'istanza

- [Procedura](#)
- [Nome e tag](#)
- [Immagini di applicazioni e sistema operativo \(Amazon Machine Image\)](#)
- [Tipo di istanza](#)
- [Coppia di chiavi \(login\)](#)
- [Impostazioni di rete](#)
- [Per configurare l'archiviazione](#)
- [Dettagli avanzati](#)

Procedura

Prima di iniziare, completa i passaggi descritti in [Configurazione per l'uso di Amazon ECS](#).

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nella barra di navigazione nella parte superiore dello schermo, viene visualizzata la AWS regione corrente (ad esempio, Stati Uniti orientali (Ohio)). Selezionare una regione in cui avviare l'istanza.
3. Dalla dashboard della EC2 console Amazon, scegli Launch instance.

Nome e tag

Il nome dell'istanza è un tag, dove la chiave è Name (Nome) e il valore è il nome specificato. È possibile aggiungere tag all'istanza, ai volumi e alla grafica elastica. Per le istanze spot, è possibile aggiungere un tag solo alla richiesta di istanza spot.

La specifica di un nome di istanza e dei tag aggiuntivi è facoltativa.

- Per Name (Nome), inserire un nome descrittivo per l'istanza. Se non si specifica un nome, l'istanza può essere identificata dal relativo ID, che viene generato automaticamente all'avvio dell'istanza.
- Per aggiungere altri tag, scegliere Add additional tags (Aggiungi altri tag). Scegliere Add tag (Aggiungi tag), quindi immettere una chiave e un valore e selezionare il tipo di risorsa da taggare. Scegliere Add tag (Aggiungi tag) per ogni tag aggiuntivo.

Immagini di applicazioni e sistema operativo (Amazon Machine Image)

Un'Amazon Machine Image (AMI) contiene tutte le informazioni necessarie per creare un'istanza. Ad esempio, un'AMI può contenere il software necessario per fungere da server Web, ad esempio Apache e il sito Web.

Utilizza la barra di ricerca per trovare un'AMI ottimizzata per Amazon ECS adatta pubblicata da AWS

1. Inserisci uno dei seguenti termini nella barra Search (cerca).

- **ami-ecs**
- Il Valore di un'AMI ottimizzata per Amazon ECS.

Per le ultime versioni ottimizzate per Amazon ECS AMIs e i relativi valori, consulta Linux AMI ottimizzata per [Amazon ECS](#).

2. Premere Invio.

3. Nella pagina Choose an Amazon Machine Image (AMI), seleziona la AMIs scheda AWS Marketplace.

4. Dal riquadro Refine results (perfeziona i risultati), seleziona Amazon Web Services come Publisher.

5. Scegli Select (seleziona) nella riga dell'AMI da utilizzare.

In alternativa, scegli Cancel (Annulla) (in alto a destra) per tornare alla procedura guidata di avvio istanza senza scegliere un'AMI. Verrà selezionata un'AMI predefinita. Assicurati che l'AMI soddisfi i requisiti delineati in Linux ottimizzato per [Amazon ECS](#). AMIs

Tipo di istanza

Il tipo di istanza definisce la configurazione hardware e le dimensioni dell'istanza. I tipi di istanza più grandi dispongono di una maggiore quantità di CPU e memoria. Per ulteriori informazioni consulta la sezione relativa ai [tipi di istanza](#).

- Per Tipo di istanza, selezionare il tipo di istanza per l'istanza.

Il tipo di istanza che selezioni determina le risorse disponibili per l'esecuzione delle attività.

Coppia di chiavi (login)

In Key pair name (Nome della coppia di chiavi), scegliere una coppia di chiavi esistente oppure scegliere Create new key pair (Crea nuova coppia di chiavi) per crearne una nuova.

Important

Se si sceglie l'opzione Proceed without key pair (Not recommended) (Procedi senza una coppia di chiavi [non consigliato]), non sarà possibile connetterti all'istanza a meno che non si scelga un'AMI configurata per offrire agli utenti un metodo di accesso alternativo.

Impostazioni di rete

Configurare le impostazioni di rete, se necessario.

- Networking platform Piattaforma di rete): scegli Virtual Private Cloud (VPC), quindi specifica la sottorete nella sezione Network interfaces (Interfacce di rete).
- VPC: selezionare un VPC esistente in cui creare il gruppo di sicurezza.
- Sottorete: è possibile avviare un'istanza in una sottorete associata a una zona di disponibilità, una Local Zone, una zona Wavelength o un Outpost.

Per avviare l'istanza in una zona di disponibilità, selezionare la sottorete in cui avviare l'istanza. Per creare una nuova sottorete, scegliere Create new subnet (Crea nuova sottorete) per passare alla console Amazon VPC. Al termine, tornare alla procedura guidata di avvio istanza e scegliere Refresh (Aggiorna) per caricare la sottorete nell'elenco.

Per avviare l'istanza in una Local Zone, selezionare una sottorete creata nella Local Zone.

Per avviare un'istanza in un Outpost, selezionare una sottorete in un VPC associato a un Outpost.

- **Auto-assign Public IP** (IP pubblico di assegnazione automatica): se l'istanza deve essere accessibile da Internet, verifica che il campo Auto-assign Public IP (Assegna automaticamente IP pubblico) sia impostato su Enable (Abilita). In caso contrario, imposta il campo su Disable (Disabilita).

Note

Le istanze di container richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite istanze di container con indirizzi IP pubblici.

Per ulteriori informazioni sugli endpoint VPC di interfaccia, vedi [Endpoint VPC dell'interfaccia di Amazon ECS \(AWS PrivateLink\)](#)

Se non disponi di un endpoint VPC di interfaccia configurato e le istanze di container non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta [NAT gateways](#) (Gateway NAT) nella Guida per l'utente di Amazon VPC e [Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux](#) in questa guida.

- **Firewall (security groups)** (Firewall [gruppi di sicurezza]): utilizza un gruppo di sicurezza per definire le regole del firewall per l'istanza di container. Tali regole specificano quale traffico di rete in entrata viene distribuito sull'istanza di container. Tutto il traffico rimanente verrà ignorato.
- Per selezionare un gruppo di sicurezza esistente, scegli **Select an existing security group** (Seleziona un gruppo di sicurezza esistente), quindi seleziona il gruppo di sicurezza creato in [Configurazione per l'uso di Amazon ECS](#).

Per configurare l'archiviazione

L'AMI selezionata include uno o più volumi di archiviazione, compreso il volume dispositivo root. È possibile specificare altri volumi da collegare all'istanza.

Puoi utilizzare la vista Semplice.

- **Storage Type** (Tipo di storage): configura lo storage per l'istanza di container.

Se stai utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS, l'istanza dispone di un singolo volume di 30 GiB configurato che è condiviso tra il sistema operativo e Docker.

Se stai utilizzando l'AMI ottimizzata per Amazon ECS, l'istanza dispone di due volumi configurati. Il volume Root è per il sistema operativo, mentre il secondo volume di Amazon EBS (collegato a `/dev/xvdcz`) è per l'utilizzo di Docker.

Puoi aumentare o diminuire le dimensioni del volume per l'istanza in modo che soddisfi le esigenze applicative.

Dettagli avanzati

Per Advanced Details (Dettagli avanzati), espandi la sezione per visualizzare i campi e specifica eventuali parametri aggiuntivi per l'istanza.

- Purchasing option (Opzioni di acquisto): seleziona Request Spot Instances (Richiedi istanze Spot) per avviare un'istanza Spot. Dovrai anche impostare altri campi correlati alle istanze Spot. Per ulteriori informazioni, consulta [Richiesta Istanza Spot](#).

Note

Se utilizzi istanze Spot e visualizzi un messaggio `Not available`, potresti dover scegliere un altro tipo di istanza.

- Profilo istanza IAM: seleziona il ruolo IAM dell'istanza di container. Questo di solito è chiamato `ecsInstanceRole`.

Important

Se non avvii l'istanza di container con le autorizzazioni IAM corrette, l'agente Amazon ECS non potrà connettersi al cluster. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

- Dati utente: configura l'istanza del contenitore Amazon ECS con i dati utente, come le variabili di ambiente dell'agente di [Configurazione dell'agente del container Amazon ECS](#). Gli script dei dati EC2 utente di Amazon vengono eseguiti solo una volta, al primo avvio dell'istanza. Di seguito sono elencati esempi comuni dei dati utente utilizzati per:

- Di default, l'istanza di container si avvia nel cluster predefinito. Per avviarla in un cluster non predefinito, scegli l'elenco Advanced Details (Dettagli avanzati). Quindi, incolla lo script seguente nel campo Dati utente, sostituendolo *your_cluster_name* con il nome del cluster.

```
#!/bin/bash
echo ECS_CLUSTER=your_cluster_name >> /etc/ecs/ecs.config
```

- Se disponi di un file `ecs.config` in Amazon S3 e hai abilitato l'accesso in sola lettura di Amazon S3 al ruolo dell'istanza di container, scegli l'elenco Dettagli avanzati. Quindi, incolla lo script seguente nel campo Dati utente, sostituendolo *your_bucket_name* con il nome del bucket per installarlo AWS CLI e scrivi il file di configurazione al momento del lancio.

Note

Per ulteriori informazioni su questa configurazione, consulta [Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3](#).

```
#!/bin/bash
yum install -y aws-cli
aws s3 cp s3://your_bucket_name/ecs.config /etc/ecs/ecs.config
```

- Specificare i tag per l'istanza di container utilizzando il parametro di configurazione `ECS_CONTAINER_INSTANCE_TAGS`. In questo modo vengono creati tag associati solo ad Amazon ECS, che non possono essere elencati utilizzando l'EC2 API Amazon.

Important

Se avvii le istanze di container utilizzando un gruppo Amazon EC2 Auto Scaling, dovresti usare il parametro di configurazione dell'agente `ECS_CONTAINER_INSTANCE_TAGS` per aggiungere tag. Ciò è dovuto al modo in cui i tag vengono aggiunti alle EC2 istanze Amazon avviate utilizzando i gruppi di Auto Scaling.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=your_cluster_name
ECS_CONTAINER_INSTANCE_TAGS={"tag_key": "tag_value"}
```

EOF

- Specificate i tag per l'istanza del contenitore, quindi utilizzate il parametro di ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM configurazione per propagarli da Amazon EC2 ad Amazon ECS.

Di seguito è riportato un esempio di script di dati utente che propaga i tag associati a un'istanza di container, nonché registra l'istanza di container con un cluster denominato `your_cluster_name`:

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=your_cluster_name
ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM=ec2_instance
EOF
```

Per ulteriori informazioni, consulta [Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati](#).

Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati

Quando avvii un' EC2 istanza Amazon, puoi passare i dati utente all' EC2 istanza. I dati possono essere utilizzati per eseguire attività di configurazione automatizzate di routine e anche per l'esecuzione di script all'avvio dell'istanza. Per Amazon ECS, i casi di utilizzo più comuni per i dati utente riguardano la trasmissione di informazioni sulla configurazione al daemon Docker e all'agente del container Amazon ECS.

Puoi trasferire diversi tipi di dati utente ad Amazon EC2, inclusi boothook cloud, script di shell e direttive. `cloud-init` Per ulteriori informazioni su questi e altri tipi di formato, consulta la [documentazione su cloud-init](#).

Per trasmettere i dati utente durante l'utilizzo della procedura guidata di EC2 avvio di Amazon, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Puoi configurare l'istanza del contenitore per passare i dati nella configurazione dell'agente del contenitore o nella configurazione del demone Docker.

Agente del container Amazon ECS

Le varianti Linux dell'AMI ottimizzata per Amazon ECS cercano i dati di configurazione dell'agente nel file `/etc/ecs/ecs.config` all'avvio dell'agente del container. Puoi specificare questi dati di configurazione al momento del lancio con i dati EC2 utente di Amazon. Per ulteriori informazioni sulle variabili di configurazione per l'agente del container di Amazon ECS disponibili, consulta [Configurazione dell'agente del container Amazon ECS](#).

Per impostare una sola variabile di configurazione dell'agente, ad esempio il nome del cluster, utilizza `echo` per copiare la variabile nel file di configurazione:

```
#!/bin/bash
echo "ECS_CLUSTER=MyCluster" >> /etc/ecs/ecs.config
```

Per scrivere più variabili nel file `/etc/ecs/ecs.config`, utilizza il formato heredoc illustrato di seguito. Questo formato scrive tutti gli elementi nel file di configurazione, inserendoli tra le righe `cat` ed EOF.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
  {"username":"my_name", "password":"my_password", "email":"email@example.com"}}
ECS_LOGLEVEL=debug
ECS_WARM_POOLS_CHECK=true
EOF
```

Per impostare gli attributi di istanza personalizzati, imposta la variabile di ambiente `ECS_INSTANCE_ATTRIBUTES`.

```
#!/bin/bash
cat <<'EOF' >> ecs.config
ECS_INSTANCE_ATTRIBUTES={"envtype":"prod"}
EOF
```

Daemon Docker

Puoi specificare le informazioni di configurazione del demone Docker con i dati utente di Amazon EC2. Per ulteriori informazioni sulle opzioni di configurazione, consulta la [documentazione del daemon Docker](#).

Nell'esempio riportato di seguito, le opzioni personalizzate vengono aggiunte al file di configurazione del daemon Docker, `/etc/docker/daemon.json`, che viene quindi specificato nei dati utente all'avvio dell'istanza.

```
#!/bin/bash
cat <<EOF >/etc/docker/daemon.json
{"debug": true}
EOF
systemctl restart docker --no-block
```

Nell'esempio riportato di seguito, le opzioni personalizzate vengono aggiunte al file di configurazione del daemon Docker, `/etc/docker/daemon.json`, che viene quindi specificato nei dati utente all'avvio dell'istanza. Questo esempio mostra come disattivare il `docker-proxy` nel file di configurazione del daemon Docker.

```
#!/bin/bash
cat <<EOF >/etc/docker/daemon.json
{"userland-proxy": false}
EOF
systemctl restart docker --no-block
```

Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot

Amazon EC2 chiude, interrompe o iberna l'istanza Spot quando il prezzo Spot supera il prezzo massimo per la richiesta o la capacità non è più disponibile. Amazon EC2 fornisce un avviso di interruzione di due minuti a un'istanza Spot per interrompere e interrompere le azioni. Non fornisce l'avviso di due minuti per l'operazione di ibernazione. Se il drenaggio delle istanze Spot di Amazon ECS è attivato sull'istanza, Amazon ECS riceve l'avviso di interruzione dell'istanza Spot e imposta lo stato dell'istanza. DRAINING

Important

Amazon ECS non riceve alcun avviso da Amazon EC2 quando le istanze vengono rimosse da Auto Scaling Capacity Rebalancing. Per ulteriori informazioni, consulta [Amazon EC2 Auto Scaling Capacity Rebalancing](#).

Quando un'istanza di container è impostata su DRAINING, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container. Le attività di

servizio nell'istanza di container di esaurimento che sono in stato PENDING vengono interrotte immediatamente. Se nel cluster sono disponibili istanze di container, le attività del servizio di sostituzione vengono avviate su di esse.

Il drenaggio delle istanze Spot è disattivato per impostazione predefinita.

Puoi attivare il drenaggio delle istanze Spot all'avvio di un'istanza. Aggiungi il seguente script nel campo Dati utente. Sostituisci *MyCluster* con il nome del cluster in cui registrare l'istanza del contenitore.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_ENABLE_SPOT_INSTANCE_DRAINING=true
EOF
```

Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Per attivare lo svuotamento dell'istanza Spot per un'istanza di container esistente

1. Connettiti all'istanza Spot su SSH.
2. Modifica il file `/etc/ecs/ecs.config` e aggiungi quanto segue:

```
ECS_ENABLE_SPOT_INSTANCE_DRAINING=true
```

3. Riavvia il servizio `ecs`.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart ecs
```

4. (Facoltativo) Puoi verificare se l'agente è in esecuzione e visualizzare alcune informazioni sulla nuova istanza di container interrogando l'operazione API di introspezione dell'agente. Per ulteriori informazioni, consulta [the section called "Introspezione dei contenitori"](#).

```
curl http://localhost:51678/v1/metadata
```

Esecuzione di uno script all'avvio di un'istanza di container Amazon ECS Linux

Potrebbe essere necessario eseguire un contenitore specifico su ogni istanza del contenitore per gestire problemi operativi o di sicurezza come il monitoraggio, la sicurezza, le metriche, l'individuazione dei servizi o la registrazione.

Per eseguire questa operazione, puoi configurare le istanze di container per chiamare il comando `docker run` con lo script di dati utente all'avvio o in un sistema di inizializzazione come Upstart o systemd. Anche se questo metodo funziona, presenta di alcuni svantaggi perché Amazon ECS non conosce i container e non può monitorare CPU, memoria, porte o le altre risorse utilizzate. Per garantire che Amazon ECS tenga correttamente conto di tutte le risorse delle attività, crea una definizione di attività per il container per l'esecuzione nelle istanze di container. Quindi, usa Amazon ECS per inserire l'attività al momento del lancio con i dati EC2 utente di Amazon.

Lo script dei dati EC2 utente di Amazon nella procedura seguente utilizza l'API di introspezione di Amazon ECS per identificare l'istanza del contenitore. Quindi, utilizza il AWS CLI `start-task` comando `and` per eseguire un'attività specifica su se stessa durante l'avvio.

Per avviare un'attività al momento dell'avvio di un'istanza di container

1. Modifica il ruolo IAM `ecsInstanceRole` per aggiungere le autorizzazioni per l'operazione API `StartTask`. Per ulteriori informazioni, consulta [Aggiornare le autorizzazioni per un ruolo](#) nella Guida per l'AWS Identity and Access Management utente.
2. Avvia una o più istanze di container utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Avvia nuove istanze di container e usa il seguente script di esempio nei dati utente. EC2 Sostituisci *`your_cluster_name`* con il cluster l'istanza del contenitore in cui registrarsi e *`my_task_def`* con la definizione dell'attività da eseguire sull'istanza all'avvio.

Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Note

Il contenuto MIME in più parti mostrato di seguito usa uno script di shell per impostare i valori di configurazione e installare pacchetti. Utilizza anche un processo systemd per avviare l'attività una volta che il servizio ecs è in esecuzione e l'API di introspezione è disponibile.

```
Content-Type: multipart/mixed; boundary==="BOUNDARY=="
```

```
MIME-Version: 1.0

--==BOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
# Specify the cluster that the container instance should register into
cluster=your_cluster_name

# Write the cluster configuration variable to the ecs.config file
# (add any other configuration variables here also)
echo ECS_CLUSTER=$cluster >> /etc/ecs/ecs.config

START_TASK_SCRIPT_FILE="/etc/ecs/ecs-start-task.sh"
cat <<- 'EOF' > ${START_TASK_SCRIPT_FILE}
exec 2>>/var/log/ecs/ecs-start-task.log
set -x

# Install prerequisite tools
yum install -y jq aws-cli

# Wait for the ECS service to be responsive
until curl -s http://localhost:51678/v1/metadata
do
    sleep 1
done

# Grab the container instance ARN and AWS Region from instance metadata
instance_arn=$(curl -s http://localhost:51678/v1/metadata | jq -r '.
| .ContainerInstanceArn' | awk -F/ '{print $NF}' )
cluster=$(curl -s http://localhost:51678/v1/metadata | jq -r '. | .Cluster' | awk
-F/ '{print $NF}' )
region=$(curl -s http://localhost:51678/v1/metadata | jq -r '.
| .ContainerInstanceArn' | awk -F: '{print $4}')

# Specify the task definition to run at launch
task_definition=my_task_def

# Run the AWS CLI start-task command to start your task on this container instance
aws ecs start-task --cluster $cluster --task-definition $task_definition --
container-instances $instance_arn --started-by $instance_arn --region $region
EOF

# Write systemd unit file
```

```
UNIT="ecs-start-task.service"
cat <<- EOF > /etc/systemd/system/${UNIT}
[Unit]
Description=ECS Start Task
Requires=ecs.service
After=ecs.service

[Service]
Restart=on-failure
RestartSec=30
ExecStart=/usr/bin/bash ${START_TASK_SCRIPT_FILE}

[Install]
WantedBy=default.target
EOF

# Enable our ecs.service dependent service with `--no-block` to prevent systemd
# deadlock
# See https://github.com/aws/amazon-ecs-agent/issues/1707
systemctl enable --now --no-block "${UNIT}"
---==BOUNDARY===---
```

3. Verifica che le istanze di container vengano avviate nel cluster corretto e che le attività siano state avviate.
 - a. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
 - b. Dalla barra di navigazione, scegli la regione in cui si trova il cluster.
 - c. Nel pannello di navigazione, scegli Clusters (Cluster) e seleziona il cluster che ospita le istanze di container.
 - d. Nella pagina Cluster, seleziona Attività e quindi le tue attività.

Su ogni istanza di container che hai avviato dovrebbe essere in esecuzione la tua attività.

Se non visualizzi le attività, puoi accedere alle istanze di container con SSH e controllare le informazioni di debug nel file `/var/log/ecs/ecs-start-task.log`.

Aumento delle interfacce di rete di istanze di container Amazon ECS Linux

Note

Questa funzione non è disponibile su Fargate.

Ogni attività che utilizza la modalità `awsvpc` di rete riceve la propria interfaccia di rete elastica (ENI), che è collegata all'istanza del contenitore che la ospita. Esiste un limite predefinito al numero di interfacce di rete che possono essere collegate a un' EC2 istanza Amazon e l'interfaccia di rete principale conta come una. Ad esempio, per impostazione predefinita, a un'`c5.large` istanza possono essere ENIs collegate fino a tre istanze. L'interfaccia di rete principale per l'istanza conta come una sola, quindi è possibile collegarne altre due ENIs all'istanza. Perché ogni operazione che utilizza la modalità `awsvpc` di rete richiede un ENI, in genere è possibile eseguire solo due di queste attività su questo tipo di istanza.

Amazon ECS supporta il lancio di istanze di container con maggiori ENI densità utilizzando i tipi di EC2 istanze Amazon supportati. Quando utilizzi questi tipi di istanze e attivi l'impostazione dell'`awsvpcTrunking` account, ne ENIs sono disponibili altre sulle istanze di container appena lanciate. Questa configurazione consente di posizionare più attività su ciascuna istanza di container. Per utilizzare la console per attivare la funzionalità, consulta [Modifica delle impostazioni dell'account Amazon ECS](#). Per utilizzare l'opzione AWS CLI per attivare la funzionalità, vedere [Gestione delle impostazioni dell'account Amazon ECS tramite AWS CLI](#).

Ad esempio, un'`c5.large` istanza con `awsvpcTrunking` ha un valore aumentato ENI limite di dodici. L'istanza di container avrà un'interfaccia di rete primaria e Amazon ECS crea e collega un'interfaccia di rete "trunk" all'istanza di container. Pertanto, questa configurazione consente di avviare dieci attività sull'istanza di container anziché le due attività correnti.

L'interfaccia di rete trunk è completamente gestita da Amazon ECS e viene eliminata quando termini o annulli la registrazione dell'istanza di container dal cluster. Per ulteriori informazioni, consulta [Opzioni di task networking di Amazon ECS per il tipo di EC2 avvio](#).

Considerazioni

Considerate quanto segue quando utilizzate il ENI funzione di trunking.

- Solo le varianti Linux dell'AMI ottimizzata per Amazon ECS o altre varianti di Amazon Linux con versione 1.28.1 o successiva dell'agente contenitore e versione 1.28.1-2 o successiva del

pacchetto `ecs-init` supportano l'aumento ENI limiti. Se utilizzi la variante Linux più recente dell'AMI ottimizzata per Amazon ECS, questi requisiti saranno soddisfatti. I container Windows non sono al momento supportati.

- Solo le nuove EC2 istanze Amazon lanciate dopo l'attivazione `awsvpcTrunking` ricevono l'aumento ENI limiti e interfaccia di rete trunk. In precedenza, le istanze avviate non ricevevano queste caratteristiche, a prescindere dalle operazioni eseguite.
- EC2 Le istanze Amazon devono avere le richieste IPv4 DNS basate sulle risorse disattivate. Per disabilitare questa opzione, deseleziona l'opzione `Abilita richieste DNS basate su risorse IPV4 (A record)` quando crei una nuova istanza nella console Amazon. EC2 Per disabilitare questa opzione usando il AWS CLI, usa il seguente comando.

```
aws ec2 modify-private-dns-name-options --instance-id i-xxxxxxx --no-enable-resource-name-dns-a-record --no-dry-run
```

- EC2 Le istanze Amazon in sottoreti condivise non sono supportate. Se vengono utilizzate, non riescono a eseguire la registrazione a un cluster.
- Le tue attività devono utilizzare la modalità di `awsvpc` rete e il EC2 tipo di avvio. Le attività che utilizzavano il tipo di lancio Fargate ricevevano sempre un codice dedicato ENI indipendentemente da quanti ne vengono lanciati, quindi questa funzionalità non è necessaria.
- Le tue attività devono essere avviate nello stesso Amazon VPC dell'istanza del contenitore. Le tue attività non verranno avviate con un errore di attributo se non sono all'interno dello stesso VPC.
- Quando si avvia una nuova istanza di container, l'istanza effettua la transizione a uno stato `REGISTERING` mentre viene eseguito il provisioning dell'interfaccia di rete elastica trunk per l'istanza. Se la registrazione ha esito negativo, l'istanza esegue la transizione a uno stato `REGISTRATION_FAILED`. È possibile risolvere i problemi relativi a una registrazione non riuscita descrivendo l'istanza di container per visualizzare il campo `statusReason` che descrive il motivo dell'errore. L'istanza di container può quindi essere annullata manualmente o terminata. Una volta che l'istanza del contenitore è stata annullata o terminata con successo, Amazon ECS eliminerà il trunk ENI.

Note

Amazon ECS emette eventi di modifica dello stato dell'istanza di container che è possibile monitorare per le istanze che passano a uno stato `REGISTRATION_FAILED`. Per ulteriori informazioni, consulta [Eventi di modifica dello stato dell'istanza del contenitore Amazon ECS](#).

- Al termine dell'istanza di container, l'istanza esegue la transizione a uno stato DEREGISTERING mentre viene annullato il provisioning dell'interfaccia di rete elastica trunk. L'istanza quindi esegue la transizione a uno stato INACTIVE.
- Se un'istanza di contenitore si trova in una sottorete pubblica con l'aumento ENI limits viene interrotta e quindi riavviata, l'istanza perde il suo indirizzo IP pubblico e l'agente del contenitore perde la connessione.
- Quando abiliti `awsVpcTrunking`, le istanze del contenitore ne ricevono un altro ENI che utilizza il gruppo di sicurezza predefinito del VPC ed è gestito da Amazon ECS.

Un VPC di default include una sottorete pubblica in ogni zona di disponibilità, un gateway Internet e impostazioni per abilitare la risoluzione DNS. La sottorete è una sottorete pubblica, poiché la tabella di routing principale invia il traffico della sottorete destinato a Internet al gateway Internet. Puoi rendere una sottorete predefinita privata rimuovendo la route dalla destinazione 0.0.0.0/0 all'Internet Gateway. Tuttavia, se si esegue questa operazione, nessuna istanza di contenitore in esecuzione in quella sottorete può accedere a Internet. È possibile aggiungere o eliminare le regole dei gruppi di sicurezza per controllare il traffico in entrata e in uscita dalle sottoreti. Per ulteriori informazioni, consulta [le regole dei gruppi di sicurezza](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Prerequisiti

Prima di avviare un'istanza di container con l'aumento ENI limiti, è necessario completare i seguenti prerequisiti.

- Il ruolo collegato ai servizi per Amazon ECS deve essere creato. Il ruolo collegato al servizio Amazon ECS fornisce ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#). Puoi anche creare il ruolo collegato al servizio con il seguente comando. AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- Il ruolo IAM dell'account o dell'istanza di container deve abilitare le impostazioni dell'account `awsVpcTrunking`. Ti consigliamo di creare 2 ruoli di istanza del contenitore (`ecsInstanceRole`). È quindi possibile abilitare l'impostazione dell'`awsVpcTrunking` account per

un ruolo e utilizzare quel ruolo per attività che richiedono il trunking ENI. Per informazioni sul ruolo dell'istanza del contenitore, vedere. [Ruolo IAM delle istanze di container Amazon ECS](#)

Una volta soddisfatti i prerequisiti, puoi avviare una nuova istanza di container utilizzando uno dei tipi di EC2 istanza Amazon supportati e l'istanza avrà l'aumento ENI limiti. Per una lista di tipi di istanze supportate, consulta [Istanze supportate per interfacce di rete di container Amazon ECS potenziate](#). La versione dell'istanza di container per l'agente del container deve essere 1.28.1 o successiva e la versione del pacchetto ecs-init deve essere 1.28.1-2 o successiva. Se utilizzi la variante Linux più recente dell'AMI ottimizzata per Amazon ECS, questi requisiti saranno soddisfatti. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Important

EC2 Le istanze Amazon devono avere le richieste IPv4 DNS basate sulle risorse disattivate. Per disabilitare questa opzione, assicurati che l'opzione Abilita richieste DNS basate su risorse IPV4 (A record) sia deselezionata quando crei una nuova istanza utilizzando la console Amazon. EC2 Per disabilitare questa opzione utilizzando il AWS CLI, usa il seguente comando.

```
aws ec2 modify-private-dns-name-options --instance-id i-xxxxxxx --no-enable-resource-name-dns-a-record --no-dry-run
```

Per visualizzare le istanze del contenitore con aumento ENI limiti con AWS CLI

Ogni istanza di container dispone di un'interfaccia di rete predefinita, nota anche come interfaccia di rete trunk. Usa il comando seguente per elencare le istanze del contenitore con aumento ENI limita interrogando l'ecs.aws-vpc-trunk-id attributo, il che indica che ha un'interfaccia di rete trunk.

- [list-attributes](#) (AWS CLI)

```
aws ecs list-attributes \  
    --target-type container-instance \  
    --attribute-name ecs.aws-vpc-trunk-id \  
    --cluster cluster_name \  
    --region us-east-1
```

- [Get-ECSAttribute List](#) ()AWS Tools for Windows PowerShell

```
Get-ECSAttributeList -TargetType container-instance -AttributeName ecs.awsipc-trunk-id -Region us-east-1
```

Istanze supportate per interfacce di rete di container Amazon ECS potenziate

Di seguito vengono mostrati i tipi di EC2 istanze Amazon supportati e quante attività che utilizzano la modalità di awsipc rete possono essere avviate su ciascun tipo di istanza prima e dopo l'attivazione dell'impostazione dell'awsipcTrunkingaccount.

Important

Sebbene altri tipi di istanza siano supportati nella stessa famiglia di istanze, i tipi di istanza a1.metal, c5.metal, c5a.8xlarge, c5ad.8xlarge, c5d.metal, m5.metal, p3dn.24xlarge, r5.metal, r5.8xlarge e r5d.metal non sono supportati. Le famiglie di istanze c5n, d3, d3en, g3, g3s, g4dn, i3, i3en, inf1, m5dn, m5n, m5zn, mac1, r5b, r5n, r5dn, u-12tb1, u-6tb1, u-9tb1 e z1d non sono supportate.

Argomenti

- [Uso generale](#)
- [Calcolo ottimizzato](#)
- [Memoria ottimizzata](#)
- [Archiviazione ottimizzata](#)
- [Elaborazione accelerata](#)
- [High Performance Computing](#)

Uso generale

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
a1.medium	1	10
a1.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
a1.xlarge	3	20
a1.2xlarge	3	40
a1.4xlarge	7	60
m5.large	2	10
m5.xlarge	3	20
m5.2xlarge	3	40
m5.4xlarge	7	60
m5.8xlarge	7	60
m5.12xlarge	7	60
m5.16xlarge	14	120
m5.24xlarge	14	120
m5a.large	2	10
m5a.xlarge	3	20
m5a.2xlarge	3	40
m5a.4xlarge	7	60
m5a.8xlarge	7	60
m5a.12xlarge	7	60
m5a.16xlarge	14	120
m5a.24xlarge	14	120
m5ad.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m5ad.xlarge	3	20
m5ad.2xlarge	3	40
m5ad.4xlarge	7	60
m5ad.8xlarge	7	60
m5ad.12xlarge	7	60
m5ad.16xlarge	14	120
m5ad.24xlarge	14	120
m5d.large	2	10
m5d.xlarge	3	20
m5d.2xlarge	3	40
m5d.4xlarge	7	60
m5d.8xlarge	7	60
m5d.12xlarge	7	60
m5d.16xlarge	14	120
m5d.24xlarge	14	120
m5d.metal	14	120
m6a.large	2	10
m6a.xlarge	3	20
m6a.2xlarge	3	40
m6a.4xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m6a.8xlarge	7	90
m6a.12xlarge	7	120
m6a.16xlarge	14	120
m6a.24xlarge	14	120
m6a.32xlarge	14	120
m6a.48xlarge	14	120
m6a.metal	14	120
m6g.medium	1	4
m6g.large	2	10
m6g.xlarge	3	20
m6g.2xlarge	3	40
m6g.4xlarge	7	60
m6g.8xlarge	7	60
m6g.12xlarge	7	60
m6g.16xlarge	14	120
m6g.metal	14	120
m6gd.medium	1	4
m6gd.large	2	10
m6gd.xlarge	3	20
m6gd.2xlarge	3	40

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m6gd.4xlarge	7	60
m6gd.8xlarge	7	60
m6gd.12xlarge	7	60
m6gd.16xlarge	14	120
m6gd.metal	14	120
m6i.large	2	10
m6i.xlarge	3	20
m6i.2xlarge	3	40
m6i.4xlarge	7	60
m6i.8xlarge	7	90
m6i.12xlarge	7	120
m6i.16xlarge	14	120
m6i.24xlarge	14	120
m6i.32xlarge	14	120
m6i.metal	14	120
m6id.large	2	10
m6id.xlarge	3	20
m6id.2xlarge	3	40
m6id.4xlarge	7	60
m6id.8xlarge	7	90

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m6id.12xlarge	7	120
m6id.16xlarge	14	120
m6id.24xlarge	14	120
m6id.32xlarge	14	120
m6id.metal	14	120
m6idn.large	2	10
m6idn.xlarge	3	20
m6idn.2xlarge	3	40
m6idn.4xlarge	7	60
m6idn.8xlarge	7	90
m6idn.12xlarge	7	120
m6idn.16xlarge	14	120
m6idn.24xlarge	14	120
m6idn.32xlarge	15	120
m6idn.metal	15	120
m6in.large	2	10
m6in.xlarge	3	20
m6in.2xlarge	3	40
m6in.4xlarge	7	60
m6in.8xlarge	7	90

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m6in.12xlarge	7	120
m6in.16xlarge	14	120
m6in.24xlarge	14	120
m6in.32xlarge	15	120
m6in.metal	15	120
m7a.medium	1	4
m7a.large	2	10
m7a.xlarge	3	20
m7a.2xlarge	3	40
m7a.4xlarge	7	60
m7a.8xlarge	7	90
m7a.12xlarge	7	120
m7a.16xlarge	14	120
m7a.24xlarge	14	120
m7a.32xlarge	14	120
m7a.48xlarge	14	120
m7a.metal-48xl	14	120
m7g.medium	1	4
m7g.large	2	10
m7g.xlarge	3	20

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m7g.2xlarge	3	40
m7g.4xlarge	7	60
m7g.8xlarge	7	60
m7g.12xlarge	7	60
m7g.16xlarge	14	120
m7g.metal	14	120
m7gd.medium	1	4
m7gd.large	2	10
m7gd.xlarge	3	20
m7gd.2xlarge	3	40
m7gd.4xlarge	7	60
m7gd.8xlarge	7	60
m7gd.12xlarge	7	60
m7gd.16xlarge	14	120
m7gd. metallo	14	120
m7i.large	2	10
m7i.xlarge	3	20
m7i.2xlarge	3	40
m7i.4xlarge	7	60
m7i.8xlarge	7	90

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m7i.12xlarge	7	120
m7i.16xlarge	14	120
m7i.24xlarge	14	120
m7a.48xlarge	14	120
m7i.metal-24xl	14	120
m7i.metal-48xl	14	120
m7i-flex.large	2	4
m7i-flex.xlarge	3	10
m7i-flex.2xlarge	3	20
m7i-flex.4xlarge	7	40
m7i-flex.8xlarge	7	60
m7i-flex. 12 x grande	7	120
m7i-flex.16 x grande	14	120
m8 g. Medio	1	4
m8 g. Grande	2	10
m8g.xlarge	3	20
m8 g. 2 x grande	3	40
m8 g. 4 x grande	7	60
m8 g. 8x grande	7	60
m8 g. 12 x grande	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m 8 g. 16 x grande	14	120
m 8 g. 24 x grande	14	120
m 8 g. 48 x grande	14	120
m 8 g. Metallo - 24 XL	14	120
m8g.metallo-48xl	14	120
m8 gd. medio	1	4
m8gd.grande	2	10
m8gd.xlarge	3	20
m8gd.2xlarge	3	40
m8gd.4xlarge	7	60
m8gd.8xlarge	7	60
m8gd.12xlarge	7	60
m8gd.16xlarge	14	120
m8gd.24xlarge	14	120
m8gd.48xlarge	14	120
m8gd.metal-24xl	14	120
m8gd.metal-48xl	14	120
mac2.metal	7	12
mac2-m1 ultra.metal	7	12
mac2-m2.metal	7	12

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione

mac2-m2pro.metal

7

12

Calcolo ottimizzato

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione

c5.large

2

10

c5.xlarge

3

20

c5.2xlarge

3

40

c5.4xlarge

7

60

c5.9xlarge

7

60

c5.12xlarge

7

60

c5.18xlarge

14

120

c5.24xlarge

14

120

c5a.large

2

10

c5a.xlarge

3

20

c5a.2xlarge

3

40

c5a.4xlarge

7

60

c5a.12xlarge

7

60

c5a.16xlarge

14

120

c5a.24xlarge

14

120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c5ad.large	2	10
c5ad.xlarge	3	20
c5ad.2xlarge	3	40
c5ad.4xlarge	7	60
c5ad.12xlarge	7	60
c5ad.16xlarge	14	120
c5ad.24xlarge	14	120
c5d.large	2	10
c5d.xlarge	3	20
c5d.2xlarge	3	40
c5d.4xlarge	7	60
c5d.9xlarge	7	60
c5d.12xlarge	7	60
c5d.18xlarge	14	120
c5d.24xlarge	14	120
c6a.large	2	10
c6a.xlarge	3	20
c6a.2xlarge	3	40
c6a.4xlarge	7	60
c6a.8xlarge	7	90

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c6a.12xlarge	7	120
c6a.16xlarge	14	120
c6a.24xlarge	14	120
c6a.32xlarge	14	120
c6a.48xlarge	14	120
c6a.metal	14	120
c6g.medium	1	4
c6g.large	2	10
c6g.xlarge	3	20
c6g.2xlarge	3	40
c6g.4xlarge	7	60
c6g.8xlarge	7	60
c6g.12xlarge	7	60
c6g.16xlarge	14	120
c6g.metal	14	120
c6gd.medium	1	4
c6gd.large	2	10
c6gd.xlarge	3	20
c6gd.2xlarge	3	40
c6gd.4xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c6gd.8xlarge	7	60
c6gd.12xlarge	7	60
c6gd.16xlarge	14	120
c6gd.metal	14	120
c6gn.medium	1	4
c6gn.large	2	10
c6gn.xlarge	3	20
c6gn.2xlarge	3	40
c6gn.4xlarge	7	60
c6gn.8xlarge	7	60
c6gn.12xlarge	7	60
c6gn.16xlarge	14	120
c6i.large	2	10
c6i.xlarge	3	20
c6i.2xlarge	3	40
c6i.4xlarge	7	60
c6i.8xlarge	7	90
c6i.12xlarge	7	120
c6i.16xlarge	14	120
c6i.24xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c6i.32xlarge	14	120
c6i.metal	14	120
c6id.large	2	10
c6id.xlarge	3	20
c6id.2xlarge	3	40
c6id.4xlarge	7	60
c6id.8xlarge	7	90
c6id.12xlarge	7	120
c6id.16xlarge	14	120
c6id.24xlarge	14	120
c6id.32xlarge	14	120
c6id.metal	14	120
c6in.large	2	10
c6in.xlarge	3	20
c6in.2xlarge	3	40
c6in.4xlarge	7	60
c6in.8xlarge	7	90
c6in.12xlarge	7	120
c6in.16xlarge	14	120
c6in.24xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c6in.32xlarge	15	120
c6in.metal	15	120
c7a.medium	1	4
c7a.large	2	10
c7a.xlarge	3	20
c7a.2xlarge	3	40
c7a.4xlarge	7	60
c7a.8xlarge	7	90
c7a.12xlarge	7	120
c7a.16xlarge	14	120
c7a.24xlarge	14	120
c7a.32xlarge	14	120
c7a.48xlarge	14	120
m7a.metal-48xl	14	120
c7g.medium	1	4
c7g.large	2	10
c7g.xlarge	3	20
c7g.2xlarge	3	40
c7g.4xlarge	7	60
c7g.8xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c7g.12xlarge	7	60
c7g.16xlarge	14	120
c7g.metal	14	120
c7gd.medium	1	4
c7gd.large	2	10
c7gd.xlarge	3	20
c7gd.2xlarge	3	40
c7gd.4xlarge	7	60
c7gd.8xlarge	7	60
c7gd.12xlarge	7	60
c7gd.16xlarge	14	120
c7gd. metallo	14	120
c7gn.medium	1	4
c7gn.large	2	10
c7gn.xlarge	3	20
c7gn.2xlarge	3	40
c7gn.4xlarge	7	60
c7gn.8xlarge	7	60
c7gn.12xlarge	7	60
c7gn.16xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c7gn.metallo	14	120
c7i.large	2	10
c7i.xlarge	3	20
c7i.2xlarge	3	40
c7i.4xlarge	7	60
c7i.8xlarge	7	90
c7i.12xlarge	7	120
c7i.16xlarge	14	120
c7i.24xlarge	14	120
c7i.48xlarge	14	120
c7i.metal-24xl	14	120
c7i.metal-48xl	14	120
c7i-flex.large	2	4
c7i-flex.xlarge	3	10
c7i-flex.2xgrande	3	20
c7i-flex.4xgrande	7	40
c7i-flex.8xgrande	7	60
c7i-flex.12xgrande	7	120
c7i-flex.16 x grande	14	120
c8 g. Medio	1	4

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c8 g. Grande	2	10
c8g.xlarge	3	20
c8 g. 2 x grande	3	40
c 8 g. 4 x grande	7	60
c 8 g. 8 x grande	7	60
c 8 g. 12 x grande	7	60
c 8 g. 16 x grande	14	120
c 8 g. 24 x grande	14	120
c 8 g. 48 x grande	14	120
c8 g. Metallo - 24 XL	14	120
c8g.metallo-48xl	14	120
c8gd. medio	1	4
c8gd.grande	2	10
c8gd.xlarge	3	20
c8gd.2xlarge	3	40
c8gd.4xlarge	7	60
c8gd.8xlarge	7	60
c8gd.12xlarge	7	60
c8gd.16 x grande	14	120
c8gd.24xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
c8gd.48 x grande	14	120
c8gd.metal-24xl	14	120
c8gd.metal-48xl	14	120

Memoria ottimizzata

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r5.large	2	10
r5.xlarge	3	20
r5.2xlarge	3	40
r5.4xlarge	7	60
r5.12xlarge	7	60
r5.16xlarge	14	120
r5.24xlarge	14	120
r5a.large	2	10
r5a.xlarge	3	20
r5a.2xlarge	3	40
r5a.4xlarge	7	60
r5a.8xlarge	7	60
r5a.12xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r5a.16xlarge	14	120
r5a.24xlarge	14	120
r5ad.large	2	10
r5ad.xlarge	3	20
r5ad.2xlarge	3	40
r5ad.4xlarge	7	60
r5ad.8xlarge	7	60
r5ad.12xlarge	7	60
r5ad.16xlarge	14	120
r5ad.24xlarge	14	120
r5b.16xlarge	14	120
r5d.large	2	10
r5d.xlarge	3	20
r5d.2xlarge	3	40
r5d.4xlarge	7	60
r5d.8xlarge	7	60
r5d.12xlarge	7	60
r5d.16xlarge	14	120
r5d.24xlarge	14	120
r5dn.16xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r 6a. Grande	2	10
r6a.xlarge	3	20
r6a.2xlarge	3	40
r6a.4xlarge	7	60
r6a.8xlarge	7	90
r6a.12xlarge	7	120
r6a.16xlarge	14	120
r6a.24xlarge	14	120
r6a.32xlarge	14	120
r6a.48xlarge	14	120
r6a.metal	14	120
r6g.medium	1	4
r6g.large	2	10
r6g.xlarge	3	20
r6g.2xlarge	3	40
r6g.4xlarge	7	60
r6g.8xlarge	7	60
r6g.12xlarge	7	60
r6g.16xlarge	14	120
r6g.metal	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r6gd.medium	1	4
r6gd.large	2	10
r6gd.xlarge	3	20
r6gd.2xlarge	3	40
r6gd.4xlarge	7	60
r6gd.8xlarge	7	60
r6gd.12xlarge	7	60
r6gd.16xlarge	14	120
r6gd.metal	14	120
r6i.large	2	10
r6i.xlarge	3	20
r6i.2xlarge	3	40
r6i.4xlarge	7	60
r6i.8xlarge	7	90
r6i.12xlarge	7	120
r6i.16xlarge	14	120
r6i.24xlarge	14	120
r6i.32xlarge	14	120
r6i.metal	14	120
r6idn.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r6idn.xlarge	3	20
r6idn.2xlarge	3	40
r6idn.4xlarge	7	60
r6idn.8xlarge	7	90
r6idn.12xlarge	7	120
r6idn.16xlarge	14	120
r6idn.24xlarge	14	120
r6idn.32xlarge	15	120
r6idn.metal	15	120
r6in.large	2	10
r6in.xlarge	3	20
r6in.2xlarge	3	40
r6in.4xlarge	7	60
r6in.8xlarge	7	90
r6in.12xlarge	7	120
r6in.16xlarge	14	120
r6in.24xlarge	14	120
r6in.32xlarge	15	120
r6in.metal	15	120
r6id.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r6id.xlarge	3	20
r6id.2xlarge	3	40
r6id.4xlarge	7	60
r6id.8xlarge	7	90
r6id.12xlarge	7	120
r6id.16xlarge	14	120
r6id.24xlarge	14	120
r6id.32xlarge	14	120
r6id. Metallo	14	120
r7a. medio	1	4
r7a.large	2	10
r7a.xlarge	3	20
r7a.2xlarge	3	40
r7a.4xlarge	7	60
r7a.8xlarge	7	90
r7a.12xlarge	7	120
r7a.16xlarge	14	120
r7a.24xlarge	14	120
r7a.32xlarge	14	120
r7a.48xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
m7a.metal-48xl	14	120
r7g.medium	1	4
r7g.large	2	10
r7g.xlarge	3	20
r7g.2xlarge	3	40
r7g.4xlarge	7	60
r7g.8xlarge	7	60
r7g.12xlarge	7	60
r7g.16xlarge	14	120
r7g.metal	14	120
r7gd.medium	1	4
r7gd.large	2	10
r7gd.xlarge	3	20
r7gd.2xlarge	3	40
r7gd.4xlarge	7	60
r7gd.8xlarge	7	60
r7gd.12xlarge	7	60
r7gd.16xlarge	14	120
r7gd. metallo	14	120
r7i.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r7i.xlarge	3	20
r7i.2xlarge	3	40
r7i.4xlarge	7	60
r7i.8xlarge	7	90
r7i.12xlarge	7	120
r7i.16xlarge	14	120
r7i.24xlarge	14	120
r7i.48xlarge	14	120
r7i.metal-24xl	14	120
r7i.metal-48xl	14	120
r7iz.large	2	10
r7iz.xlarge	3	20
r7iz.2xlarge	3	40
r7iz.4xlarge	7	60
r7iz.8xlarge	7	90
r7iz.12xlarge	7	120
r7iz.16xlarge	14	120
r7iz.32xlarge	14	120
r7iz.metal-16xl	14	120
r7iz.metal-32xl	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r8g. Medio	1	4
r8 g. Grande	2	10
r8g.xlarge	3	20
r8 g. 2 x grande	3	40
r8 g. 4x grande	7	60
r8 g. 8x grande	7	60
r8 g. 12 x grande	7	60
r8 g. 16 x grande	14	120
r8 g. 24 x grande	14	120
r8 g. 48 x grande	14	120
r8 g., metallo, 24 XL	14	120
r8g.metallo-48xl	14	120
r8 gd. medio	1	4
r8gd.grande	2	10
r8gd.xlarge	3	20
r8gd.2xlarge	3	40
r8gd.4xlarge	7	60
r8gd.8xlarge	7	60
r8gd.12xlarge	7	60
r8gd.16xlarge	14	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
r8gd.24xlarge	14	120
r8gd.48xlarge	14	120
r8gd.metal-24xl	14	120
r8gd.metal-48xl	14	120
u-3tb1.56xlarge	7	12
u-6tb1.56xlarge	14	12
u-18tb1.112xlarge	14	12
u-18tb1.metal	14	12
u-24tb1.112xlarge	14	12
u-24tb1.metal	14	12
u7i-6 tb.112 x grande	14	120
u7i-8 tb.112 x grande	14	120
u7i-12 tb.224 x grande	14	120
u7in-16 tb.224xlarge	15	120
u7in-24 tb.224xlarge	15	120
u7in-32 tb.224xlarge	15	120
u7 pollici - 32 tb.480 x grande	15	120
x2gd.medium	1	10
x2gd.large	2	10

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
x2gd.xlarge	3	20
x2gd.2xlarge	3	40
x2gd.4xlarge	7	60
x2gd.8xlarge	7	60
x2gd.12xlarge	7	60
x2gd.16xlarge	14	120
x2gd.metal	14	120
x2idn.16xlarge	14	120
x2idn.24xlarge	14	120
x2idn.32xlarge	14	120
x2idn.metal	14	120
x2iedn.xlarge	3	13
x2iedn.2xlarge	3	29
x2iedn.4xlarge	7	60
x2iedn.8xlarge	7	120
x2iedn.16xlarge	14	120
x2iedn.24xlarge	14	120
x2iedn.32xlarge	14	120
x2iedn.metal	14	120
x2iezn.2xlarge	3	64

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
x2iezn.4xlarge	7	120
x2iezn.6xlarge	7	120
x2iezn.8xlarge	7	120
x2iezn.12xlarge	14	120
x2iezn.metal	14	120
x8 g. Medio	1	4
x8 g. Grande	2	10
x8 g.xlarge	3	20
x8 g. 2 x grande	3	40
x 8 g. 4 x grande	7	60
x 8 g. 8 x grande	7	60
x 8 g. 12 x grande	7	60
x 8 g. 16 x grande	14	120
x 8 g. 24 x grande	14	120
x 8 g. 48 x grande	14	120
x8 g. Metallo - 24 XL	14	120
x8g.metallo-48xl	14	120

Archiviazione ottimizzata

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
i4g.large	2	10
i4g.xlarge	3	20
i4g.2xlarge	3	40
i4g.4xlarge	7	60
i4i.8xlarge	7	60
i4i.16xlarge	14	120
i4i.xlarge	3	8
i4i.2xlarge	3	28
i4i.4xlarge	7	58
i4i.8xlarge	7	118
i4i.12xlarge	7	118
i4i.16xlarge	14	248
i4i.24xlarge	14	118
i4i.32xlarge	14	498
i4i.metal	14	498
i7 i.large	2	10
i7i.xlarge	3	20
i7i.2xlarge	3	40
i7i.4xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
i7i.8xlarge	7	90
i7i.12xlarge	7	90
i7i.16xlarge	14	120
i7i.24xlarge	14	120
i7i.48 xlarge	14	120
i7i.metal-24xl	14	120
i7i.metal-48xl	14	120
i7ie.large	2	20
i7ie.xlarge	3	29
i7ie.2xlarge	3	29
i7ie.3xlarge	3	29
i7ie.6xlarge	7	60
i7ie.12xlarge	7	60
i7ie.18xlarge	14	120
i7ie.24xlarge	14	120
i7ie.48xlarge	14	120
i7ie.metal-24xl	14	120
i7ie.metal-48xl	14	120
i8 g. Grande	2	10
i8g.xlarge	3	20

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
i8g. 2 x grande	3	40
i8 g. 4 x grande	7	60
i8 g. 8x grande	7	60
i8 g. 12 x grande	7	60
i8 g. 16 x grande	14	120
i8 g. 24 x grande	14	120
i8 g. 48 x grande	14	120
i8 g., in metallo, 24 XL	14	120
im4gn.large	2	10
im4gn.xlarge	3	20
im4gn.2xlarge	3	40
im4gn.4xlarge	7	60
im4gn.8xlarge	7	60
im4gn.16xlarge	14	120
is4gen.medium	1	4
is4gen.large	2	10
is4gen.xlarge	3	20
is4gen.2xlarge	3	40
is4gen.4xlarge	7	60
is4gen.8xlarge	7	60

Elaborazione accelerata

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
dl1.24xlarge	59	120
dl2q.24xlarge	14	120
f 2,6 x grande	7	90
f 2,12 x grande	7	120
f2,48 x grande	14	120
g4ad.xlarge	1	12
g4ad.2xlarge	1	12
g4ad.4xlarge	2	12
g4ad.8xlarge	3	12
g4ad.16xlarge	7	12
g5.xlarge	3	6
g5.2xlarge	3	19
g5.4xlarge	7	40
g5.8xlarge	7	90
g5.12xlarge	14	120
g5.16xlarge	7	120
g5.24xlarge	14	120
g5.48xlarge	6	120
g5g.xlarge	3	20

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
g5g.2xlarge	3	40
g5g.4xlarge	7	60
g5g.8xlarge	7	60
g5g.16xlarge	14	120
g5g.metal	14	120
g6.x grande	3	20
g6,2 x grande	3	40
g 6,4 x grande	7	60
g 6,8 x grande	7	90
g 6,12 x grande	7	120
g6,16 x grande	14	120
g6,24 x grande	14	120
g6,48 x grande	14	120
g6e.xlarge	3	20
g6e.2x grande	3	40
g6e.4x grande	7	60
g6e.8x grande	7	90
g6e.12 x grande	9	120
g6 e.16 x grande	14	120
g 6 e.24 x grande	19	120

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
G6 e.48 x grande	39	120
gr 6,4 x grande	7	60
gr 6,8 x grande	7	90
inf2.xlarge	3	20
inf2.8xlarge	7	90
inf2.24xlarge	14	120
inf2.48xlarge	14	120
p4d.24xlarge	59	120
p4de.24xlarge	59	120
p5.48xlarge	63	242
p 5. 48 x grande	63	242
p5en. 48 x grande	63	242
p6-b 200. 48 x grande	31	242
trn1.2xlarge	3	19
trn1.32xlarge	39	120
trn1n.32xlarge	79	242
trn 2,48 x grande	31	242
trn 2u.48 x grande	31	242
vt1.3xlarge	3	40
vt1.6xlarge	7	60

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
vt1.24xlarge	14	120

High Performance Computing

Tipo di istanza	Limite di attività senza ENI canalizzazione	Limite di attività con ENI canalizzazione
hpc6a.48xlarge	1	120
hpc6id.32xlarge	1	120
hpc7g.4xlarge	3	120
hpc7g.8xlarge	3	120
hpc7g.16xlarge	3	120

Riservare la memoria delle istanze del contenitore Amazon ECS Linux

Quando l'agente container Amazon ECS registra un'istanza di contenitore in un cluster, deve determinare la quantità di memoria disponibile nell'istanza del contenitore da riservare per le tue attività. A causa del sovraccarico di memoria della piattaforma e della memoria occupata dal kernel di sistema, questo numero è diverso dalla quantità di memoria installata pubblicizzata per le istanze Amazon. EC2 Ad esempio, un'istanza `m4.large` dispone di 8 GiB di memoria installata. Tuttavia, ciò non sempre si traduce esattamente in 8192 MiB di memoria disponibili per le attività quando l'istanza del contenitore si registra.

L'agente container Amazon ECS fornisce una variabile di configurazione chiamata `ECS_RESERVED_MEMORY`, che puoi usare per rimuovere un numero specifico di MiB di memoria dal pool assegnato alle tue attività. In questo modo si riserva la memoria per i processi di sistema critici.

Se occupi tutta la memoria di un'istanza del contenitore per le tue attività, è possibile che queste abbiano a che fare con processi di sistema critici per la memoria e che possano causare un errore di sistema.

Ad esempio, se si specifica `ECS_RESERVED_MEMORY=256` nel file di configurazione dell'agente del container, l'agente registra la memoria totale -256 MiB per quell'istanza e 256 MiB di memoria non possono essere allocati da task ECS. Per ulteriori informazioni sulle variabili dell'agente e su come impostarle, vedere [Configurazione dell'agente del container Amazon ECS](#) e [Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati](#).

Se specifichi 8192 MiB per l'attività e nessuna delle istanze del contenitore dispone di 8192 MiB o più di memoria disponibile per soddisfare questo requisito, l'attività non può essere inserita nel cluster. Se si utilizza un ambiente di elaborazione gestito, è AWS Batch necessario avviare un tipo di istanza più grande per soddisfare la richiesta.

È inoltre necessario riservare parte della memoria per l'agente del container di Amazon ECS e per altri processi di sistema critici nelle istanze di container, in modo che i container dell'attività non si contendano la stessa memoria e finiscano con l'attivare un errore di sistema.

L'agente del container di Amazon ECS utilizza la funzione `Docker ReadMemInfo()` per eseguire una query sulla memoria totale disponibile per il sistema operativo. Sia Linux che Windows forniscono utilità da riga di comando per determinare la memoria totale.

Example - Determinare la memoria totale in Linux

Il comando `free` restituisce la memoria totale riconosciuta dal sistema operativo.

```
$ free -b
```

Esempio di output per un'istanza `m4.large` che esegue l'AMI Amazon Linux ottimizzata per Amazon ECS.

	total	used	free	shared	buffers	cached
Mem:	8373026816	348180480	8024846336	90112	25534464	205418496
-/+ buffers/cache:		117227520	8255799296			

Questa istanza ha 8373026816 byte di memoria totale, che si traducono in 7985 MiB disponibili per le attività.

Example - Determinare la memoria totale in Windows

Il comando `wmic` restituisce la memoria totale riconosciuta dal sistema operativo.

```
C:\> wmic ComputerSystem get TotalPhysicalMemory
```

Esempio di output per un'istanza `m4.large` che esegue l'AMI Windows Server ottimizzata per Amazon ECS.

```
TotalPhysicalMemory  
8589524992
```

Questa istanza ha 8589524992 byte di memoria totale, che si traducono in 8191 MiB disponibili per le attività.

Visualizzazione della memoria dell'istanza del contenitore

Puoi visualizzare la quantità di memoria con cui registra un'istanza del contenitore nella console Amazon ECS (o con l'operazione [DescribeContainerInstances](#) API). Se stai cercando di massimizzare l'utilizzo delle risorse fornendo alle tue attività quanta più memoria possibile per un particolare tipo di istanza, puoi osservare la memoria disponibile per quell'istanza del contenitore e quindi assegnare alle tue attività quella quantità di memoria.

Per visualizzare la memoria dell'istanza del contenitore

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel riquadro di navigazione, scegli Cluster, quindi scegli il cluster che ospita l'istanza del contenitore.
3. Scegli Infrastruttura, quindi in Istanze di container, scegli un'istanza di contenitore.
4. La sezione Risorse mostra la memoria registrata e disponibile per l'istanza del contenitore.

Il valore di memoria registrata è quello dell'istanza del contenitore, registrato con Amazon ECS al primo avvio, e il valore di memoria disponibile è quello che non è già stato assegnato alle attività.

Gestione remota delle istanze di container Amazon ECS tramite AWS Systems Manager

Puoi utilizzare la funzionalità Run Command in AWS Systems Manager (Systems Manager) per gestire in modo sicuro e remoto la configurazione delle tue istanze di container Amazon ECS. Run Command fornisce un modo semplice per eseguire le attività di amministrazione comuni senza accedere localmente all'istanza. Puoi gestire le modifiche della configurazione dei tuoi cluster tramite l'esecuzione simultanea di comandi su più istanze di container. Run Command tiene traccia dello stato e dei risultati di ciascun comando.

Di seguito sono elencati alcuni esempi dei tipi di attività che puoi eseguire mediante Run Command:

- Installare o disinstallare pacchetti.
- Eseguire aggiornamenti di sicurezza.
- Eliminare immagini Docker.
- Interrompere o avviare i servizi.
- Visualizzare le risorse di sistema.
- Visualizzare file di log.
- Eseguire operazioni sui file.

Per ulteriori informazioni su Run Command, consulta [AWS Systems Manager Run Command](#) nella Guida per l'utente di AWS Systems Manager .

Di seguito sono riportati i prerequisiti per l'utilizzo di Systems Manager con Amazon ECS.

1. È necessario concedere all'istanza del contenitore le autorizzazioni role (`ecsInstanceRole`) per accedere a Systems Manager APIs. Puoi farlo assegnando Amazon SSMManaged InstanceCore al `ecsInstanceRole` ruolo. Per informazioni su come allegare una policy a un ruolo, consulta [Aggiornare le autorizzazioni per un ruolo nella Guida](#) per l'AWS Identity and Access Management utente
2. Verifica che SSM Agent sia installato nelle istanze di container. Per ulteriori informazioni, consulta [Installazione e disinstallazione manuale di SSM Agent su EC2 istanze](#) per Linux.

Dopo aver collegato le policy gestite di Systems Manager alle istanze del contenitore `ecsInstanceRole` e aver verificato che AWS Systems Manager l'agente (agente SSM) sia installato sulle istanze del contenitore, è possibile iniziare a utilizzare Run Command per inviare comandi alle istanze del contenitore. Per ulteriori informazioni sull'esecuzione di comandi e script di shell sulle istanze e sulla visualizzazione dell'output risultante, consulta [Esecuzione di comandi mediante Run Command di Systems Manager](#) e [Spiegazione passo per passo di Run Command](#) nella Guida per l'utente di AWS Systems Manager .

Un caso d'uso comune consiste nell'aggiornare il software delle istanze del contenitore con Run Command. È possibile seguire le procedure riportate nella Guida per l' AWS Systems Manager utente con i seguenti parametri.

Parametro	Valore	
Documento di comando	AWS-RunShellScript	
Comando	<pre>\$ yum update -y</pre>	
Istanze target	Le tue istanze di container	

Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux

Puoi configurare le tue istanze di container Amazon ECS per l'utilizzo di un proxy HTTP sia per l'agente del container Amazon ECS che per il daemon Docker. Ciò è utile se le tue istanze di container non hanno accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

Per configurare l'istanza del contenitore Amazon ECS Linux per l'utilizzo di un proxy HTTP, imposta le seguenti variabili nei file pertinenti al momento dell'avvio (con i dati EC2 utente Amazon). Puoi anche modificare manualmente il file di configurazione e quindi riavviare l'agente.

`/etc/ecs/ecs.config` (Amazon Linux 2 e AmazonLinux AMI)

```
HTTP_PROXY=10.0.0.131:3128
```

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per l'agente Amazon ECS per la connessione a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
NO_PROXY=169.254.169.254,169.254.170.2,/var/run/docker.sock
```

Imposta questo valore su `169.254.169.254,169.254.170.2,/var/run/docker.sock` filtrare i metadati delle EC2 istanze, i ruoli IAM per le attività e il traffico del demone Docker dal proxy.

`/etc/systemd/system/ecs.service.d/http-proxy.conf` (Amazon Linux 2 solo)

```
Environment="HTTP_PROXY=10.0.0.131:3128/"
```

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per connettere `ecs-init` a Internet. Ad esempio, le tue istanze di container

potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
Environment="NO_PROXY=169.254.169.254,169.254.170.2,/var/run/docker.sock"
```

Imposta questo valore su per 169.254.169.254,169.254.170.2,/var/run/docker.sock filtrare i metadati delle EC2 istanze, i ruoli IAM per le attività e il traffico del demone Docker dal proxy.

/etc/init/ecs.override (solo AMI Amazon Linux)

```
env HTTP_PROXY=10.0.0.131:3128
```

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per connettere ecs-init a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
env NO_PROXY=169.254.169.254,169.254.170.2,/var/run/docker.sock
```

Imposta questo valore su per 169.254.169.254,169.254.170.2,/var/run/docker.sock filtrare i metadati delle EC2 istanze, i ruoli IAM per le attività e il traffico del demone Docker proveniente dal proxy.

/etc/systemd/system/docker.service.d/http-proxy.conf (Amazon Linux 2 solo)

```
Environment="HTTP_PROXY=http://10.0.0.131:3128"
```

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per connettere il daemon Docker a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
Environment="NO_PROXY=169.254.169.254,169.254.170.2"
```

Imposta questo valore su per 169.254.169.254,169.254.170.2 filtrare i metadati dell'EC2istanza dal proxy.

/etc/sysconfig/docker(AMI Amazon Linux e Amazon Linux 2 solo)

```
export HTTP_PROXY=http://10.0.0.131:3128
```

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per connettere il daemon Docker a Internet. Ad esempio, le tue istanze di

container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
export NO_PROXY=169.254.169.254,169.254.170.2
```

Imposta questo valore su per 169.254.169.254,169.254.170.2 filtrare i metadati dell' EC2istanza dal proxy.

L'impostazione di queste variabili di ambiente nei suddetti file influisce solo sull'agente del container di Amazon ECS, su `ecs-init` e sul daemon Docker. Non configurano altri servizi (ad esempio il comando `yum`) per l'utilizzo del proxy.

Per informazioni su come configurare il proxy, consulta [Come configurare un proxy HTTP per Docker e l'agente contenitore Amazon ECS in Amazon Linux 2](#) o 023. AL2

Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling

Amazon ECS supporta i pool caldi di Amazon EC2 Auto Scaling. Un pool caldo è un gruppo di EC2 istanze Amazon preinizializzate pronte per essere messe in servizio. Ogni volta che l'applicazione deve essere scalata orizzontalmente, Amazon EC2 Auto Scaling utilizza le istanze preinizializzate del pool caldo anziché lanciare istanze fredde, consente l'esecuzione di qualsiasi processo di inizializzazione finale e quindi mette l'istanza in servizio.

Per ulteriori informazioni sulle piscine calde e su come aggiungere una piscina calda al tuo gruppo Auto Scaling, consulta [Warm pools for Amazon Auto EC2 Scaling nella Amazon Auto EC2 Scaling User Guide](#).

Quando crei o aggiorni un warm pool per un gruppo con scalabilità automatica per Amazon ECS, non puoi impostare l'opzione che restituisce le istanze al warm pool alla riduzione orizzontale (`ReuseOnScaleIn`). Per ulteriori informazioni, consulta [put-warm-pool](#) nella documentazione di riferimento AWS Command Line Interface .

Per utilizzare pool caldi con il tuo cluster Amazon ECS, imposta la variabile di configurazione dell'ECS_WARM_POOLS_CHECK agente `true` nel campo Dati utente del modello di lancio del gruppo Amazon EC2 Auto Scaling.

Di seguito viene mostrato un esempio di come la variabile di configurazione dell'agente può essere specificata nel campo Dati utente di un modello di EC2 lancio di Amazon. Sostituiscilo *MyCluster* con il nome del tuo cluster.

```
#!/bin/bash
```



```
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_WARM_POOLS_CHECK=true
EOF
```

La variabile ECS_WARM_POOLS_CHECK è supportata sull'agente solo a partire dalla versione 1.59.0. Per ulteriori informazioni sulle variabili, consulta [Configurazione dell'agente del container Amazon ECS](#).

Aggiornamento dell'agente del container Amazon ECS

Occasionalmente, potrebbe essere necessario aggiornare l'agente container Amazon ECS per rilevare correzioni di bug e nuove funzionalità. L'aggiornamento dell'agente del container di Amazon ECS non interrompe i processi o i servizi in esecuzione nell'istanza di container. Il processo per l'aggiornamento dell'agente differisce a seconda se la tua istanza di container è stata avviata con l'AMI ottimizzata per Amazon ECS o con un altro sistema operativo.

Note

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Verifica della versione dell'agente del container Amazon ECS

Puoi verificare la versione dell'agente del container che è in esecuzione nelle tue istanze di container per vedere se è necessario aggiornarla. La vista dell'istanza di container nella console di Amazon ECS fornisce la versione dell'agente. Utilizza la procedura seguente per verificare la versione del tuo agente.

Amazon ECS console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Dalla barra di navigazione, scegli la Regione in cui l'istanza esterna è registrata.
3. Nel riquadro di navigazione, scegli Cluster e seleziona il cluster che ospita l'istanza esterna.
4. Nella **name** pagina Cluster:, scegli la scheda Infrastruttura.
5. In Container instances (Istanze di container), osserva la colonna Agent version (Versione agente) per le tue istanze di container. Se l'istanza di container non contiene la versione

più recente dell'agente del container, la console invia un avviso con un messaggio e contrassegna la versione dell'agente obsoleta.

Se la versione del tuo agente del container è obsoleta, puoi aggiornare l'agente con le seguenti procedure:

- Se la tua istanza di container sta eseguendo un'AMI ottimizzata per Amazon ECS, consulta [Come aggiornare l'agente del container Amazon ECS su un'AMI ottimizzata per Amazon ECS](#).
- Se la tua istanza di container non sta eseguendo un'AMI ottimizzata per Amazon ECS, consulta [Aggiornamento manuale dell'agente container Amazon ECS \(per prodotti non ottimizzati per Amazon ECS\) AMIs](#).

 Important

Per aggiornare la versione dell'agente di Amazon ECS da versioni precedenti alla v1.0.0 sull'AMI ottimizzata per Amazon ECS, ti consigliamo di terminare la tua attuale istanza di container e avviare una nuova istanza con la versione dell'AMI più recente. Qualsiasi istanza di container che utilizza una versione di anteprima deve essere ritirata e sostituita con l'AMI più recente. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Amazon ECS container agent introspection API

Puoi inoltre utilizzare l'API di introspezione dell'agente del container di Amazon ECS per verificare la versione dell'agente dall'istanza di container stessa. Per ulteriori informazioni, consulta [Introspezione dei container Amazon ECS](#).

Come verificare se il tuo agente del container di Amazon ECS sta eseguendo l'ultima versione con l'API di introspezione

1. Accedi alla tua istanza di container con SSH.
2. Interroga l'API di introspezione.

```
[ec2-user ~]$ curl -s 127.0.0.1:51678/v1/metadata | python3 -mjson.tool
```

 Note

L'API di introspezione ha aggiunto delle informazioni di `Version` nella versione v1.0.0 dell'agente del container di Amazon ECS. Se `Version` non è presente quando si interroga l'API di introspezione oppure se quest'ultima è completamente assente nel tuo agente, la versione in esecuzione è v0.0.3 o precedente. È necessario aggiornare la versione.

Come aggiornare l'agente del container Amazon ECS su un'AMI ottimizzata per Amazon ECS

Se utilizzi l'AMI ottimizzata per Amazon ECS, hai diverse opzioni per ottenere la versione più recente dell'agente del container di Amazon ECS (mostrato in ordine di raccomandazione):

- Termina l'istanza di container e avvia la versione più recente dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS (manualmente o tramite l'aggiornamento della configurazione di avvio di Auto Scaling con l'AMI più recente). Ciò fornisce una nuova istanza di contenitore con le versioni testate e convalidate più recenti di Amazon Linux, Docker e `ecs-init` l'agente container Amazon ECS. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).
- Connettiti all'istanza con SSH e aggiorna il pacchetto `ecs-init` (e le relative dipendenze) alla versione più recente. Questa operazione fornisce le versioni più recenti testate e convalidate di Docker e `ecs-init` che sono disponibili in Amazon Linux repository e la versione più recente dell'agente container Amazon ECS. Per ulteriori informazioni, consulta [Come aggiornare il pacchetto `ecs-init` su un'AMI ottimizzata per Amazon ECS](#).
- Aggiorna l'agente del contenitore con il funzionamento dell'`UpdateContainerAgentAPI`, tramite la console o con o. AWS CLI AWS SDKs Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container di Amazon ECS con l'operazione `API UpdateContainerAgent`](#).

 Note

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Come aggiornare il pacchetto **ecs-init** su un'AMI ottimizzata per Amazon ECS

1. Accedi alla tua istanza di container con SSH.
2. Aggiorna il pacchetto `ecs-init` con il comando seguente.

```
sudo yum update -y ecs-init
```

Note

Il pacchetto `ecs-init` e l'agente del container di Amazon ECS vengono aggiornati immediatamente. Tuttavia, le versioni più recenti di Docker non vengono caricate finché il daemon Docker non viene riavviato. Utilizza il riavvio dell'istanza oppure esegui i comandi seguenti sull'istanza:

- AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart docker
```

- AMI Amazon Linux ottimizzata per Amazon ECS:

```
sudo service docker restart && sudo start ecs
```

Aggiornamento dell'agente del container di Amazon ECS con l'operazione API **UpdateContainerAgent**

Important

L'API `UpdateContainerAgent` è supportata solo sulle varianti Linux dell'AMI ottimizzata per Amazon ECS, ad eccezione dell'AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS. Per le istanze di container che utilizzano l'AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS, aggiorna il pacchetto `ecs-init` per aggiornare l'agente. Per le istanze di container che eseguono altri sistemi operativi, consulta [Aggiornamento manuale dell'agente container Amazon ECS \(per prodotti non ottimizzati per Amazon ECS\) AMIs](#). Se utilizzi istanze di container Windows, ti consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei tuoi cluster Windows.

Il processo `UpdateContainerAgent` API inizia quando richiedi l'aggiornamento di un agente, tramite la console o con l'opzione AWS CLI o AWS SDKs. Amazon ECS verifica la versione corrente dell'agente confrontandola con l'ultima versione disponibile dell'agente e verifica se è possibile effettuare un aggiornamento. Se non è disponibile alcun aggiornamento, ad esempio se l'agente sta già eseguendo la versione più recente, allora viene restituito `NoUpdateAvailableException`.

Le fasi del processo di aggiornamento riportate in precedenza sono le seguenti:

PENDING

È disponibile un aggiornamento dell'agente e il processo di aggiornamento è stato avviato.

STAGING

L'agente ha iniziato a scaricare il relativo aggiornamento. Se l'agente non è in grado di scaricare l'aggiornamento, oppure se i contenuti dell'aggiornamento non sono corretti o sono danneggiati, l'agente invia una notifica dell'errore e l'aggiornamento passa allo stato `FAILED`.

STAGED

Il download dell'agente è stato completato e i contenuti dell'agente sono stati verificati.

UPDATING

Il servizio `ecs-init` è stato riavviato e ottiene la nuova versione dell'agente. Se l'agente non è in grado di riavviarsi per qualsiasi motivo, l'aggiornamento passa allo stato `FAILED`. In caso contrario, l'agente segnala ad Amazon ECS che l'aggiornamento è completo.

Note

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Come aggiornare l'agente del container Amazon ECS su un'AMI ottimizzata per Amazon ECS nella console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Dalla barra di navigazione, scegli la Regione in cui l'istanza esterna è registrata.
3. Nel pannello di navigazione, seleziona Clusters (Cluster), quindi seleziona il cluster.

4. Nella **name** pagina Cluster:, scegli la scheda Infrastruttura.
5. In Istanze di container, seleziona le istanze da aggiornare, quindi scegli Operazioni, Aggiorna agente.

Aggiornamento manuale dell'agente container Amazon ECS (per prodotti non ottimizzati per Amazon ECS) AMIs

Occasionalmente, potrebbe essere necessario aggiornare l'agente container Amazon ECS per rilevare correzioni di bug e nuove funzionalità. L'aggiornamento dell'agente del container di Amazon ECS non interrompe i processi o i servizi in esecuzione nell'istanza di container.

Note

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

1. Accedi alla tua istanza di container con SSH.
2. Controlla se il tuo agente utilizza la variabile di ambiente ECS_DATADIR per salvare il suo stato.

```
ubuntu:~$ docker inspect ecs-agent | grep ECS_DATADIR
```

Output:

```
"ECS_DATADIR=/data",
```

Important

Se il comando precedente non restituisce la variabile di ambiente ECS_DATADIR, è necessario arrestare qualsiasi attività in esecuzione in questa istanza di container prima di aggiornare l'agente. Gli agenti più recenti con la variabile di ambiente ECS_DATADIR salvano il proprio stato e puoi aggiornarli mentre le attività vengono eseguite senza problemi.

3. Arresta l'agente del container di Amazon ECS.

```
ubuntu:~$ docker stop ecs-agent
```

4. Elimina l'agente del container.

```
ubuntu:~$ docker rm ecs-agent
```

5. Assicurati che la directory `/etc/ecs` e il file di configurazione dell'agente del container di Amazon ECS esistano in `/etc/ecs/ecs.config`.

```
ubuntu:~$ sudo mkdir -p /etc/ecs && sudo touch /etc/ecs/ecs.config
```

6. Modifica il file `/etc/ecs/ecs.config` e assicurati che contenga almeno le seguenti dichiarazioni di variabili. Se non vuoi che la tua istanza di container sia registrata con il cluster predefinito, specifica il nome del tuo cluster come il valore per `ECS_CLUSTER`.

```
ECS_DATADIR=/data
ECS_ENABLE_TASK_IAM_ROLE=true
ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST=true
ECS_LOGFILE=/log/ecs-agent.log
ECS_AVAILABLE_LOGGING_DRIVERS=["json-file", "awslogs"]
ECS_LOGLEVEL=info
ECS_CLUSTER=default
```

Per ulteriori informazioni su queste e altre opzioni di runtime dell'agente, consulta [Configurazione dell'agente del container Amazon ECS](#).

Note

Facoltativamente, puoi archiviare le variabili di ambiente dell'agente in Amazon S3 (che può essere scaricata nelle istanze del contenitore al momento del lancio utilizzando i dati utente di EC2 Amazon). Ciò è consigliato per le informazioni sensibili, come le credenziali di autenticazione per i repository privati. Per ulteriori informazioni, consultare [Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3](#) e [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

7. Estrai l'ultima immagine dell'agente del container di Amazon ECS da Amazon Elastic Container Registry Public.

```
ubuntu:~$ docker pull public.ecr.aws/ecs/amazon-ecs-agent:latest
```

Output:

```
Pulling repository amazon/amazon-ecs-agent
a5a56a5e13dc: Download complete
511136ea3c5a: Download complete
9950b5d678a1: Download complete
c48ddcf21b63: Download complete
Status: Image is up to date for amazon/amazon-ecs-agent:latest
```

8. Esegui l'agente del container di Amazon ECS più recente sulla tua istanza di container.

Note

Utilizza le policy di riavvio di Docker o un gestore di processo (come upstart o systemd) per trattare l'agente del container come un servizio o un daemon e assicurarne il riavvio dopo l'uscita. L'AMI ottimizzata per Amazon ECS utilizza l'`ecs-init` RPM per questo scopo e puoi visualizzare il [codice sorgente di questo](#) RPM su GitHub.

Nell'esempio seguente, il comando di esecuzione dell'agente è suddiviso in righe separate per mostrare ciascuna opzione. Per ulteriori informazioni su queste e altre opzioni di runtime dell'agente, consulta [Configurazione dell'agente del container Amazon ECS](#).

Important

I sistemi operativi con SELinux abilitato richiedono l'`--privileged` opzione nel comando `docker run`. Inoltre, per le istanze SELinux di container abilitate, si consiglia di aggiungere l'`:Z` opzione ai montaggi `/log` e `/data` volume. Tuttavia, i montaggi degli host per questi volumi devono esistere prima che tu esegua il comando o che riceva un errore `no such file or directory`. Se riscontri difficoltà nell'esecuzione dell'agente Amazon ECS su un'istanza di container SELinux abilitata, intraprendi la seguente azione:

- Crea i punti di montaggio dei volumi degli host sulla tua istanza di container.

```
ubuntu:~$ sudo mkdir -p /var/log/ecs /var/lib/ecs/data
```


- Aggiungi l'opzione `--privileged` al comando `docker run` seguente.
- Aggiungi l'opzione `:Z` ai montaggi dei volumi dei container `/log` e `/data` (ad esempio `--volume=/var/log/ecs/:/log:Z`) al comando `docker run` seguente.

```
ubuntu:~$ sudo docker run --name ecs-agent \  
--detach=true \  
--restart=on-failure:10 \  
--volume=/var/run:/var/run \  
--volume=/var/log/ecs/:/log \  
--volume=/var/lib/ecs/data:/data \  
--volume=/etc/ecs:/etc/ecs \  
--volume=/etc/ecs:/etc/ecs/pki \  
--net=host \  
--env-file=/etc/ecs/ecs.config \  
amazon/amazon-ecs-agent:latest
```

Note

Se ricevi un messaggio `Error response from daemon: Cannot start container`, puoi eliminare il container che presenta l'errore con il comando `sudo docker rm ecs-agent` e tentare di eseguire nuovamente l'agente.

Windows ottimizzato per Amazon ECS AMIs

I modelli ottimizzati per Amazon ECS AMIs sono preconfigurati con i componenti necessari per eseguire i carichi di lavoro Amazon ECS. Sebbene tu possa creare la tua AMI di istanza di container che soddisfi le specifiche di base necessarie per eseguire i carichi di lavoro containerizzati su Amazon ECS, le AMI ottimizzate per Amazon ECS sono AMIs preconfigurate e testate su Amazon ECS da ingegneri. AWS È il modo più semplice per iniziare ed eseguire i container su AWS rapidamente.

I metadati dell'AMI ottimizzata per Amazon ECS, incluso il nome dell'AMI, la versione dell'agente del container di Amazon ECS e la versione runtime di Amazon ECS, che include la versione Docker, possono essere recuperati a livello di codice per ciascuna variante. Per ulteriori informazioni, consulta [the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”](#).

 Important

Tutte le varianti AMI ottimizzate per ECS prodotte dopo agosto 2022 migreranno da Docker EE (Mirantis) a Docker CE (progetto Moby).

Per garantire che i clienti dispongano degli ultimi aggiornamenti di sicurezza per impostazione predefinita, Amazon ECS mantiene almeno gli ultimi tre aggiornamenti di sicurezza ottimizzati per Windows Amazon AMIs ECS. Dopo aver rilasciato il nuovo Windows ottimizzato per Amazon ECS, AMIs Amazon ECS rende private le versioni precedenti ottimizzate per Amazon ECS AMIs . Se devi accedere a un'AMI privata, comunicacelo compilando un ticket con Cloud Support.

Varianti AMI ottimizzate per Amazon ECS

Le seguenti varianti di Windows Server dell'AMI ottimizzata per Amazon ECS sono disponibili per le tue istanze Amazon EC2 .

 Important

Tutte le varianti AMI ottimizzate per ECS prodotte dopo agosto migreranno da Docker EE (Mirantis) a Docker CE (progetto Moby).

- AMI Windows Server 2022 Full ottimizzata per Amazon ECS
- AMI Windows Server 2022 Core ottimizzata per Amazon ECS
- AMI Windows Server 2019 Full ottimizzata per Amazon ECS
- AMI Windows Server 2019 Core ottimizzata per Amazon ECS
- AMI Windows Server 2016 Full ottimizzata per Amazon ECS

 Important

Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:

- Windows Server 2022 Full

- Windows Server 2022 Core
- Windows Server 2019 Full
- Windows Server 2019 Core

Il 9 agosto 2022, l'AMI Windows Server 20H2 Core ottimizzata per Amazon ECS ha raggiunto la data di fine del supporto. Non verranno rilasciate nuove versioni di questa AMI. Per ulteriori informazioni, consulta [Informazioni sulle versioni di Windows Server](#).

Windows Server 2022, Windows Server 2019 e Windows Server 2016 sono versioni LTSC (Long-Term Servicing Channel). Windows Server 20H2 è una versione SAC (Semi-Annual Channel). Per ulteriori informazioni, consulta [Informazioni sulle versioni di Windows Server](#).

Considerazioni

Ecco alcune cose che dovresti sapere sui contenitori Amazon EC2 Windows e Amazon ECS.

- I container Windows non possono essere eseguiti su istanze di container Linux e viceversa. Per assicurare il corretto posizionamento dei processi Windows e Linux, devi mantenere le relative istanze di container in cluster separati e posizionare solo i processi Windows nei cluster Windows. Puoi assicurare che le definizioni di processo di Windows vengano posizionate solo nelle istanze Windows impostando il seguente vincolo di posizionamento: `memberOf(ec2.os-type=='windows')`.
- I contenitori Windows sono supportati per le attività che utilizzano i tipi di avvio EC2 e Fargate.
- I container Windows e le istanze di container non supportano tutti i parametri di definizione di attività disponibili per le istanze di container e i container Linux. Alcuni parametri non sono supportati, mentre altri si comportano in modo diverso su Windows e su Linux. Per ulteriori informazioni, consulta [Differenze nella definizione delle attività di Amazon ECS per le EC2 istanze che eseguono Windows](#).
- Per la funzionalità dei ruoli IAM per i processi, è necessario configurare le istanze di container Windows per consentire la funzionalità all'avvio. I contenitori devono eseguire il PowerShell codice fornito quando utilizzano la funzionalità. Per ulteriori informazioni, consulta [Configurazione aggiuntiva dell'istanza Amazon EC2 Windows](#).
- La funzionalità dei ruoli IAM per i processi utilizza un proxy di credenziali per fornire le credenziali ai container. Questo proxy di credenziali occupa la porta 80 nell'istanza di container, perciò se utilizzi i ruoli IAM per i processi, la porta 80 non è disponibile per i processi. Per i container di servizi Web, puoi utilizzare un Application Load Balancer e la mappatura di porte dinamiche per

fornire connessioni di porta 80 HTTP standard ai container. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

- Le immagini Docker del server Windows sono di grandi dimensioni (9 GiB). Pertanto, le istanze di container di Windows richiedono più spazio di archiviazione rispetto alle istanze di container Linux.
- Per eseguire un container Windows su Windows Server, la versione del sistema operativo dell'immagine di base del container deve corrispondere a quella dell'host. Per ulteriori informazioni, consulta [Compatibilità con la versione del container Windows](#) sul sito Web della documentazione Microsoft. Se il tuo cluster esegue più versioni di Windows, puoi assicurarti che un'attività venga inserita su un' EC2 istanza in esecuzione sulla stessa versione utilizzando il vincolo di posizionamento: `memberOf(attribute:ecs.os-family == WINDOWS_SERVER_<OS_Release>_<FULL or CORE>)` Per ulteriori informazioni, consulta [the section called "Recupero di metadati AMI Windows ottimizzati per Amazon ECS"](#).

Recupero di metadati AMI Windows ottimizzati per Amazon ECS

L'ID AMI, il nome dell'immagine, il sistema operativo, la versione dell'agente contenitore e la versione di runtime per ogni variante di Amazon ECS Optimized AMIs possono essere recuperati a livello di codice interrogando l'API Systems Manager Parameter Store. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere [GetParameters](#) e [GetParametersByPath](#).

Note

Per recuperare i metadati dell'AMI ottimizzata per Amazon ECS, l'utente di amministrazione deve disporre delle seguenti autorizzazioni IAM. Queste autorizzazioni sono state aggiunte alla policy IAM `AmazonECS_FullAccess`.

- `ssm: GetParameters`
- `ssm: GetParameter`
- `ssm: GetParametersByPath`

Formato del parametro dell'archivio parametri di Systems Manager

Note

I seguenti parametri dell'API Systems Manager Parameter Store sono obsoleti e non devono essere utilizzati per recuperare le versioni più recenti di Windows: AMIs

- `/aws/service/ecs/optimized-ami/windows_server/2016/english/full/recommended/image_id`
- `/aws/service/ecs/optimized-ami/windows_server/2019/english/full/recommended/image_id`

Di seguito è riportato il formato del nome del parametro per ogni variante AMI ottimizzata per Amazon ECS.

- Metadati dell'AMI di Windows Server 2022 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2022 Core:

```
/aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2019 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2019 Core:

```
/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2016 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized
```

Il formato dei nomi di parametro seguente recupera i metadati dell'AMI stabile Windows Server 2019 completa più recente.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized
```

Di seguito viene mostrato un esempio dell'oggetto JSON restituito per il valore del parametro.

```
{
```

```

    "Parameters": [
      {
        "Name": "/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-
ECS_Optimized",
        "Type": "String",
        "Value": "{\"image_name\":\"Windows_Server-2019-English-Full-
ECS_Optimized-2023.06.13\", \"image_id\":\"ami-0debc1fb48e4aee16\", \"ecs_runtime_version
\": \"Docker (CE) version 20.10.21\", \"ecs_agent_version\": \"1.72.0\"}",
        "Version": 58,
        "LastModifiedDate": "2023-06-22T19:37:37.841000-04:00",
        "ARN": "arn:aws:ssm:us-east-1::parameter/aws/service/ami-windows-latest/
Windows_Server-2019-English-Full-ECS_Optimized",
        "DataType": "text"
      }
    ],
    "InvalidParameters": []
  }

```

Ognuno dei campi nell'output riportato sopra sono disponibili per essere interrogati come parametri secondari. Costruisci il percorso di parametro per un parametro secondario aggiungendo il nome del parametro secondario al percorso per l'AMI selezionata. Sono disponibili i seguenti parametri secondari:

- `schema_version`
- `image_id`
- `image_name`
- `os`
- `ecs_agent_version`
- `ecs_runtime_version`

Esempi

Negli esempi seguenti vengono illustrati i modi in cui è possibile recuperare i metadati per ogni variante dell'AMI ottimizzata per Amazon ECS.

Recupero dei metadati dell'AMI ottimizzata per Amazon ECS stabile più recente

Puoi recuperare l'ultima AMI stabile ottimizzata per Amazon ECS utilizzando AWS CLI i AWS CLI seguenti comandi.

- Per l'AMI Windows Server 2022 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2022 Core ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2019 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2019 Core ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2016 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized --region us-east-1
```

Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello AWS CloudFormation facendo riferimento al nome dell'archivio parametri di Systems Manager.

```
Parameters:
  LatestECSOptimizedAMI:
    Description: AMI ID
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized/image_id
```

Versioni AMI Windows ottimizzate per Amazon ECS

Visualizza le versioni attuali e precedenti di Amazon ECS Optimized AMIs e le versioni corrispondenti dell'agente container Amazon ECS, Docker e del pacchetto. `ecs-init`

I metadati dell'AMI ottimizzata per Amazon ECS, incluso l'ID AMI per ogni variante, possono essere recuperati a livello di programmazione. Per ulteriori informazioni, consulta [the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”](#).

Le seguenti schede mostrano un elenco di versioni ottimizzate per Windows Amazon ECS AMIs . Per informazioni dettagliate su come fare riferimento al parametro Systems Manager Parameter Store in un AWS CloudFormation modello, vedere [Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation](#).

Important

Per garantire che i clienti dispongano degli ultimi aggiornamenti di sicurezza per impostazione predefinita, Amazon ECS mantiene almeno gli ultimi tre aggiornamenti di sicurezza ottimizzati per Windows Amazon AMIs ECS. Dopo aver rilasciato il nuovo Windows ottimizzato per Amazon ECS, AMIs Amazon ECS rende private le versioni precedenti ottimizzate per Amazon ECS AMIs . Se devi accedere a un'AMI privata, comunicacelo compilando un ticket con Cloud Support.

Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:

- Windows Server 2022 Full
- Windows Server 2022 Core
- Windows Server 2019 Full
- Windows Server 2019 Core

Windows Server 2022 Full AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2022 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-Ingles-Full-ECS_Optimized-2025.4.14	1.91.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.3.17	1.91.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.2.14	1.90.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.01.15	1.89.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.12.11	1.89.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.11.13	1.88.0	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.10.17	1.87.0	25.0.6 (Docker CE)	Privata

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.09.10	1.86.3	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.08.19	1.86.2	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.07.09	1.84.0	25.0.3 (Docker CE)	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2022 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized
```

Windows Server 2022 Core AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2022 Core ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-Inglese-Core-ECS	1.91.2	25.0.6 (Docker CE)	Pubblica

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
_Optimized-2025.4.14			
Windows_Server-2022-inglese-core-ecs-optimized-2025.3.17	1.91.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs-optimized-2025.2.14	1.90.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs-optimized-2025.01.15	1.89.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs-optimized-2024.12.11	1.89.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs-optimized-2024.11.13	1.88.0	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-core-ecs-optimized-2024.10.17	1.87.0	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-core-ecs-optimized-2024.09.10	1.86.3	25.0.6 (Docker CE)	Privata

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-inglese-core-ecs_optimized-2024.08.19	1.86.2	25.0.6 (Docker CE)	Privata
Windows_Server-2022-inglese-core-ecs_optimized-2024.07.09	1.84.0	25.0.3 (Docker CE)	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2022 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized
```

Windows Server 2019 Full AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2019 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-Inglese-Full-ECS_Optimized-2025.4.14	1.91.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-Inglese-Full-ECS	1.91.0	25.0.6 (Docker CE)	Pubblica

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
_Optimized-2025.3.17			
Windows_Server-2019-Ingles-Full-ECS_Optimized-2025.2.14	1.90.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-Ingles-Full-ECS_Optimized-2025.01.15	1.89.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-Full-ECS_Optimized-2024.12.11	1.89.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-Ingles-Full-ECS_Optimized-2024.11.13	1.88.0	25.0.6 (Docker CE)	Privata
Windows_Server-2019-Ingles-Full-ECS_Optimized-2024.10.17	1.87.0	25.0.6 (Docker CE)	Privata
Windows_Server-2019-Ingles-Full-ECS_Optimized-2024.09.10	1.86.3	25.0.6 (Docker CE)	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2024.08.16	1.86.2	25.0.6 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2024.07.09	1.84.0	25.0.3 (Docker CE)	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2019 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized
```

Windows Server 2019 Core AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2019 Core ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Core-ECS_Optimized-2025.4.14	1.91.2	25.0.6 (Docker CE)	Pubblica

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-inglese-core-ecs_optimized-2025.3.17	1.91.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2025.2.14	1.90.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2025.01.15	1.89.2	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.12.11	1.89.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.11.13	1.88.0	25.0.6 (Docker CE)	Privata
Windows_Server-2019-inglese-core-ecs_optimized-2024.10.17	1.87.0	25.0.6 (Docker CE)	Privata
Windows_Server-2019-inglese-core-ecs_optimized-2024.09.10	1.86.3	25.0.6 (Docker CE)	Privata

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-inglese-core-ecs_optimized-2024.08.19	1.86.2	25.0.6 (Docker CE)	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2019 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized
```

Windows Server 2016 Full AMI versions

Important

Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:

- Windows Server 2022 Full
- Windows Server 2022 Core
- Windows Server 2019 Full
- Windows Server 2019 Core

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2016 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-Ingles-Full-ECS_Optimized-2025.4.14	1.91.2	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-Ingles-Full-ECS_Optimized-2025.3.17	1.91.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-Ingles-Full-ECS_Optimized-2025.2.14	1.90.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-Ingles-Full-ECS_Optimized-2025.01.15	1.89.2	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2024.12.11	1.89.1	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2024.11.13	1.88.0	20.10.23 (Docker CE)	Privata
Windows_Server-2016-inglese-Full-ECS_Optimized-2024.10.17	1.87.0	20.10.23 (Docker CE)	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-inglese-Full-ECS_Optimized-2024.09.10	1.86.3	20.10.23 (Docker CE)	Privata
Windows_Server-2016-Inglese-Full-ECS_Optimized-2024.08.19	1.86.2	20.10.23 (Docker CE)	Privata
Windows_Server-2016-inglese-Full-ECS_Optimized-2024.07.09	1.84.0	20.10.23 (Docker CE)	Privata

Utilizza la seguente AMI completa per Windows Server 2016 ottimizzata per AWS CLI Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized
```

Creazione dell'AMI Windows ottimizzata per Amazon ECS

Usa EC2 Image Builder per creare la tua AMI Windows personalizzata ottimizzata per Amazon ECS. In questo modo è facile utilizzare un'AMI di Windows con la propria licenza su Amazon ECS. Amazon ECS fornisce un componente di Image Builder gestito che fornisce la configurazione del sistema necessaria per eseguire istanze di Windows per ospitare i container. Ogni componente gestito da Amazon ECS include un agente del container specifico e una versione Docker. Puoi personalizzare l'immagine in modo da utilizzare l'ultimo componente gestito da Amazon ECS oppure, se è necessario un agente del container o una versione Docker precedente, puoi specificare un componente diverso.

Per una guida completa sull'utilizzo di Image EC2 Builder, [consulta la Guida introduttiva a Image EC2 Builder nella Guida per l'EC2 utente di Image Builder](#).

Quando crei un'AMI Windows ottimizzata per Amazon ECS utilizzando Image EC2 Builder, crei una ricetta per le immagini. La tua ricetta per le immagini deve soddisfare i seguenti requisiti:

- L'immagine sorgente deve essere basata su Windows Server 2019 Core, Windows Server 2019 Full, Windows Server 2022 Core o Windows Server 2022 Full. Altre eventuali versioni di Windows non sono supportate e potrebbero non essere compatibili con il componente.
- Quando si specifica l'opzione Crea componenti, il componente `ecs-optimized-ami-windows` è obbligatorio. Il componente `update-windows`, che garantisce che l'immagine contenga gli aggiornamenti della sicurezza più recenti, è consigliato.

Per specificare una versione diversa del componente, espandi il menu Opzioni di controllo delle versioni e specifica la versione del componente da utilizzare. Per ulteriori informazioni, consulta [Elenco delle versioni del componente `ecs-optimized-ami-windows`](#).

Elenco delle versioni del componente **ecs-optimized-ami-windows**

Quando si crea una ricetta di EC2 Image Builder e si specifica il `ecs-optimized-ami-windows` componente, è possibile utilizzare l'opzione predefinita o specificare una versione specifica del componente. Per determinare le versioni dei componenti che sono disponibili, insieme alle versioni dell'agente del container di Amazon ECS e del Docker contenute nel componente, puoi utilizzare la AWS Management Console.

Come elencare le versioni del componente **ecs-optimized-ami-windows** disponibili

1. Aprire la console EC2 Image Builder all'indirizzo. <https://console.aws.amazon.com/imagebuilder/>
2. Nella barra di navigazione seleziona la regione in cui viene creata l'immagine.
3. Nel pannello di navigazione, sotto il menu Configurazioni salvate, scegli Componenti.
4. Sulla pagina Componenti, nella barra di ricerca digita `ecs-optimized-ami-windows` ed espandi verso il basso il menu di qualificazione, quindi seleziona Avvio rapido (gestito da Amazon).
5. Utilizza la colonna Descrizione per determinare la versione del componente con l'agente del container Amazon ECS e la versione Docker richiesta dall'immagine.

Gestione delle istanze di container Amazon ECS Windows

Quando utilizzi EC2 istanze per i tuoi carichi di lavoro Amazon ECS, sei responsabile della manutenzione delle istanze.

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Procedure di gestione

- [Avvio di un'istanza di container Windows di Amazon ECS](#)
- [Avvio delle istanze di container Amazon ECS Windows per il trasferimento di dati](#)
- [Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Windows](#)
- [Configurazione delle istanze di container Amazon ECS Windows per ricevere avvisi sulle istanze Spot](#)

Avvio di un'istanza di container Windows di Amazon ECS

Le tue istanze di container Amazon ECS vengono create utilizzando la console Amazon EC2 . Prima di iniziare, devi accertarti di aver completato le fasi in [Configurazione per l'uso di Amazon ECS](#).

Per ulteriori informazioni sulla procedura guidata di avvio, consulta [Launch an instance using the new launch instance wizard](#) nella Amazon EC2 User Guide.

Puoi utilizzare la nuova EC2 procedura guidata di Amazon per avviare un'istanza. È possibile utilizzare il seguente elenco per i parametri e lasciare i parametri non elencati come predefiniti. Le seguenti istruzioni sono relative a ogni gruppo di parametri.

Procedura

Prima di iniziare, completa i passaggi descritti in [Configurazione per l'uso di Amazon ECS](#).

1. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
2. Nella barra di navigazione nella parte superiore dello schermo, viene visualizzata la AWS regione corrente (ad esempio, Stati Uniti orientali (Ohio)). Selezionare una regione in cui avviare l'istanza. Questa scelta è importante perché alcune EC2 risorse Amazon possono essere condivise tra regioni, mentre altre no.
3. Dalla dashboard della EC2 console Amazon, scegli Launch instance.

Nome e tag

Il nome dell'istanza è un tag, dove la chiave è Name (Nome) e il valore è il nome specificato. È possibile aggiungere tag all'istanza, ai volumi e alla grafica elastica. Per le istanze spot, è possibile aggiungere un tag solo alla richiesta di istanza spot.

La specifica di un nome di istanza e dei tag aggiuntivi è facoltativa.

- Per Name (Nome), inserire un nome descrittivo per l'istanza. Se non si specifica un nome, l'istanza può essere identificata dal relativo ID, che viene generato automaticamente all'avvio dell'istanza.
- Per aggiungere altri tag, scegliere Add additional tags (Aggiungi altri tag). Scegliere Add tag (Aggiungi tag), quindi immettere una chiave e un valore e selezionare il tipo di risorsa da taggare. Scegliere Add tag (Aggiungi tag) per ogni tag aggiuntivo.

Immagini di applicazioni e sistema operativo (Amazon Machine Image)

Un'Amazon Machine Image (AMI) contiene tutte le informazioni necessarie per creare un'istanza. Ad esempio, un'AMI può contenere il software necessario per fungere da server Web, ad esempio Apache e il sito Web.

Per le ultime versioni ottimizzate per Amazon ECS AMIs e i relativi valori, consulta Windows AMI ottimizzata per [Amazon ECS](#).

Utilizza la barra di ricerca per trovare un'AMI ottimizzata per Amazon ECS adatta pubblicata da AWS

1. In base alle tue esigenze, inserisci una delle seguenti opzioni AMIs nella barra di ricerca e premi Invio.
 - Windows_Server-2022-English-Full-ECS_Optimized
 - Windows_Server-2022-English-Core-ECS_Optimized
 - Windows_Server-2019-English-Full-ECS_Optimized
 - Windows_Server-2019-English-Core-ECS_Optimized
 - Windows_Server-2016-English-Full-ECS_Optimized
2. Nella pagina Choose an Amazon Machine Image (AMI), seleziona la AMIs scheda Community.
3. Dall'elenco visualizzato, scegli un'AMI verificata da Microsoft con la data di pubblicazione più recente e fai clic su Select (seleziona).

Tipo di istanza

Il tipo di istanza definisce la configurazione hardware e le dimensioni dell'istanza. I tipi di istanza più grandi dispongono di una maggiore quantità di CPU e memoria. Per ulteriori informazioni consulta la sezione relativa ai [tipi di istanza](#).

- Per Tipo di istanza, selezionare il tipo di istanza per l'istanza.

Il tipo di istanza che selezioni determina le risorse disponibili per l'esecuzione delle attività.

Coppia di chiavi (login)

In Key pair name (Nome della coppia di chiavi), scegliere una coppia di chiavi esistente oppure scegliere Create new key pair (Crea nuova coppia di chiavi) per crearne una nuova.

Important

Se si sceglie l'opzione Proceed without key pair (Not recommended) (Procedi senza una coppia di chiavi [non consigliato]), non sarà possibile connetterti all'istanza a meno che non si scelga un'AMI configurata per offrire agli utenti un metodo di accesso alternativo.

Impostazioni di rete

Configurare le impostazioni di rete, se necessario.

- Networking platform Piattaforma di rete): scegli Virtual Private Cloud (VPC), quindi specifica la sottorete nella sezione Network interfaces (Interfacce di rete).
- VPC: selezionare un VPC esistente in cui creare il gruppo di sicurezza.
- Sottorete: è possibile avviare un'istanza in una sottorete associata a una zona di disponibilità, una Local Zone, una zona Wavelength o un Outpost.

Per avviare l'istanza in una zona di disponibilità, selezionare la sottorete in cui avviare l'istanza. Per creare una nuova sottorete, scegliere Create new subnet (Crea nuova sottorete) per passare alla console Amazon VPC. Al termine, tornare alla procedura guidata di avvio istanza e scegliere Refresh (Aggiorna) per caricare la sottorete nell'elenco.

Per avviare l'istanza in una Local Zone, selezionare una sottorete creata nella Local Zone.

Per avviare un'istanza in un Outpost, selezionare una sottorete in un VPC associato a un Outpost.

- **Auto-assign Public IP** (IP pubblico di assegnazione automatica): se l'istanza deve essere accessibile da Internet, verifica che il campo Auto-assign Public IP (Assegna automaticamente IP pubblico) sia impostato su Enable (Abilita). In caso contrario, imposta il campo su Disable (Disabilita).

Note

Le istanze di container richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite istanze di container con indirizzi IP pubblici.

Per ulteriori informazioni sugli endpoint VPC di interfaccia, vedi [Endpoint VPC dell'interfaccia di Amazon ECS \(AWS PrivateLink\)](#)

Se non disponi di un endpoint VPC di interfaccia configurato e le istanze di container non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta [NAT gateways](#) (Gateway NAT) nella Guida per l'utente di Amazon VPC e [Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux](#) in questa guida.

- **Firewall (security groups)** (Firewall [gruppi di sicurezza]): utilizza un gruppo di sicurezza per definire le regole del firewall per l'istanza di container. Tali regole specificano quale traffico di rete in entrata viene distribuito sull'istanza di container. Tutto il traffico rimanente verrà ignorato.
- Per selezionare un gruppo di sicurezza esistente, scegli **Select an existing security group** (Seleziona un gruppo di sicurezza esistente), quindi seleziona il gruppo di sicurezza creato in [Configurazione per l'uso di Amazon ECS](#)

Per configurare l'archiviazione

L'AMI selezionata include uno o più volumi di archiviazione, compreso il volume dispositivo root. È possibile specificare altri volumi da collegare all'istanza.

Puoi utilizzare la vista Semplice.

- **Storage Type** (Tipo di storage): configura lo storage per l'istanza di container.

Se utilizzi l'AMI Amazon Linux ottimizzata per Amazon ECS, l'istanza ha due volumi configurati. Il volume Root è per il sistema operativo, mentre il secondo volume di Amazon EBS (collegato a `/dev/xvdcz`) è per l'utilizzo di Docker.

Puoi aumentare o diminuire le dimensioni del volume per l'istanza in modo che soddisfi le esigenze applicative.

Dettagli avanzati

Per Advanced Details (Dettagli avanzati), espandi la sezione per visualizzare i campi e specifica eventuali parametri aggiuntivi per l'istanza.

- Purchasing option (Opzioni di acquisto): seleziona Request Spot Instances (Richiedi istanze Spot) per avviare un'istanza Spot. Dovrai anche impostare altri campi correlati alle istanze Spot. Per ulteriori informazioni, consulta [Richiesta Istanza Spot](#).

Note

Se utilizzi istanze Spot e visualizzi un messaggio Not available, potresti dover scegliere un altro tipo di istanza.

- Profilo istanza IAM: seleziona il ruolo IAM dell'istanza di container. Questo di solito è chiamato ecsInstanceRole.

Important

Se non avvii l'istanza di container con le autorizzazioni IAM corrette, l'agente Amazon ECS non potrà connettersi al cluster. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

- (Facoltativo) Dati utente: configura l'istanza di container di Amazon ECS con i dati utente, ad esempio le variabili di ambiente dell'agente da [Configurazione dell'agente del container Amazon ECS](#). Gli script dei dati EC2 utente di Amazon vengono eseguiti solo una volta, al primo avvio dell'istanza. Di seguito sono elencati esempi comuni dei dati utente utilizzati per:
 - Di default, l'istanza di container si avvia nel cluster predefinito. Per avviarla in un cluster non predefinito, scegli l'elenco Advanced Details (Dettagli avanzati). Quindi, incolla lo script seguente nel campo Dati utente, sostituendolo *your_cluster_name* con il nome del cluster.

La EnableTaskIAMRole attiva la funzionalità ruoli IAM dei processi per i processi.

Inoltre, le seguenti opzioni sono disponibili quando si utilizza la modalità di rete `awsvpc`.

- `EnableTaskENI`: questo flag attiva la rete di processi ed è necessario quando si utilizza la modalità di rete `awsvpc`.
- `AwsvpcBlockIMDS`: questo flag facoltativo blocca l'accesso IMDS per i contenitori di processi in esecuzione con la modalità di rete `awsvpc`.
- `AwsvpcAdditionalLocalRoutes`: questo flag facoltativo consente di avere percorsi aggiuntivi nello spazio dei nomi dei processi.

Sostituisci `ip-address` con l'indirizzo IP per le route aggiuntive, ad esempio `172.31.42.23/32`.

```
<powershell>
Import-Module ECSTools
Initialize-ECSAgent -Cluster your_cluster_name -EnableTaskIAMRole -EnableTaskENI -
AwsvpcBlockIMDS -AwsvpcAdditionalLocalRoutes
'["ip-address"]'
</powershell>
```

Avvio delle istanze di container Amazon ECS Windows per il trasferimento di dati

Quando avvii un' EC2 istanza Amazon, puoi passare i dati utente all' EC2 istanza. I dati possono essere utilizzati per eseguire attività di configurazione automatizzate di routine e anche per l'esecuzione di script all'avvio dell'istanza. Per Amazon ECS, i casi di utilizzo più comuni per i dati utente riguardano la trasmissione di informazioni sulla configurazione al daemon Docker e all'agente del container Amazon ECS.

Puoi trasferire diversi tipi di dati utente ad Amazon EC2, inclusi `boothook` cloud, script di shell e direttive `cloud-init`. Per ulteriori informazioni su questi e altri tipi di formato, consulta la [documentazione su cloud-init](#).

Puoi trasmettere questi dati utente quando usi la procedura guidata di EC2 avvio di Amazon. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Data utente di Windows di default

Questo script di dati utente di esempio mostra i dati utente predefiniti che le istanze di container Windows ricevono se si utilizza la console. Lo script sottostante esegue le seguenti operazioni:

- Imposta il nome del cluster in base al nome inserito.

- Imposta i ruoli IAM per i processi.
- Imposta json-file e awslogs come i driver di log disponibili.

Inoltre, le seguenti opzioni sono disponibili quando si utilizza la modalità di rete awsvpc.

- EnableTaskENI: questo flag attiva la rete di processi ed è necessario quando si utilizza la modalità di rete awsvpc.
- AwsvpcBlockIMDS: questo flag facoltativo blocca l'accesso IMDS per i contenitori di processi in esecuzione con la modalità di rete awsvpc.
- AwsvpcAdditionalLocalRoutes: questo flag facoltativo consente di avere route aggiuntive.

Sostituisci ip-address con l'indirizzo IP per le route aggiuntive, ad esempio 172.31.42.23/32.

Puoi usare questo script per le istanze di container, purché siano avviate da un'AMI Windows Server ottimizzata per Amazon ECS.

Sostituisci la riga -Cluster *cluster-name* per specificare il nome del cluster.

```
<powershell>
Initialize-ECSAgent -Cluster cluster-name -EnableTaskIAMRole -LoggingDrivers '["json-
file","awslogs"]' -EnableTaskENI -AwsvpcBlockIMDS -AwsvpcAdditionalLocalRoutes
'["ip-address"]'
</powershell>
```

Per le attività di Windows configurate per utilizzare il driver di registrazione awslogs, è necessario impostare anche la variabile di ambiente ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE nell'istanza di container. Utilizzare la seguente sintassi.

Sostituisci la riga -Cluster *cluster-name* per specificare il nome del cluster.

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE",
$TRUE, "Machine")
Initialize-ECSAgent -Cluster cluster-name -EnableTaskIAMRole -LoggingDrivers '["json-
file","awslogs"]'
</powershell>
```

Dati utente dell'installazione dell'agente Windows

In questo esempio lo script dei dati utente installa l'agente del container di Amazon ECS su un'istanza avviata con un'AMI Windows_Server-2016-English-Full-Containers. È stato adattato dalle istruzioni di installazione dell'agente nella pagina README del [GitHub repository Amazon ECS Container Agent](#).

Note

Questo script è condiviso a scopo esemplificativo. È molto più facile iniziare a utilizzare i container di Windows con l'AMI Windows Server ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta [Creazione di un cluster Amazon ECS per il tipo di lancio Fargate](#).

Per informazioni su come installare l'agente Amazon ECS su Windows Server 2022 Full, consulta il [numero 3753](#) su GitHub

Puoi usare questo script per le tue istanze di container (purché siano avviate con una versione dell'AMI Windows_Server-2016-English-Full-Containers). Assicurati di sostituire la riga *windows* per specificare il nome del tuo cluster (se non utilizzi un cluster denominato windows).

```
<powershell>
# Set up directories the agent uses
New-Item -Type directory -Path ${env:ProgramFiles}\Amazon\ECS -Force
New-Item -Type directory -Path ${env:ProgramData}\Amazon\ECS -Force
New-Item -Type directory -Path ${env:ProgramData}\Amazon\ECS\data -Force
# Set up configuration
$ecsExeDir = "${env:ProgramFiles}\Amazon\ECS"
[Environment]::SetEnvironmentVariable("ECS_CLUSTER", "windows", "Machine")
[Environment]::SetEnvironmentVariable("ECS_LOGFILE", "${env:ProgramData}\Amazon\ECS\log\ecs-agent.log", "Machine")
[Environment]::SetEnvironmentVariable("ECS_DATADIR", "${env:ProgramData}\Amazon\ECS\data", "Machine")
# Download the agent
$agentVersion = "latest"
$agentZipUri = "https://s3.amazonaws.com/amazon-ecs-agent/ecs-agent-windows-$agentVersion.zip"
$zipFile = "${env:TEMP}\ecs-agent.zip"
Invoke-RestMethod -OutFile $zipFile -Uri $agentZipUri
# Put the executables in the executable directory.
Expand-Archive -Path $zipFile -DestinationPath $ecsExeDir -Force
Set-Location ${ecsExeDir}
```

```
# Set $EnableTaskIAMRoles to $true to enable task IAM roles
# Note that enabling IAM roles will make port 80 unavailable for tasks.
[bool]$EnableTaskIAMRoles = $false
if (${$EnableTaskIAMRoles}) {
    $HostSetupScript = Invoke-WebRequest https://raw.githubusercontent.com/aws/amazon-ecs-agent/master/misc/windows-deploy/hostsetup.ps1
    Invoke-Expression $($HostSetupScript.Content)
}
# Install the agent service
New-Service -Name "AmazonECS" `
    -BinaryPathName "$ecsExeDir\amazon-ecs-agent.exe -windows-service" `
    -DisplayName "Amazon ECS" `
    -Description "Amazon ECS service runs the Amazon ECS agent" `
    -DependsOn Docker `
    -StartupType Manual
sc.exe failure AmazonECS reset=300 actions=restart/5000/restart/30000/restart/60000
sc.exe failureflag AmazonECS 1
Start-Service AmazonECS
</powershell>
```

Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Windows

Puoi configurare le tue istanze di container Amazon ECS per l'utilizzo di un proxy HTTP sia per l'agente del container Amazon ECS che per il daemon Docker. Ciò è utile se le tue istanze di container non hanno accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

Per configurare l'istanza del contenitore Amazon ECS Windows per l'utilizzo di un proxy HTTP, imposta le seguenti variabili al momento dell'avvio (con i dati EC2 utente Amazon).

```
[Environment]::SetEnvironmentVariable("HTTP_PROXY",
"http://proxy.mydomain:port", "Machine")
```

Imposta HTTP_PROXY sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per l'agente Amazon ECS per connettersi a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

```
[Environment]::SetEnvironmentVariable("NO_PROXY",  
"169.254.169.254,169.254.170.2,\\.\pipe\docker_engine", "Machine")
```

Imposta su NO_PROXY per 169.254.169.254,169.254.170.2,\\.\pipe\docker_engine filtrare i metadati dell' EC2 istanza, i ruoli IAM per le attività e il traffico del demone Docker dal proxy.

Example Script di dati utente per il proxy HTTP Windows

PowerShell Lo script di dati utente di esempio riportato di seguito configura l'agente contenitore Amazon ECS e il daemon Docker per utilizzare un proxy HTTP specificato dall'utente. Puoi anche specificare un cluster in cui l'istanza di container si registra automaticamente.

Per utilizzare questo script all'avvio di un'istanza di container, segui la procedura descritta in [the section called “Avvio di un'istanza di container”](#). Basta copiare e incollare lo PowerShell script seguente nel campo Dati utente (assicurati di sostituire i valori di esempio rossi con le informazioni sul tuo proxy e cluster).

Note

L'opzione `-EnableTaskIAMRole` è obbligatoria per l'abilitazione dei ruoli IAM per i processi. Per ulteriori informazioni, consulta [Configurazione aggiuntiva dell'istanza Amazon EC2 Windows](#).

```
<powershell>  
Import-Module ECSTools  
  
$proxy = "http://proxy.mydomain:port"  
[Environment]::SetEnvironmentVariable("HTTP_PROXY", $proxy, "Machine")  
[Environment]::SetEnvironmentVariable("NO_PROXY", "169.254.169.254,169.254.170.2,\\.  
\pipe\docker_engine", "Machine")  
  
Restart-Service Docker  
Initialize-ECSAgent -Cluster MyCluster -EnableTaskIAMRole  
</powershell>
```

Configurazione delle istanze di container Amazon ECS Windows per ricevere avvisi sulle istanze Spot

Amazon EC2 chiude, interrompe o iberna l'istanza Spot quando il prezzo Spot supera il prezzo massimo per la richiesta o la capacità non è più disponibile. Amazon EC2 fornisce un avviso di interruzione dell'istanza Spot, che invia all'istanza un avviso di due minuti prima che venga interrotta. Se la funzione di svuotamento dell'istanza Spot di Amazon ECS è abilitata sull'istanza, ECS riceve l'avviso di interruzione dell'istanza Spot e posiziona l'istanza nello stato DRAINING.

Important

Amazon ECS monitora gli avvisi di interruzione dell'istanza Spot con le operazioni di istanza `terminate` e `stop`. Se hai specificato il comportamento di interruzione dell'istanza `hibernate` durante la richiesta di istanze Spot o del parco istanze Spot, la funzione di svuotamento delle istanze Spot di Amazon ECS non è supportata per tali istanze.

Quando un'istanza di container è impostata su DRAINING, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container. Le attività di servizio nell'istanza di container di esaurimento che sono in stato PENDING vengono interrotte immediatamente. Se nel cluster sono disponibili istanze di container, le attività del servizio di sostituzione vengono avviate su di esse.

Puoi attivare il drenaggio delle istanze Spot all'avvio di un'istanza. È necessario impostare il parametro `ECS_ENABLE_SPOT_INSTANCE_DRAINING` prima di avviare l'agente container. Sostituisci *my-cluster* con il nome del cluster.

```
[Environment]::SetEnvironmentVariable("ECS_ENABLE_SPOT_INSTANCE_DRAINING", "true",  
    "Machine")
```

```
# Initialize the agent  
Initialize-ECSAgent -Cluster my-cluster
```

Per ulteriori informazioni, consulta [the section called “Avvio di un'istanza di container”](#).

Cluster Amazon ECS per il tipo di lancio esterno

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premise o una macchina virtuale (VM) nel cluster Amazon ECS. Le istanze esterne

sono ottimizzate per l'esecuzione di applicazioni che generano traffico in uscita o dati di processo. Se l'applicazione richiede traffico in entrata, la mancanza di supporto Elastic Load Balancing rende l'esecuzione di questi carichi di lavoro meno efficiente. Amazon ECS ha aggiunto un nuovo tipo di avvio EXTERNAL che è possibile utilizzare per creare servizi o eseguire processi sulle istanze esterne.

Sistemi operativi e architetture di sistema supportati

Di seguito è riportato l'elenco dei sistemi operativi supportati e delle architetture di sistema.

- Amazon Linux 2
- Amazon Linux 2023
- CentOS Stream 9
- RHEL 7, RHEL 8, RHEL 9: né Docker né i repository di pacchetti aperti di RHEL supportano l'installazione nativa di Docker su RHEL. Prima di eseguire lo script di installazione descritto in questo documento assicurati che Docker sia installato.
- Fedora 32, Fedora 33, Fedora 40
- openSUSE Tumbleweed
- Ubuntu 18, Ubuntu 20, Ubuntu 22, Ubuntu 24
- Debian 10

Important

Il supporto a lungo termine di Debian 9 (supporto LTS) è terminato il 30 giugno 2022 e non è più supportato da Amazon ECS Anywhere.

- Debian 11
- Debian 12
- SUSE Enterprise Server 15
- Le architetture CPU x86_64 e ARM64 sono supportate.
- Sono supportate le seguenti versioni dei sistemi operativi Windows:
 - Windows Server 2022
 - Windows Server 2019
 - Windows Server 2016

- Windows Server 20H2

Considerazioni

Prima di iniziare a utilizzare le istanze esterne, tieni presente le considerazioni riportate di seguito.

- Puoi registrare un'istanza esterna in un cluster alla volta. Per le istruzioni su come registrare un'istanza esterna con un cluster diverso, consulta [Annullamento della registrazione di un'istanza esterna Amazon ECS](#).
- Le istanze esterne richiedono un ruolo IAM con cui comunicare. AWS APIs Per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#).
- Le istanze esterne non devono avere una catena di credenziali di istanze preconfigurata definita in locale in quanto ciò interferirà con lo script di registrazione.
- Per inviare i log dei container a CloudWatch Logs, assicurati di creare e specificare un ruolo IAM per l'esecuzione delle attività nella definizione dell'attività.
- Quando un'istanza esterna viene registrata in un cluster, l'attributo `ecs.capability.external` è associato all'istanza. Questo attributo identificherà l'istanza come istanza esterna. Puoi aggiungere attributi personalizzati alle istanze esterne da utilizzare come vincolo di posizionamento dei processi. Per ulteriori informazioni, consulta [Attributi personalizzati](#).
- È possibile aggiungere tag di risorsa all'istanza esterna. Per ulteriori informazioni, consulta [Istanze di container esterni](#).
- ECS Exec è supportato sulle istanze esterne. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS con ECS Exec](#).
- Di seguito sono riportate considerazioni aggiuntive specifiche per la rete con le istanze esterne. Per ulteriori informazioni, consulta [Rete](#).
 - Il bilanciamento del carico dei servizi non è supportato.
 - L'individuazione dei servizi non è supportata.
 - I processi eseguiti su istanze esterne devono utilizzare le modalità di rete `bridge`, `host` oppure `none`. La modalità di rete `awsvpc` non è supportata.
 - Esistono domini di servizio Amazon ECS in ogni AWS regione. Questi domini di servizio devono essere autorizzati a inviare traffico alle istanze esterne.
 - SSM Agent installato nell'istanza esterna mantiene le credenziali IAM che vengono ruotate ogni 30 minuti utilizzando un'impronta digitale hardware. Se l'istanza esterna perde la connessione a AWS, l'agente SSM aggiorna automaticamente le credenziali dopo aver ristabilito la

connessione. Per ulteriori informazioni, consulta [Convalida di server on-premise e macchine virtuali con un'impronta digitale hardware](#) nella Guida per l'utente di AWS Systems Manager .

- L'API `UpdateContainerAgent` non è supportata. Per istruzioni su come aggiornare SSM Agent o l'agente Amazon ECS nelle istanze esterne, consulta [Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS su un'istanza esterna](#).
- I provider di capacità Amazon ECS non sono supportati. Per creare un servizio o eseguire un processo autonomo nelle istanze esterne, utilizza il tipo di avvio `EXTERNAL`.
- SELinux non è supportato.
- L'uso di volumi Amazon EFS o la specifica di un `EFSVolumeConfiguration` non sono supportati.
- L'integrazione con App Mesh non è supportata.
- Se utilizzi la console per creare una definizione delle attività di istanza esterna, devi creare la definizione delle attività con l'editor JSON della console.
- Quando esegui ECS Anywhere su Windows, devi utilizzare la tua licenza Windows sull'infrastruttura on-premise.
- Quando utilizzi un'AMI non ottimizzata per Amazon ECS, esegui i seguenti comandi sull'istanza di container esterna per configurare le regole per utilizzare i ruoli IAM per le attività. Per ulteriori informazioni, consulta [Configurazione aggiuntiva dell'istanza esterna](#).

```
$ sysctl -w net.ipv4.conf.all.route_localnet=1
$ iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-destination 127.0.0.1:51679
$ iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 51679
```

Rete

Le istanze esterne di Amazon ECS sono ottimizzate per l'esecuzione di applicazioni che generano traffico in uscita o dati di processo. Se l'applicazione richiede traffico in entrata, ad esempio un servizio Web, la mancanza di supporto per Elastic Load Balancing rende l'esecuzione di questi carichi di lavoro meno efficiente perché non è disponibile il supporto per il posizionamento di questi carichi di lavoro dietro un load balancer.

Di seguito sono riportate considerazioni aggiuntive specifiche per la rete con le istanze esterne.

- Il bilanciamento del carico dei servizi non è supportato.

- L'individuazione dei servizi non è supportata.
- I processi Linux eseguiti su istanze esterne devono utilizzare le modalità di rete bridge, host oppure none. La modalità di rete awsvpc non è supportata.

Per ulteriori informazioni su ciascuna modalità di rete, consulta le [opzioni di task networking di Amazon ECS per il tipo di EC2 avvio](#).

- I processi Windows eseguiti su istanze esterne devono utilizzare le modalità di rete default.
- Esistono domini del servizio Amazon ECS in ogni regione e devi disporre dell'autorizzazione per inviare traffico alle istanze esterne.
- SSM Agent installato nell'istanza esterna mantiene le credenziali IAM che vengono ruotate ogni 30 minuti utilizzando un'impronta digitale hardware. Se l'istanza esterna perde la connessione a AWS, l'agente SSM aggiorna automaticamente le credenziali dopo che la connessione è stata ristabilita. Per ulteriori informazioni, consulta [Convalida di server on-premise e macchine virtuali con un'impronta digitale hardware](#) nella Guida per l'utente di AWS Systems Manager .

I seguenti domini vengono utilizzati per la comunicazione tra il servizio Amazon ECS e l'agente Amazon ECS installato nell'istanza esterna. Assicurati che il traffico sia consentito e che la risoluzione DNS funzioni. Per ogni endpoint, *region* rappresenta l'identificatore della regione per una AWS regione supportata da Amazon ECS, ad esempio us-east-2 per la regione Stati Uniti orientali (Ohio). Gli endpoint per tutte le regioni utilizzate devono essere consentiti. Per gli endpoint ecs-a e ecs-t, è necessario includere un asterisco (ad esempio, ecs-a-*).

- ecs-a-*.*region*.amazonaws.com: questo endpoint viene utilizzato per la gestione dei processi.
- ecs-t-*.*region*.amazonaws.com: questo endpoint viene utilizzato per gestire i parametri di processi e container.
- ecs.*region*.amazonaws.com: questo è l'endpoint del servizio per Amazon ECS.
- ssm.*region*.amazonaws.com — Questo è l'endpoint del servizio per AWS Systems Manager
- ec2messages.*region*.amazonaws.com— Questo è l'endpoint di servizio AWS Systems Manager utilizzato per comunicare tra l'agente Systems Manager e il servizio Systems Manager nel cloud.
- ssmmessages.*region*.amazonaws.com: questo endpoint del servizio necessario per creare ed eliminare canali di sessione con il servizio Session Manager nel cloud.
- Se le tue attività richiedono la comunicazione con altri AWS servizi, assicurati che tali endpoint di servizio siano consentiti. Le applicazioni di esempio includono l'uso di Amazon ECR per estrarre

le immagini dei contenitori o l'utilizzo CloudWatch per i CloudWatch log. Per ulteriori informazioni, consulta [Endpoint e quote](#) in Riferimenti generali di AWS .

Amazon FSx for Windows File Server con ECS Anywhere

Per utilizzarlo Amazon FSx for Windows File Server con le istanze esterne di Amazon ECS, devi stabilire una connessione tra il tuo data center locale e il Cloud AWS. Per maggiori informazioni sulle opzioni per connettere la rete al VPC, consulta [Opzioni di connettività di Amazon Virtual Private Cloud](#).

gMSA con ECS Anywhere

Sono supportati i seguenti casi d'uso per ECS Anywhere.

- Active Directory è in Cloud AWS : per questa configurazione, crei una connessione tra la rete locale e il Cloud AWS utilizzando una connessione AWS Direct Connect. Per informazioni su come creare la connessione, consulta [Opzioni di connettività di Amazon Virtual Private Cloud](#). Crei un Active Directory nel Cloud AWS. Per informazioni su come iniziare AWS Directory Service, consulta [Configurazione AWS Directory Service](#) nella Guida all'AWS Directory Service amministrazione. È quindi possibile aggiungere le istanze esterne al dominio utilizzando la AWS Direct Connect connessione. Per informazioni sull'utilizzo di GMSA con Amazon ECS, consulta [the section called “Scopri come usare g MSAs per i contenitori EC2 Windows”](#).
- Active Directory si trova nel data center on-premise. - Per questa configurazione, unisci le istanze esterne all'Active Directory on-premise. Quindi utilizzi le credenziali disponibili localmente quando esegui i processi di Amazon ECS.

Creazione di un cluster Amazon ECS per il tipo di avvio esterno

Crei un cluster per definire l'infrastruttura su cui vengono eseguiti i tuoi compiti e i tuoi servizi.

Prima di iniziare, accertati di aver completato le fasi in [Configurazione per l'uso di Amazon ECS](#) e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta [the section called “Esempi di cluster Amazon ECS”](#). La console Amazon ECS offre un modo semplice per creare le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

Per rendere il processo di creazione del cluster il più semplice possibile, la console dispone di selezioni predefinite per molte scelte che descriviamo di seguito. Ci sono anche pannelli di aiuto disponibili per la maggior parte delle sezioni della console che forniscono ulteriore contesto.

È possibile modificare le seguenti opzioni:

- Aggiungi uno spazio dei nomi al cluster.

Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnetti i servizi Amazon ECS](#).

- Configurazione del cluster per le istanze esterne
- Attiva Container Insights con una migliore osservabilità o Container Insights.

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta [the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata”](#).

Il 2 dicembre 2024, AWS ha rilasciato Container Insights con osservabilità migliorata per Amazon ECS. Questa versione supporta l'osservabilità migliorata per i cluster Amazon ECS utilizzando i tipi di lancio Amazon e EC2 Fargate. Dopo aver configurato Container Insights con una migliore osservabilità su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura dal livello del cluster fino al livello del contenitore nel tuo ambiente e visualizza i tuoi dati in dashboard che mostrano una varietà di metriche e dimensioni. Puoi quindi utilizzare questi out-of-the-box dashboard sulla console Container Insights per comprendere meglio lo stato e le prestazioni dei container e mitigare i problemi più rapidamente identificando le anomalie.

Ti consigliamo di utilizzare Container Insights con una migliore osservabilità anziché Container Insights perché offre una visibilità dettagliata nell'ambiente dei container, riducendo il tempo medio di risoluzione.

- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Creare una chiave KMS nella Guida per](#) l'AWS Key Management Service utente.
- Aggiungi tag per facilitare l'identificazione del cluster.

Creazione di un nuovo cluster (console Amazon ECS)

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Seleziona la Regione da utilizzare nella barra di navigazione.

3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Clusters (Cluster), scegli Create cluster (Crea cluster).
5. In Configurazione del cluster, configura gli elementi seguenti:

- In Nome cluster, inserisci un nome univoco.

Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.

- (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. (Facoltativo) Usa Container Insights, espandi Monitoring, quindi scegli una delle seguenti opzioni:
 - Per utilizzare la soluzione consigliata di Container Insights con osservabilità migliorata, scegli Container Insights con osservabilità migliorata.
 - Per utilizzare Container Insights, scegli Container Insights.
 7. (Facoltativo) Crittografa i dati sullo storage gestito. In Crittografia, per Storage gestito, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati di storage gestito.
 8. (Facoltativo) Per identificare il tuo cluster, espandi la sezione Tags (Tag), quindi configura i tag.

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
 9. Scegli Create (Crea).

Passaggi successivi

È necessario registrare le istanze nel cluster. Per ulteriori informazioni, consulta [Registrazione di un'istanza esterna in un cluster Amazon ECS](#).

Crea una definizione di attività per il tipo di avvio esterno. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#)

Esegui le tue applicazioni come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Esecuzione di un'applicazione come attività Amazon ECS](#)

- [Creazione di un servizio Amazon ECS utilizzando la console](#)

Registrazione di un'istanza esterna in un cluster Amazon ECS

Per ogni istanza esterna che si registra con un cluster Amazon ECS, è necessario che sia installato SSM Agent, l'agente del container di Amazon ECS e Docker. Per registrare l'istanza esterna in un cluster Amazon ECS, deve prima essere registrata come istanza AWS Systems Manager gestita. Puoi creare lo script di installazione in pochi clic dalla console Amazon ECS. Lo script di installazione include una chiave di attivazione di Systems Manager e i comandi per installare ciascuno degli agenti richiesti e Docker. Per completare le fasi di installazione e registrazione, è necessario eseguire lo script di installazione sul server on-premise o sulla macchina virtuale.

Note

Prima di registrare l'istanza esterna Linux con il cluster, crea il file `/etc/ecs/ecs.config` sull'istanza esterna e aggiungi i parametri di configurazione dell'agente del container desiderati. Non è possibile eseguire questa operazione dopo aver registrato l'istanza esterna in un cluster. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

AWS Management Console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Sulla pagina Cluster scegli un cluster in cui registrare l'istanza esterna.
5. Nella **name** pagina Cluster:, scegli la scheda Infrastruttura.
6. Alla pagina Register external instances (Registra istanze esterne), completa la procedura seguente.
 - a. Per Durata della chiave di attivazione (in giorni) specifica il numero di giorni per cui la chiave di attivazione rimane attiva. Una volta passati i giorni specificati, la chiave non funzionerà più quando si registra un'istanza esterna.
 - b. Per Numero di istanze specifica il numero di istanze esterne che si desidera registrare nel cluster con la chiave di attivazione.

- c. Per Ruolo dell'istanza, scegli il ruolo IAM da associare alle istanze esterne. Se un ruolo non è già stato creato, scegli Crea nuovo ruolo perché Amazon ECS crei un ruolo per tuo conto. Per ulteriori informazioni sulle autorizzazioni IAM necessarie per le istanze esterne, consulta [Ruolo IAM di Amazon ECS Anywhere](#).
- d. Copia il comando di registrazione. Questo comando deve essere eseguito su ogni istanza esterna che si desidera registrare nel cluster.

 Important

La parte bash dello script deve essere eseguita come root. Se il comando non viene eseguito come root, viene restituito un errore.

- e. Scegli Chiudi.

AWS CLI for Linux operating systems

1. Crea una coppia di attivazione di Systems Manager. Viene utilizzato per l'attivazione dell'istanza gestita di Systems Manager. L'output include un `ActivationId` e un `ActivationCode`. Utilizzerai questa funzione in una fase successiva. Assicurati di specificare il ruolo IAM ECS Anywhere creato. Per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#).

```
aws ssm create-activation --iam-role ecsAnywhereRole | tee ssm-activation.json
```

2. Scarica lo script di installazione sul server on-premise o nella macchina virtuale (VM).

```
curl --proto "https" -o "/tmp/ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh"
```

3. (Facoltativo) Sul server on-premise o nella macchina virtuale (VM), completa la procedura seguente per verificare lo script di installazione utilizzando il file di firma dello script.
 - a. Scarica e installa GnuPG. Per maggiori informazioni GNUpG, consulta il sito web di [GnuPG](#). Per i sistemi Linux, installa gpg usando il programma di gestione dei pacchetti che preferisci per Linux.
 - b. Recupera la chiave pubblica PGP di Amazon ECS.

```
gpg --keyserver hkp://keys.gnupg.net:80 --recv BCE9D9A42D51784F
```

- c. Scarica la firma dello script di installazione. La firma è una firma PGP scollegata da ASCII archiviata in file con estensione .asc.

```
curl --proto "https" -o "/tmp/ecs-anywhere-install.sh.asc" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh.asc"
```

- d. Verifica il file dello script di installazione utilizzando la chiave.

```
gpg --verify /tmp/ecs-anywhere-install.sh.asc /tmp/ecs-anywhere-install.sh
```

L'output previsto è il seguente:

```
gpg: Signature made Tue 25 May 2021 07:16:29 PM UTC
gpg:                using RSA key 50DECCC4710E61AF
gpg: Good signature from "Amazon ECS <ecs-security@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the
gpg:                owner.
Primary key fingerprint: F34C 3DDA E729 26B0 79BE  AEC6 BCE9 D9A4 2D51 784F
Subkey fingerprint: D64B B6F9 0CF3 77E9 B5FB  346F 50DE CCC4 710E 61AF
```

4. Esegui lo script di installazione sul server on-premise o nella macchina virtuale (VM). Specifica il nome del cluster, la regione, l'ID di attivazione e il codice di attivazione di Systems Manager dalla prima fase.

```
sudo bash /tmp/ecs-anywhere-install.sh \
  --region $REGION \
  --cluster $CLUSTER_NAME \
  --activation-id $ACTIVATION_ID \
  --activation-code $ACTIVATION_CODE
```

Per un server on-premise o una macchina virtuale (VM) con il driver NVIDIA installato per i carichi di lavoro della GPU, è necessario aggiungere il flag `--enable-gpu` allo script di installazione. Quando viene specificato questo flag, lo script di installazione verifica che il driver NVIDIA sia in esecuzione e quindi aggiunge le variabili di configurazione necessarie per eseguire i processi Amazon ECS. Per ulteriori informazioni sull'esecuzione dei carichi

di lavoro della GPU e sulla definizione dei requisiti della GPU in una definizione di attività, consulta [Specificazione GPU in una definizione di attività Amazon ECS](#).

```
sudo bash /tmp/ecs-anywhere-install.sh \  
  --region $REGION \  
  --cluster $CLUSTER_NAME \  
  --activation-id $ACTIVATION_ID \  
  --activation-code $ACTIVATION_CODE \  
  --enable-gpu
```

Utilizza la seguente procedura per registrare un'istanza esterna esistente con un cluster diverso.

Come registrare un'istanza esterna esistente con un cluster diverso

1. Arresta l'agente del container di Amazon ECS.

```
sudo systemctl stop ecs.service
```

2. Modifica il file `/etc/ecs/ecs.config` e alla riga `ECS_CLUSTER`, assicurati che il nome del cluster corrisponda al nome del cluster con cui registrare l'istanza esterna.
3. Rimuovi i dati esistenti dell'agente Amazon ECS.

```
sudo rm /var/lib/ecs/data/agent.db
```

4. Avvia l'agente del container di Amazon ECS.

```
sudo systemctl start ecs.service
```

AWS CLI for Windows operating systems

1. Crea una coppia di attivazione di Systems Manager. Viene utilizzato per l'attivazione dell'istanza gestita di Systems Manager. L'output include un `ActivationId` e un `ActivationCode`. Utilizzerai questa funzione in una fase successiva. Assicurati di specificare il ruolo IAM ECS Anywhere creato. Per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#).

```
aws ssm create-activation --iam-role ecsAnywhereRole | tee ssm-activation.json
```

2. Scarica lo script di installazione sul server on-premise o nella macchina virtuale (VM).

```
Invoke-RestMethod -URI "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install.ps1" -OutFile "ecs-anywhere-install.ps1"
```

3. (Facoltativo) Lo script Powershell è firmato da Amazon e pertanto Windows esegue automaticamente la convalida del certificato sullo stesso. Non è necessario eseguire alcuna convalida manuale.

Per verificare manualmente il certificato, fai clic con il pulsante destro del mouse sul file, accedi alle proprietà e utilizza la scheda Digital Signatures (Firme digitali) per ottenere ulteriori dettagli.

Questa opzione è disponibile solo quando l'host ha il certificato nell'archivio dei certificati.

La verifica dovrebbe restituire informazioni simili alle seguenti:

```
# Verification (PowerShell)
Get-AuthenticodeSignature -FilePath .\ecs-anywhere-install.ps1

SignerCertificate          Status      Path
-----
EXAMPLECERTIFICATE       Valid      ecs-anywhere-install.ps1

...

Subject                   : CN="Amazon Web Services, Inc.",...
-----
```

4. Esegui lo script di installazione sul server on-premise o nella macchina virtuale (VM). Specifica il nome del cluster, la regione, l'ID di attivazione e il codice di attivazione di Systems Manager dalla prima fase.

```
.\ecs-anywhere-install.ps1 -Region $Region -Cluster $Cluster -
ActivationID $ActivationID -ActivationCode $ActivationCode
```

5. Verifica che l'agente del container Amazon ECS sia in esecuzione.

```
Get-Service AmazonECS
```

```
Status      Name      DisplayName
```

```
-----  ----  -----
Running  AmazonECS      Amazon ECS
```

Utilizza la seguente procedura per registrare un'istanza esterna esistente con un cluster diverso.

Come registrare un'istanza esterna esistente con un cluster diverso

1. Arresta l'agente del container di Amazon ECS.

```
Stop-Service AmazonECS
```

2. Modifica il parametro ECS_CLUSTER in modo che il nome del cluster corrisponda al nome del cluster con cui registrare l'istanza esterna.

```
[Environment]::SetEnvironmentVariable("ECS_CLUSTER", $ECSCluster,
[System.EnvironmentVariableTarget]::Machine)
```

3. Rimuovi i dati esistenti dell'agente Amazon ECS.

```
Remove-Item -Recurse -Force $env:ProgramData\Amazon\ECS\data\*
```

4. Avvia l'agente del container di Amazon ECS.

```
Start-Service AmazonECS
```

AWS CLI Può essere utilizzato per creare un'attivazione di Systems Manager prima di eseguire lo script di installazione per completare il processo di registrazione dell'istanza esterna.

Annullamento della registrazione di un'istanza esterna Amazon ECS

Ti consigliamo di annullare la registrazione dell'istanza sia da Amazon ECS che AWS Systems Manager dopo aver finito con l'istanza. In seguito all'annullamento della registrazione, l'istanza esterna non è più in grado di accettare nuovi processi.

Se hai processi in esecuzione nell'istanza di container al momento dell'annullamento della registrazione, questi processi continueranno a essere in esecuzione fino all'arresto in altri modi. Tuttavia, questi processi sono orfani (non più monitorati o tenuti in conto da Amazon ECS). Se questi processi sull'istanza esterna fanno parte di un servizio Amazon ECS, allora se possibile il pianificatore di servizi avvia un'altra copia del processo su un'istanza differente.

Dopo aver annullato la registrazione dell'istanza, pulisci le risorse rimanenti AWS sull'istanza. È quindi possibile registrarla in un nuovo cluster.

Procedura

AWS Management Console

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Dalla barra di navigazione, scegli la Regione in cui l'istanza esterna è registrata.
3. Nel riquadro di navigazione, scegli Cluster e seleziona il cluster che ospita l'istanza esterna.
4. Nella *name* pagina Cluster:, scegli la scheda Infrastruttura.
5. In Container instances (Istanze di container), seleziona l'ID dell'istanza esterna per la quale annullare la registrazione. Si sarà reindirizzati alla pagina dei dettagli dell'istanza di container.
6. Nella *id* pagina Container Instance:, scegli Annulla registrazione.
7. Esamina il messaggio di annullamento della registrazione. Seleziona Annulla registrazione da AWS Systems Manager per annullare la registrazione dell'istanza esterna come istanza gestita da Systems Manager. Scegli Annulla registrazione.

Note

È possibile annullare la registrazione dell'istanza esterna come istanza gestita da Systems Manager nella console di Systems Manager. Per istruzioni, consulta [Annullamento della registrazione dei nodi gestiti in un ambiente ibrido e multicloud](#) nella Guida per l'utente.AWS Systems Manager

8. Dopo aver annullato la registrazione dell'istanza, pulisci le AWS risorse sul server o sulla macchina virtuale locale.

Sistema operativo	Fasi	
Linux	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza.	

Sistema operativo	Fasi	
	<pre>sudo systemctl stop ecs amazon-ssm- agent</pre> b. Rimuovi i pacchetti Amazon ECS e Systems Manager. Per CentOS 7, CentOS 8 e RHEL 7 <pre>sudo yum remove -y amazon-ecs-init amazon-ssm-agent</pre> Per SUSE Enterprise Server 15 <pre>sudo zypper remove -y amazon-ecs-init amazon-ssm-agent</pre> Per Debian e Ubuntu <pre>sudo apt remove -y amazon-ecs-init amazon-ssm-agent</pre> c. Rimuovi le directory rimanenti. <pre>sudo rm -rf /var/ lib/ecs /etc/ecs / var/lib/amazon/ss m /var/log/ecs / var/log/amazon/ssm</pre>	

Sistema operativo	Fasi	
Windows	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza. Stop-Service AmazonECS Stop-Service AmazonSSMAgent b. Rimuovi il pacchetto Amazon ECS. .\ecs-anywhere-install.ps1 -Uninstall	

AWS CLI

1. Per annullare la registrazione dell'istanza di container, sono necessari l'ID istanza e l'ARN dell'istanza di container. Se non disponi di questi valori, esegui i comandi seguenti

Esegui il comando seguente per ottenere l'ID istanza.

Utilizza l'ID dell'istanza (`instanceID`) per ottenere l'ARN dell'istanza di container (`containerInstanceARN`).

```
instanceId=$(aws ssm describe-instance-information --region "{{ region }}" |
jq ".InstanceInformationList[] |select(.IPAddress==\"{{ IPv4 Address }}\")
| .InstanceId" | tr -d'"')
```

Esegui i comandi seguenti.

Utilizza `containerInstanceArn` come parametro nel comando per annullare la registrazione dell'istanza (`deregister-container-instance`).

```
instances=$(aws ecs list-container-instances --cluster "{{ cluster }}" --region
"{{ region }}" | jq -c '.containerInstanceArns')
containerInstanceArn=$(aws ecs describe-container-instances --cluster
"{{ cluster }}" --region "{{ region }}" --container-instances $instances
| jq ".containerInstances[] | select(.ec2InstanceId==\"{{ instanceId }}\")
| .containerInstanceArn" | tr -d '')
```

2. Esegui il seguente comando per svuotare l'istanza.

```
aws ecs update-container-instances-state --cluster "{{ cluster }}" --region
"{{ region }}" --container-instances "{{ containerInstanceArn }}" --status
DRAINING
```

3. Al termine dell'operazione, esegui il comando seguente per annullare la registrazione dell'istanza.

```
aws ecs deregister-container-instance --cluster "{{ cluster }}" --region
"{{ region }}" --container-instance "{{ containerInstanceArn }}"
```

4. Esegui il comando seguente per rimuovere l'istanza di container da SSM.

```
aws ssm deregister-managed-instance --region "{{ region }}" --instance-id
"{{ instanceId }}"
```

5. Dopo aver annullato la registrazione dell'istanza, pulisci le AWS risorse sul server o sulla macchina virtuale locale.

Sistema operativo	Fasi
Linux	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza. <pre>sudo systemctl stop ecs amazon-ssm- agent</pre>

Sistema operativo	Fasi	
	b. Rimuovi i pacchetti Amazon ECS e Systems Manager. <pre>sudo (yum/apt/zypper) remove amazon-ecs-init amazon-ssm-agent</pre> c. Rimuovi le directory rimanenti. <pre>sudo rm -rf /var/ lib/ecs /etc/ecs / var/lib/amazon/ss m /var/log/ecs / var/log/amazon/ssm</pre>	
Windows	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza. <pre>Stop-Service AmazonECS</pre> <pre>Stop-Service AmazonSSMAgent</pre> b. Rimuovi il pacchetto Amazon ECS. <pre>.\ecs-anywhere-ins tall.ps1 -Uninstal l</pre>	

Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS su un'istanza esterna

Il server o la macchina virtuale locale deve eseguire sia l' AWS Systems Manager agente (agente SSM) che l'agente contenitore Amazon ECS durante l'esecuzione di carichi di lavoro Amazon ECS. AWS rilascia nuove versioni di questi agenti ogni volta che vengono aggiunte o aggiornate funzionalità. Se le istanze esterne utilizzano una versione precedente di entrambi gli agenti, è possibile aggiornarle utilizzando le procedure riportate di seguito.

Aggiornamento di SSM Agent su un'istanza esterna

AWS Systems Manager consiglia di automatizzare il processo di aggiornamento dell'agente SSM sulle istanze. Forniscono diversi metodi per automatizzare gli aggiornamenti. Per ulteriori informazioni, consulta [Automazione degli aggiornamenti a SSM Agent](#) nella Guida per l'utente di AWS Systems Manager .

Aggiornamento dell'agente di Amazon ECS su un'istanza esterna

Nelle istanze esterne, l'agente del container Amazon ECS viene aggiornato aggiornando il pacchetto `ecs-init`. L'aggiornamento dell'agente di Amazon ECS non interrompe i processi o i servizi in esecuzione. Amazon ECS fornisce il pacchetto `ecs-init` e il file di firma in un bucket Amazon S3 in ciascuna regione. A cominciare da `ecs-init` versione 1.52.1-1, Amazon ECS fornisce pacchetti `ecs-init` da utilizzare in base al sistema operativo e all'architettura di sistema utilizzata dall'istanza esterna.

Utilizza la seguente tabella per determinare il pacchetto `ecs-init` da scaricare in base al sistema operativo e all'architettura di sistema utilizzata dall'istanza esterna.

Note

È possibile determinare il sistema operativo e l'architettura di sistema utilizzati dall'istanza esterna utilizzando i comandi riportati di seguito.

```
cat /etc/os-release  
uname -m
```

Sistemi operativi (architettura)	Pacchetto ecs-init
CentOS 7 (x86_64)	amazon-ecs-init-latest.x86_64.rpm
CentOS 8 (x86_64)	
CentOS Stream 9 (x86_64)	
SUSE Enterprise Server 15 (x86_64)	
RHEL 7 (x86_64)	
RHEL 8 (x86_64)	
CentOS 7 (aarch64)	amazon-ecs-init-latest.aarch64.rpm
CentOS 8 (aarch64)	
CentOS Stream 9 (aarch64)	
RHEL 7 (aarch64)	
Debian 9 (x86_64)	amazon-ecs-init-latest.amd64.deb
Debian 10 (x86_64)	
Debian 11 (x86_64)	
Debian 12 (x86_64)	
Ubuntu 18 (x86_64)	
Ubuntu 20 (x86_64)	
Ubuntu 22 (x86_64)	
Ubuntu 24 (x86_64)	
Debian 9 (aarch64)	amazon-ecs-init-latest.arm64.deb
Debian 10 (aarch64)	
Debian 11 (aarch64)	

Sistemi operativi (architettura)	Pacchetto ecs-init
Debian 12 (aarch64)	
Ubuntu 18 (aarch64)	
Ubuntu 20 (aarch64)	
Ubuntu 22 (aarch64)	
Ubuntu 24 (aarch64)	

Completa questa procedura per aggiornare l'agente Amazon ECS.

Come aggiornare l'agente Amazon ECS

1. Conferma la versione dell'agente Amazon ECS in esecuzione.

```
curl -s 127.0.0.1:51678/v1/metadata | python3 -mjson.tool
```

2. Scarica il pacchetto `ecs-init` per il sistema operativo e l'architettura di sistema in uso. Amazon ECS fornisce il file del pacchetto `ecs-init` in un bucket Amazon S3 in ciascuna regione. Assicurati di sostituire l'<region> identificatore nel comando con il nome della regione (ad esempio, `us-west-2`) a cui sei geograficamente più vicino.

amazon-ecs-init-latest.x86_64.rpm

```
curl -o amazon-ecs-init.rpm https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.x86_64.rpm
```

amazon-ecs-init-latest.aarch64 rpm

```
curl -o amazon-ecs-init.rpm https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.aarch64.rpm
```

amazon-ecs-init-latest.amd64.deb

```
curl -o amazon-ecs-init.deb https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.amd64.deb
```

amazon-ecs-init-latest.arm64.deb

```
curl -o amazon-ecs-init.deb https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.arm64.deb
```

3. (Facoltativo) Verifica la validità del file del pacchetto ecs-init utilizzando la firma PGP.
 - a. Scarica e installa GnuPG. Per maggiori informazioni GnuPG, consulta il sito web di [GnuPG](https://gnupg.org/). Per i sistemi Linux, installa gpg usando il programma di gestione dei pacchetti che preferisci per Linux.
 - b. Recupera la chiave pubblica PGP di Amazon ECS.

```
gpg --keyserver hkp://keys.gnupg.net:80 --recv BCE9D9A42D51784F
```

- c. Scarica la firma del pacchetto ecs-init. La firma è una firma PGP scollegata da ASCII archiviata in file con estensione .asc. Amazon ECS fornisce il file di firma in un bucket Amazon S3 in ciascuna regione. Assicurati di sostituire l'<region> identificatore nel comando con il nome della regione (ad esempio, us-west-2) a cui sei geograficamente più vicino.

amazon-ecs-init-latest.x86_64.rpm

```
curl -o amazon-ecs-init.rpm.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.x86_64.rpm.asc
```

amazon-ecs-init-latest.aarch64.rpm

```
curl -o amazon-ecs-init.rpm.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.aarch64.rpm.asc
```

amazon-ecs-init-latest.amd64.deb

```
curl -o amazon-ecs-init.deb.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.amd64.deb.asc
```

amazon-ecs-init-latest.arm64.deb

```
curl -o amazon-ecs-init.deb.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.arm64.deb.asc
```

- d. Verificare il file del pacchetto `ecs-init` usando la chiave.

Per i pacchetti **rpm**

```
gpg --verify amazon-ecs-init.rpm.asc ./amazon-ecs-init.rpm
```

Per i pacchetti **deb**

```
gpg --verify amazon-ecs-init.deb.asc ./amazon-ecs-init.deb
```

L'output previsto è il seguente:

```
gpg: Signature made Fri 14 May 2021 09:31:36 PM UTC
gpg:                using RSA key 50DECCC4710E61AF
gpg: Good signature from "Amazon ECS <ecs-security@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:                There is no indication that the signature belongs to the owner.
Primary key fingerprint: F34C 3DDA E729 26B0 79BE  AEC6 BCE9 D9A4 2D51 784F
Subkey fingerprint: D64B B6F9 0CF3 77E9 B5FB  346F 50DE CCC4 710E 61AF
```

4. Installare il pacchetto `ecs-init`.

Per il pacchetto **rpm** su CentOS 7, CentOS 8 e RHEL 7

```
sudo yum install -y ./amazon-ecs-init.rpm
```

Per il pacchetto **rpm** su SUSE Enterprise Server 15

```
sudo zypper install -y --allow-unsigned-rpm ./amazon-ecs-init.rpm
```

Per il pacchetto **deb**

```
sudo dpkg -i ./amazon-ecs-init.deb
```

5. Riavvia il servizio `ecs`.

```
sudo systemctl restart ecs
```

6. Verifica che la versione dell'agente Amazon ECS sia stata aggiornata.

```
curl -s 127.0.0.1:51678/v1/metadata | python3 -mjson.tool
```

Aggiornamento di un cluster Amazon ECS

Puoi modificare le seguenti proprietà del cluster:

- Imposta un provider di capacità di default

Ogni cluster ha uno o più provider di capacità e una strategia di provider di capacità facoltativa. La strategia del provider di capacità determina il modo in cui le attività vengono distribuite tra i provider di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del provider di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster.

- Attiva Container Insights.

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta [the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata”](#).

- Aggiungi tag per facilitare l'identificazione dei cluster.

Procedura

1. Apri <https://console.aws.amazon.com/ecs/la> console nella versione 2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella *name* pagina Cluster:, scegli Aggiorna cluster.
5. Per impostare il provider di capacità predefinito, in Strategia predefinita del provider di capacità, scegli Aggiungi altro.

- a. Scegli il provider in Provider di capacità.
 - b. (Facoltativo) Per Base, inserisci il numero minimo di attività eseguite sul provider di capacità.

Puoi impostare solo un valore Base per ogni provider di capacità.
 - c. (Facoltativo) Per Peso, inserisci la percentuale relativa al numero totale di attività avviate che utilizzano il provider di capacità specificato.
 - d. (Facoltativo) Ripeti i passaggi per eventuali provider di capacità aggiuntivi.
6. Per attivare o disattivare Container Insights, espandi Monitoraggio e poi attiva Utilizza Container Insights.
 7. Per identificare il tuo cluster, espandi la sezione Tag, quindi configura i tag.

[Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
[Rimuovi un tag] Scegli Rimuovi a destra della Chiave e del Valore del tag.
 8. Scegli Aggiorna.

Eliminazione di un cluster Amazon ECS

Al termine dell'utilizzo di un cluster, puoi eliminarlo. Dopo aver eliminato il cluster, lo stato di quest'ultimo diventa INACTIVE. I cluster con stato INACTIVE potrebbero rimanere individuabili nel tuo account per un periodo di tempo. Tuttavia, questo comportamento è soggetto a modifiche in futuro, quindi non dovresti fare affidamento sulla persistenza dei cluster INACTIVE.

Prima di eliminare un cluster, esegui le operazioni seguenti:

- Elimina tutti i servizi del cluster. Per ulteriori informazioni, consulta [the section called “Eliminazione di un servizio”](#).
- Interrompi tutti i processi attualmente in esecuzione. Per ulteriori informazioni, consulta [the section called “Interruzione di un'attività”](#).
- Annulla la registrazione di tutte le istanze del container registrate nel cluster. Per ulteriori informazioni, consulta [the section called “Annullamento della registrazione di un'istanza di container”](#).

- Elimina lo spazio dei nomi . Per ulteriori informazioni, consulta [Eliminazione degli spazi dei nomi](#) nella Guida per gli sviluppatori di AWS Cloud Map .

Procedura

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel pannello di navigazione scegli Cluster.
4. Nella pagina Cluster, seleziona il cluster da eliminare.
5. In alto a destra della pagina, scegli Elimina Cluster.

Se non sono state eliminate tutte le risorse associate al cluster, viene visualizzato un messaggio.

6. Nella casella di conferma, inserisci delete **cluster name**.

Annullamento della registrazione di un'istanza di container Amazon ECS

Important

Questo argomento riguarda EC2 solo le istanze di container create in Amazon. Per ulteriori informazioni sull'annullamento delle registrazioni delle istanze gestite, consulta [Annullamento della registrazione di un'istanza esterna Amazon ECS](#).

Quando hai finito con un'istanza di container EC2 supportata da Amazon, devi annullarne la registrazione dal cluster. In seguito all'annullamento della registrazione, l'istanza di container non è più in grado di accettare nuove attività.

Se hai attività in esecuzione nell'istanza di container al momento dell'annullamento della registrazione, queste attività continueranno a essere in esecuzione fino al termine dell'istanza o fino a quando le attività non verranno interrotte in altri modi. Tuttavia, questi processi sono orfani (non più monitorati o tenuti in considerazione da Amazon ECS). Se un processo orfano nell'istanza di container è parte di un servizio Amazon ECS, il pianificatore di servizi avvia un'altra copia di tale attività in un'altra istanza di container, se possibile. La registrazione di eventuali container in attività di servizio orfana registrati con un gruppo di destinazione di Application Load Balancer viene annullata. Ha inizio lo svuotamento della connessione in base alle impostazioni nel sistema di bilanciamento del

carico o nel gruppo di destinazione. Se un processo orfano utilizza la modalità di rete `awsvpc`, le sue interfacce di rete elastiche vengono eliminate.

Se hai intenzione di utilizzare l'istanza di container per altri motivi in seguito all'annullamento della registrazione, devi interrompere tutte le attività in esecuzione nell'istanza di container prima dell'annullamento. Questa operazione interrompe il consumo di risorse da parte delle attività orfane.

Quando annulli la registrazione di un'istanza di container, tieni presente le considerazioni riportate di seguito.

- Dato che ogni istanza di container presenta informazioni sullo stato univoche, non deve essere annullata da un cluster e registrata nuovamente in un altro. Per riposizionare le risorse dell'istanza di container, ti consigliamo di terminare le istanze di container da un cluster e avviare nuove istanze di container nel nuovo cluster. Per ulteriori informazioni, consulta [Terminare l'istanza](#) nella Amazon EC2 User Guide e [Avvio di un'istanza di container Linux di Amazon ECS](#).
- Se l'istanza del contenitore è gestita da un gruppo o da uno AWS CloudFormation stack di Auto Scaling, interrompi l'istanza aggiornando il gruppo o lo stack Auto Scaling. AWS CloudFormation In caso contrario, il gruppo con dimensionamento automatico o AWS CloudFormation creerà una nuova istanza dopo averla terminata.
- Se termini un'istanza di container in esecuzione con un agente del container Amazon ECS connesso, l'agente annulla automaticamente la registrazione dell'istanza dal cluster. Non viene automaticamente annullata la registrazione delle istanze di container interrotte o delle istanze con agenti disconnessi al loro termine.
- L'annullamento della registrazione di un'istanza di container rimuove l'istanza da un cluster, ma non interrompe l'istanza Amazon. EC2 Al termine dell'utilizzo dell'istanza, assicurati di terminarla in modo da interrompere la fatturazione. Per ulteriori informazioni, consulta [Terminare l'istanza](#) nella Amazon EC2 User Guide.

Procedura

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Dalla barra di navigazione, scegli la Regione in cui l'istanza esterna è registrata.
3. Nel pannello di navigazione, scegli Clusters (Cluster) e seleziona il cluster che ospita l'istanza.
4. Nella **name** pagina Cluster:, scegli la scheda Infrastruttura.
5. In Container instances (Istanze di container), seleziona l'ID dell'istanza per la quale annullare la registrazione. Si sarà reindirizzati alla pagina dei dettagli dell'istanza di container.

6. Nella **id** pagina Container Instance:, scegli Annulla registrazione.
7. Nella schermata di conferma, seleziona Annulla registrazione.
8. Se hai finito con l'istanza del contenitore, termina l' EC2istanza Amazon sottostante. Per ulteriori informazioni, consulta [Terminate Your Instance](#) nella Amazon EC2 User Guide.

Drenaggio delle istanze di container Amazon ECS

In alcuni casi potrebbe essere necessario rimuovere un'istanza di container dal cluster, ad esempio per eseguire aggiornamenti di sistema o per ridurre la capacità del cluster. Amazon ECS offre la possibilità di passare un'istanza di container a uno stato DRAINING. Questa operazione è nota come svuotamento dell'istanza di container. Quando un'istanza di container è impostata su DRAINING, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container.

Comportamento di svuotamento per i servizi

Qualsiasi processo che fa parte di un servizio che si trova in uno stato PENDING viene arrestato immediatamente. Se nel cluster è disponibile la capacità dell'istanza di container, il pianificatore di servizi avvierà i processi di sostituzione. Se la capacità dell'istanza di container non è sufficiente, verrà inviato un messaggio di evento del servizio che indica il problema.

I processi che fanno parte di un servizio nell'istanza di container che si trovano in uno stato RUNNING passano a uno stato STOPPED. Lo scheduler di servizi prova a sostituire le attività in base al tipo di implementazione e ai parametri di configurazione dell'implementazione del servizio, `minimumHealthyPercent` e `maximumPercent`. Per ulteriori informazioni, consultare [Servizi Amazon ECS](#) e [Parametri di definizione del servizio Amazon ECS](#).

- Se `minimumHealthyPercent` è inferiore al 100%, il pianificatore può ignorare `desiredCount` temporaneamente durante la sostituzione delle attività. Ad esempio, `desiredCount` sono quattro attività, un minimo del 50% permette al pianificatore di interrompere due attività esistenti prima di avviare due nuove attività. Se il minimo è del 100%, il pianificatore del servizio non può rimuovere le attività esistenti fino a quando le attività di sostituzione non vengono considerate integre. Se le attività per i servizi che non utilizzano un load balancer sono in stato RUNNING, vengono considerate integre. Le attività per i servizi che utilizzano un load balancer vengono considerate integre se sono in stato RUNNING e se il load balancer considera integra l'istanza di container su cui sono ospitate.

⚠ Important

Se utilizzi le istanze Spot e `minimumHealthyPercent` è maggiore o uguale al 100%, il servizio non avrà abbastanza tempo per sostituire l'attività prima della cessazione dell'istanza Spot.

- Il parametro `maximumPercent` rappresenta un limite superiore al numero di attività in esecuzione durante la sostituzione delle attività. Ciò permette di definire le dimensioni del batch di sostituzione. Ad esempio, se `desiredCount` di quattro attività, un massimo di 200% avvia quattro nuove attività prima di interrompere le quattro attività affinché vengano esaurite (a condizione che le risorse del cluster necessarie per questa operazione siano disponibili). Se il massimo è 100%, le attività di sostituzione non possono avviarsi fino all'interruzione delle attività di esaurimento.

⚠ Important

Se `minimumHealthyPercent` e `maximumPercent` sono entrambi al 100%, il servizio non può rimuovere i processi esistenti e non può inoltre avviare processi di sostituzione. Ciò impedisce il corretto svuotamento delle istanze del container e impedisce la creazione di nuove implementazioni.

Comportamento di svuotamento per processi autonomi

Qualsiasi processo autonomo nello stato PENDING o RUNNING non ne è influenzato; devi attenderne l'interruzione o interromperlo manualmente. L'istanza di container rimarrà nello DRAINING stato.

Un'istanza di container ha completato lo svuotamento quando tutti i processi in esecuzione sull'istanza passano a uno stato STOPPED. L'istanza di container rimane nello stato DRAINING finché non viene nuovamente attivata o eliminata. Puoi verificare lo stato delle attività sull'istanza del contenitore utilizzando l'[ListTasks](#) operazione con il `containerInstance` parametro per ottenere un elenco di attività sull'istanza seguita da un'[DescribeTasks](#) operazione con l'Amazon Resource Name (ARN) o l'ID di ciascuna attività per verificare lo stato dell'attività.

Quando desideri che l'istanza di container avvi nuovamente i processi di hosting, puoi modificare lo stato dell'istanza di container da DRAINING a ACTIVE. Lo scheduler del servizio Amazon ECS considererà nuovamente l'istanza di container per il posizionamento del processo.

Procedura

Le seguenti fasi possono essere utilizzate per impostare lo svuotamento di un'istanza di container utilizzando la nuova AWS Management Console.

Puoi anche utilizzare l'azione [UpdateContainerInstancesState](#) API o il [update-container-instances-state](#) comando per modificare lo stato di un'istanza del contenitore in DRAINING.

AWS Management Console

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Alla pagina Clusters (Cluster), scegli un cluster che ospita le istanze.
4. Nella *name* pagina Cluster:, scegli la scheda Infrastruttura. Quindi, in Container instances (Istanze di container) e seleziona la casella di controllo per ciascuna istanza di container che desideri svuotare.
5. Scegli Operazioni, Drain.

Agente del container Amazon ECS

L'agente Amazon ECS è un processo che viene eseguito su ogni istanza di container registrata nel cluster. Facilita la comunicazione tra le istanze del contenitore e Amazon ECS.

Note

Nelle istanze di container Linux, il contenitore dell'agente monta directory di primo livello come, e. `/lib /lib64 /proc` Ciò è necessario per caratteristiche e funzionalità ECS come volumi Amazon EBS, modalità di `aws-ipc` rete, Amazon ECS Service Connect e FireLens per Amazon ECS.

Ogni versione dell'agente del container di Amazon ECS supporta una serie di funzioni differenti e assicura le correzioni dei bug delle versioni precedenti. Quando possibile, consigliamo sempre di utilizzare la versione più recente dell'agente del container di Amazon ECS. Per passare all'ultima versione dell'agente del container, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

L'agente contenitore Amazon ECS contiene l'`amazon-ecs-pause` immagine. Amazon ECS utilizza questa immagine per attività che utilizzano la modalità di rete. `aws-ipc`

Per vedere quali funzionalità e miglioramenti sono inclusi in ogni versione dell'agente, consulta / releases. <https://github.com/aws/amazon-ecs-agent>

Important

La versione Docker minima per parametri affidabili è v20.10.13 e successive, inclusa nell'AMI 20220607 ottimizzata per Amazon ECS e versioni successive.

Le versioni dell'agente Amazon ECS 1.20.0 e successive non supportano più le versioni di Docker precedenti alla 18.01.0.

Ciclo di vita

Quando l'agente container Amazon ECS registra EC2 un'istanza Amazon nel tuo cluster, l'istanza Amazon riporta il suo stato ACTIVE e lo stato della connessione dell'agente come TRUE. Questa istanza di container può accettare richieste di esecuzione di attività.

Se interrompi un'istanza di container (senza terminarla), lo stato rimane ACTIVE, ma lo stato di connessione dell'agente passa a FALSE entro pochi minuti. Tutte le attività in esecuzione nell'istanza di container vengono interrotte. Se avvii di nuovo l'istanza di container, l'agente del container si riconnetterà al servizio Amazon ECS e potrai nuovamente eseguire attività nell'istanza.

Important

Se interrompi e avvii un'istanza di container oppure riavvii l'istanza, alcune versioni precedenti dell'agente del container di Amazon ECS registreranno di nuovo l'istanza senza annullare la registrazione dell'ID dell'istanza di container originale. In questo caso, Amazon ECS elenca più istanze di container nel cluster rispetto a quelle effettivamente disponibili. (Se disponi di un'istanza contenitore duplicata IDs per lo stesso ID di EC2 istanza Amazon, puoi annullare in sicurezza la registrazione dei duplicati elencati ACTIVE con uno stato di connessione dell'agente pari a.) FALSE Questo problema è stato risolto nella versione corrente dell'agente del container di Amazon ECS. Per ulteriori informazioni sull'aggiornamento alla versione corrente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Se modifichi lo stato di un'istanza di container in DRAINING, le nuove attività non vengono posizionate nell'istanza di container. Qualsiasi attività di servizio in esecuzione nell'istanza di

container viene rimossa, se possibile, in modo che possano essere eseguiti gli aggiornamenti di sistema. Per ulteriori informazioni, consulta [Drenaggio delle istanze di container Amazon ECS](#).

Se annulli la registrazione di un'istanza di container o la termini, lo stato dell'istanza di container passa immediatamente a INACTIVE e l'istanza di container non viene più segnalata quando elenchi le istanze di container. Tuttavia, puoi ancora descrivere l'istanza di container per un'ora in seguito alla terminazione. Dopo un'ora, la descrizione dell'istanza non è più disponibile.

Important

Puoi svuotare manualmente le istanze o creare un hook del ciclo di vita del gruppo Auto Scaling per impostare lo stato dell'istanza su DRAINING. Per ulteriori informazioni sui ganci del ciclo di [vita di Amazon EC2 Auto Scaling](#), consulta [gli hook del ciclo](#) di vita di Amazon Auto Scaling.

AMI ottimizzate per Amazon ECS

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 come base. Il nome dell'AMI di origine di Amazon Linux 2 per ogni variante può essere recuperato interrogando l'API Archivio dei parametri Systems Manager. Per ulteriori informazioni, consulta [Recupero di metadati AMI Linux ottimizzati per Amazon ECS](#). Quando avvii le istanze di container dall'AMI Amazon Linux 2 ottimizzata per Amazon ECS più recente, ricevi la versione dell'agente del container corrente. Per avviare un'istanza di container con l'AMI Amazon Linux 2 ottimizzata per Amazon ECS, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Informazioni aggiuntive

Nelle pagine seguenti vengono fornite ulteriori informazioni sulle modifiche:

- Registro delle [modifiche di Amazon ECS Agent attivo](#) GitHub
- Il codice sorgente dell'applicazione `ecs-init`, gli script e la configurazione per il pacchetto dell'agente fanno ora parte del repository dell'agente. Per le versioni `ecs-init` e i pacchetti precedenti di Amazon `ecs-init`, consulta il changelog di [Amazon ecs-init](#) su GitHub
- [Note di rilascio di Amazon Linux 2](#).
- [Note di rilascio di Docker Engine](#) nella documentazione Docker
- [Documentazione dei driver NVIDIA](#) nella documentazione NVIDIA

Configurazione dell'agente del container Amazon ECS

Si applica a: istanze EC2

L'agente container Amazon ECS supporta una serie di opzioni di configurazione, la maggior parte delle quali impostate tramite variabili di ambiente.

Se l'istanza di container è stata avviata con una variante Linux dell'AMI ottimizzata per Amazon ECS, puoi impostare queste variabili di ambiente nel file `/etc/ecs/ecs.config` e quindi riavviare l'agente. Puoi anche scrivere queste variabili di configurazione nelle tue istanze di container con i dati EC2 utente di Amazon al momento del lancio. Per ulteriori informazioni, consulta [Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati](#).

Se l'istanza del contenitore è stata avviata con una variante Windows dell'AMI ottimizzata per Amazon ECS, puoi impostare queste variabili di ambiente con il PowerShell `SetEnvironmentVariable` comando e quindi riavviare l'agente. Per ulteriori informazioni, consulta [Esegui comandi all'avvio di un' EC2 istanza con input di dati utente](#) nella Amazon EC2 User Guide e [the section called “Avvio delle istanze dei container”](#).

Se avvii manualmente l'agente container Amazon ECS (per dispositivi non ottimizzati per Amazon ECS AMIs), puoi utilizzare queste variabili di ambiente nel `docker run` comando che usi per avviare l'agente. Utilizza queste variabili con la sintassi `--env=VARIABLE_NAME=VARIABLE_VALUE`. Nel caso di informazioni sensibili, come ad esempio le credenziali di autenticazione a repository privati, le variabili di ambiente dell'agente vanno archiviate in un file e trasmesse tutte in una volta con l'opzione `--env-file path_to_env_file`. Per aggiungere le variabili, puoi utilizzare i seguenti comandi.

```
sudo systemctl stop ecs
sudo vi /etc/ecs/ecs.config
# And add the environment variables with VARIABLE_NAME=VARIABLE_VALUE format.
sudo systemctl start ecs
```

Parametri disponibili

Per informazioni sui parametri di configurazione dell'agente container Amazon ECS disponibili, consulta [Amazon ECS Container Agent on GitHub](#).

Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3

La configurazione dell'agente container Amazon ECS è controllata con la variabile di ambiente. Le varianti Linux dell'AMI ottimizzata per Amazon ECS cercano queste variabili in `/etc/ecs/ecs.config` all'avvio dell'agente del container e lo configurano di conseguenza. Le variabili di ambiente non sensibili, ad esempio `ECS_CLUSTER`, possono essere passate all'istanza del contenitore all'avvio tramite i dati EC2 utente di Amazon e scritte in questo file senza conseguenze. Tuttavia, altre informazioni sensibili, come le AWS credenziali o la `ECS_ENGINE_AUTH_DATA` variabile, non devono mai essere passate a un'istanza nei dati dell'utente o scritte `/etc/ecs/ecs.config` in un modo che consenta loro di essere visualizzate in un file. `.bash_history`

Archiviare le informazioni di configurazione in un bucket privato in Amazon S3 e concedere l'accesso in sola lettura al ruolo IAM dell'istanza di container è un modo sicuro e conveniente per permettere la configurazione delle istanze di container all'avvio. È possibile memorizzare una copia del file `ecs.config` in un bucket privato. Puoi quindi utilizzare i dati EC2 utente di Amazon per installare AWS CLI e copiare le informazioni di configurazione all'`/etc/ecs/ecs.config` all'avvio dell'istanza.

Come memorizzare un file **ecs.config** in Amazon S3

1. È necessario concedere all'istanza del contenitore le autorizzazioni role (`ecsInstanceRole`) per avere accesso in sola lettura ad Amazon S3. Puoi farlo assegnando `AmazonS3` al ruolo `ReadOnlyAccess`. `ecsInstanceRole` Per informazioni su come allegare una politica a un ruolo, consulta [Aggiornare le autorizzazioni per un ruolo nella Guida per l'utente](#) AWS Identity and Access Management
2. Crea un file `ecs.config` con variabili di configurazione dell'agente Amazon ECS valide utilizzando il seguente formato. In questo esempio viene configurata l'autenticazione di un registro privato. Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

```
ECS_ENGINE_AUTH_TYPE=dockercfg
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"auth":"zq212MzEXAMPLE7o6T25Dk0i","email":"email@example.com"}}
```


 Note

Per un elenco completo delle variabili di configurazione dell'agente Amazon ECS disponibili, consulta [Amazon ECS Container Agent on GitHub](#).

3. Per archiviare il file di configurazione, crea un bucket privato in Amazon S3. Per ulteriori informazioni, consulta [Creare un bucket](#) nella Guida per l'utente di Amazon Simple Storage Service.
4. Carica il file `ecs.config` nel bucket S3. Per ulteriori informazioni, consulta [Caricamento degli oggetti](#) nella Guida per l'utente di Amazon Simple Storage Service.

Come caricare un file **ecs.config** da Amazon S3 all'avvio

1. Procedi come indicato in questa sezione per consentire l'accesso Amazon S3 in sola lettura alle istanze di container e archiviare un file `ecs.config` in un bucket privato S3.
2. Avvia nuove istanze di container e usa il seguente script di esempio nei dati EC2 utente. Lo script installa AWS CLI e copia il file di configurazione in `/etc/ecs/ecs.config`. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

```
#!/bin/bash
yum install -y aws-cli
aws s3 cp s3://your_bucket_name/ecs.config /etc/ecs/ecs.config
```

Installazione dell'agente del container Amazon ECS

Se desideri registrare un' EC2 istanza Amazon con il tuo cluster Amazon ECS e quell'istanza non utilizza un'AMI basata sull'AMI ottimizzata per Amazon ECS, puoi installare l'agente contenitore Amazon ECS manualmente utilizzando la seguente procedura. A tale scopo, puoi scaricare l'agente da uno dei bucket Amazon S3 regionali o da Amazon Elastic Container Registry Public. Se scarichi da uno dei bucket regionali di Amazon S3, puoi opzionalmente verificare la validità del file dell'agente contenitore utilizzando la firma PGP.

 Note

Le unità `systemd` per i servizi Amazon ECS e Docker hanno l'istruzione di attendere il completamento di `cloud-init` prima di avviare entrambi i servizi. Il `cloud-init` processo

non è considerato completato fino al termine dell'esecuzione dei dati EC2 utente Amazon. Pertanto, l'avvio di Amazon ECS o Docker tramite i dati EC2 utente di Amazon può causare un deadlock. Puoi usare i dati utente di Amazon per avviare l'agente container utilizzando i dati EC2 utente di Amazon `systemctl enable --now --no-block ecs.service`.

Installazione dell'agente container Amazon ECS su un dispositivo diverso da Amazon Linux EC2 istanza

Per installare l'agente container Amazon ECS su un' EC2 istanza Amazon, puoi scaricare l'agente da uno dei bucket Amazon S3 regionali e installarlo.

Note

Quando si utilizza un'AMI non Amazon Linux, l' EC2 istanza Amazon richiede il `cgroupfs` supporto per il `cgroup` driver affinché l'agente Amazon ECS supporti i limiti di risorse a livello di attività. Per ulteriori informazioni, consulta [Amazon ECS agent on GitHub](#).

A titolo di riferimento, di seguito sono elencati i file più recenti dell'agente del container Amazon ECS, per regione e per ciascun tipo di architettura di sistema.

Regione	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
us-east-2	Stati Uniti orientali (Ohio)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
us-east-1	Stati Uniti orientali (Virginia settentrionale)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)

Regione	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
us-west-1	Stati Uniti occidentali (California settentrionale)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
us-west-2	US West (Oregon)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-east-1	Asia Pacifico (Hong Kong)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-northeast-1	Asia Pacifico (Tokyo)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-northeast-2	Asia Pacifico (Seoul)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-south-1	Asia Pacifico (Mumbai)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)

Regione	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
ap-southeast-1	Asia Pacifico (Singapore)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ap-southeast-2	Asia Pacifico (Sydney)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ca-central-1	Canada (Centrale)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
eu-central-1	Europa (Francoforte)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
eu-west-1	Europa (Irlanda)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
eu-west-2	Europa (Londra)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)

Regione	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
eu-west-3	Europa (Parigi)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
sa-east-1	Sud America (San Paolo)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
us-gov-east-1	AWS GovCloud (Stati Uniti orientali)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
us-gov-west-1	AWS GovCloud (Stati Uniti occidentali)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)

Per installare l'agente container Amazon ECS su un' EC2 istanza Amazon utilizzando un dispositivo non-Amaon Linux AMI

1. Avvia un' EC2 istanza Amazon con un ruolo IAM che consente l'accesso ad Amazon ECS. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).
2. Connettiti alla tua istanza.
3. Installare la versione più recente di Docker nell'istanza.
4. Controlla la versione Docker per verificare che il tuo sistema soddisfi il requisito di versione minima.

 Note

La versione Docker minima per parametri affidabili è v20.10.13 e successive, inclusa nell'AMI 20220607 ottimizzata per Amazon ECS e versioni successive.

Le versioni dell'agente Amazon ECS 1.20.0 e successive non supportano più le versioni di Docker precedenti alla 18.01.0.

```
docker --version
```

5. Scarica il file di agente Amazon ECS appropriato per il sistema operativo e per l'architettura di sistema in uso e installalo.

Per le architetture deb:

```
ubuntu:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-ecs-agent-us-west-2/  
amazon-ecs-init-latest.amd64.deb  
ubuntu:~$ sudo dpkg -i amazon-ecs-init-latest.amd64.deb
```

Per le architetture rpm:

```
fedora:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-ecs-agent-us-west-2/  
amazon-ecs-init-latest.x86_64.rpm  
fedora:~$ sudo yum localinstall -y amazon-ecs-init-latest.x86_64.rpm
```

6. Modifica il `/lib/systemd/system/ecs.service` file e aggiungi la riga seguente alla fine della `[Unit]` sezione.

```
After=cloud-final.service
```

7. (Facoltativo) Per registrare l'istanza con un cluster diverso dal cluster default, modifica il file `/etc/ecs/ecs.config` e aggiungi i contenuti seguenti. L'esempio seguente specifica il cluster `MyCluster`.

```
ECS_CLUSTER=MyCluster
```

Per ulteriori informazioni su queste e altre opzioni di runtime dell'agente, consulta [Configurazione dell'agente del container Amazon ECS](#).

 Note

Facoltativamente, puoi archiviare le variabili di ambiente dell'agente in Amazon S3 (che può essere scaricata nelle istanze del contenitore al momento del lancio utilizzando i dati utente di EC2 Amazon). Ciò è consigliato per le informazioni sensibili, come le credenziali di autenticazione per i repository privati. Per ulteriori informazioni, consultare [Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3](#) e [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

8. Avviare il servizio ecs.

```
ubuntu:~$ sudo systemctl start ecs
```

Esecuzione dell'agente Amazon ECS con modalità di rete host

Quando si esegue l'agente del container di Amazon ECS, `ecs-init` crea il relativo container dell'agente del container con la modalità di rete host. Questa è l'unica modalità di rete supportata per il container dell'agente del container.

Ciò consente di bloccare l'accesso all'[endpoint del servizio di metadati dell' EC2 istanza Amazon](#) per i contenitori avviati dall'agente contenitore. `http://169.254.169.254` Questo garantisce che i container non possano accedere alle credenziali del ruolo IAM dal profilo dell'istanza di container e che le attività utilizzino solo le credenziali per il ruolo del processo IAM. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Assicura inoltre che l'agente del container non debba contendersi le connessioni e il traffico di rete sul bridge `docker0`.

Parametri di configurazione del registro dell'agente container Amazon ECS

L'agente del container di Amazon ECS archivia i log nelle istanze del container.

Per l'agente del container versione 1.36.0 e successive, di default i log si trovano in `/var/log/ecs/ecs-agent.log` sulle istanze Linux e in `C:\ProgramData\Amazon\ECS\log\ecs-agent.log` sulle istanze Windows.

Per l'agente del container versione 1.35.0 e precedenti, di default i log si trovano in `/var/log/ecs/ecs-agent.log.timestamp` sulle istanze Linux e `C:\ProgramData\Amazon\ECS\log\ecs-agent.log.timestamp` sulle istanze Windows.

Di default, i log dell'agente vengono ruotati ogni ora per un massimo di 24 log archiviati.

Di seguito sono riportate le variabili di configurazione dell'agente del container che possono essere utilizzate per modificare il comportamento di registrazione dell'agente predefinito. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

ECS_LOGFILE

Valori di esempio: `/ecs-agent.log`

Valore predefinito in Linux: `Null`

Valore predefinito in Windows: `Null`

Determina la posizione in cui devono essere scritti i log degli agenti. Se stai eseguendo l'agente tramite `ecs-init`, che è il metodo predefinito quando si utilizza l'AMI ottimizzata per Amazon ECS, il percorso all'interno del contenitore è `/log` e lo `ecs-init` monta sull'host. `/var/log/ecs/`

ECS_LOGLEVEL

Valori di esempio: `crit`, `error`, `warn`, `info`, `debug`

Valore predefinito in Linux: `info`

Valore predefinito in Windows: `info`

Il livello di dettaglio da registrare.

ECS_LOGLEVEL_ON_INSTANCE

Valori di esempio: `none`, `crit`, `error`, `warn`, `info`, `debug`

Valore di default su Linux: `none`, se `ECS_LOG_DRIVER` viene impostato esplicitamente su un valore non vuoto; altrimenti lo stesso valore di `ECS_LOGLEVEL`

Valore di default su Windows: `none`, se `ECS_LOG_DRIVER` viene impostato esplicitamente su un valore non vuoto; altrimenti lo stesso valore di `ECS_LOGLEVEL`

Può essere usato per sovrascrivere `ECS_LOGLEVEL` e impostare un livello di dettaglio da registrare nel file di log nell'istanza, separato dal livello registrato nel driver di registrazione.

Se un driver di registrazione è impostato in modo esplicito, i log in istanza sono disattivati per impostazione predefinita. Possono essere riattivati con questa variabile.

ECS_LOG_DRIVER

Valori di esempio: `awslogsfluentd,gelf,json-file,journald,syslog,splunk`

Valore predefinito in Linux: `json-file`

Valore predefinito in Windows: non applicabile

Determina il driver di registrazione utilizzato dal contenitore dell'agente.

ECS_LOG_ROLLOVER_TYPE

Valori di esempio: `size` e `hourly`

Valore predefinito in Linux: `hourly`

Valore predefinito in Windows: `hourly`

Determina se il file di registro dell'agente contenitore viene ruotato ogni ora o in base alle dimensioni. Di default, il file di log dell'agente viene ruotato ogni ora.

ECS_LOG_OUTPUT_FORMAT

Valori di esempio: `logfmt` e `json`

Valore predefinito in Linux: `logfmt`

Valore predefinito in Windows: `logfmt`

Determina il formato di output del log. Quando viene utilizzato il `json` formato, ogni riga del log è una mappa JSON strutturata.

ECS_LOG_MAX_FILE_SIZE_MB

Valori di esempio: `10`

Valore predefinito in Linux: `10`

Valore predefinito in Windows: `10`

Quando la `ECS_LOG_ROLLOVER_TYPE` variabile è impostata su `size`, questa variabile determina la dimensione massima (in MB) del file di registro prima che venga ruotato. Se il tipo di rollover è impostato su `hourly`, questa variabile viene ignorata.

ECS_LOG_MAX_ROLL_COUNT

Valori di esempio: 24

Valore predefinito in Linux: 24

Valore predefinito in Windows: 24

Determina il numero di file di log ruotati da mantenere. I file di log più vecchi vengono eliminati una volta raggiunto questo limite.

Per l'agente del container versione 1.36.0 e successive, il seguente è un file di log di esempio quando viene utilizzato il formato `logfmt`.

```
level=info time=2019-12-12T23:43:29Z msg="Loading configuration" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Image excluded from cleanup: amazon/amazon-ecs-agent:latest" module=parse.go
level=info time=2019-12-12T23:43:29Z msg="Image excluded from cleanup: amazon/amazon-ecs-pause:0.1.0" module=parse.go
level=info time=2019-12-12T23:43:29Z msg="Amazon ECS agent Version: 1.36.0, Commit: ca640387" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Creating root ecs cgroup: /ecs" module=init_linux.go
level=info time=2019-12-12T23:43:29Z msg="Creating cgroup /ecs" module=cgroup_controller_linux.go
level=info time=2019-12-12T23:43:29Z msg="Loading state!" module=statemanager.go
level=info time=2019-12-12T23:43:29Z msg="Event stream ContainerChange start listening..." module=eventstream.go
level=info time=2019-12-12T23:43:29Z msg="Restored cluster 'auto-robc'" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Restored from checkpoint file. I am running as 'arn:aws:ecs:us-west-2:0123456789:container-instance/auto-robc/3330a8a91d15464ea30662d5840164cd' in cluster 'auto-robc'" module=agent.go
```

Di seguito è riportato un file di log di esempio quando viene utilizzato il formato JSON.

```
{"time": "2019-11-07T22:52:02Z", "level": "info", "msg": "Starting Amazon Elastic Container Service Agent", "module": "engine.go"}
```

Per le versioni dell'agente del container 1.35.0 e precedenti, il seguente è il formato del file di log.

```
2016-08-15T15:54:41Z [INFO] Starting Agent: Amazon ECS Agent - v1.12.0 (895f3c1)
2016-08-15T15:54:41Z [INFO] Loading configuration
```

```
2016-08-15T15:54:41Z [WARN] Invalid value for task cleanup duration, will be overridden
to 3h0m0s, parsed value 0, minimum threshold 1m0s
2016-08-15T15:54:41Z [INFO] Checkpointing is enabled. Attempting to load state
2016-08-15T15:54:41Z [INFO] Loading state! module="statemanager"
2016-08-15T15:54:41Z [INFO] Detected Docker versions [1.17 1.18 1.19 1.20 1.21 1.22]
2016-08-15T15:54:41Z [INFO] Registering Instance with ECS
2016-08-15T15:54:41Z [INFO] Registered! module="api client"
```

Configurazione delle istanze di container Amazon ECS per immagini Docker private

L'agente di container di Amazon ECS è in grado di eseguire l'autenticazione con registri privati utilizzando l'autenticazione di base. Quando si abilita l'autenticazione dei registri privati, puoi utilizzare le immagini Docker private nelle tue definizioni di attività. Questa funzionalità è supportata solo dalle attività che utilizzano il tipo di lancio EC2 .

Un altro metodo per abilitare l'autenticazione del registro privato consiste nell' AWS Secrets Manager archiviare le credenziali del registro privato in modo sicuro e quindi farvi riferimento nella definizione del contenitore. In questo modo le attività possono utilizzare immagini da repository privati. Questo metodo supporta attività che utilizzano i tipi di avvio EC2 o Fargate. Per ulteriori informazioni, consulta [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#).

L'agente del container di Amazon ECS all'avvio ricerca due variabili di ambiente:

- ECS_ENGINE_AUTH_TYPE, che specifica il tipo di dati di autenticazione inviati.
- ECS_ENGINE_AUTH_DATA, che contiene le credenziali di autenticazione effettive.

Le varianti Linux dell'AMI ottimizzata per Amazon ECS analizzano il `/etc/ecs/ecs.config` file alla ricerca di queste variabili all'avvio dell'istanza del contenitore e ogni volta che il servizio viene avviato (con il `sudo start ecs` comando). AMIs che non sono ottimizzate per Amazon ECS devono archiviare queste variabili di ambiente in un file e passarle con l'`--env-file` *path_to_env_file* opzione al `docker run` comando che avvia l'agente contenitore.

Important

Non è consigliabile inserire queste variabili di ambiente di autenticazione ad esempio, avviarle con dati EC2 utente Amazon o passarle con l'`--env` opzione al `docker run` comando. Questi metodi non sono ideali per i dati sensibili, come le credenziali di autenticazione. Per informazioni sull'aggiunta sicura delle credenziali di autenticazione alle istanze di container,

consulta [Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3](#).

Formati di autenticazione

Sono disponibili due formati per l'autenticazione di registri privati, `dockercfg` e `docker`.

Formato di autenticazione `dockercfg`

Il formato `dockercfg` utilizza le informazioni di autenticazione archiviate nel file di configurazione che viene creato quando esegui il comando `docker login`. Puoi creare questo file eseguendo il comando `docker login` sul sistema locale e inserendo il nome utente, la password e l'indirizzo e-mail del registro. Puoi inoltre effettuare l'accesso a un'istanza di container ed eseguire il comando qui. A seconda della tua versione Docker, questo file viene salvato come `~/.dockercfg` o `~/.docker/config.json`.

```
cat ~/.docker/config.json
```

Output:

```
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "zq212MzEXAMPLE7o6T25Dk0i"
    }
  }
}
```

Important

Le nuove versioni di Docker creano un file di configurazione come mostrato in precedenza con un oggetto `auths` esterno. L'agente Amazon ECS supporta solo i dati di autenticazione `dockercfg` nel formato seguente senza l'oggetto `auths`. Se è installata l'utilità `jq`, puoi estrarre questi dati con il seguente comando: `cat ~/.docker/config.json | jq .auths`

```
cat ~/.docker/config.json | jq .auths
```

Output:

```
{
  "https://index.docker.io/v1/": {
    "auth": "zq212MzEXAMPLE7o6T25Dk0i",
    "email": "email@example.com"
  }
}
```

Nell'esempio precedente, è necessario aggiungere le seguenti variabili di ambiente al file delle variabili di ambiente (/etc/ecs/ecs.config per l'AMI ottimizzata per Amazon ECS) che l'agente del container di Amazon ECS carica in fase di runtime. Se non stai utilizzando l'AMI ottimizzata per Amazon ECS e stai avviando l'agente manualmente con `docker run`, specifica il file delle variabili d'ambiente con l'opzione `--env-file` *path_to_env_file* all'avvio dell'agente.

```
ECS_ENGINE_AUTH_TYPE=dockercfg
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"auth":"zq212MzEXAMPLE7o6T25Dk0i","email":"email@example.com"}}
```

Puoi configurare più registri privati con la seguente sintassi:

```
ECS_ENGINE_AUTH_TYPE=dockercfg
ECS_ENGINE_AUTH_DATA={"repo.example-01.com":
{"auth":"zq212MzEXAMPLE7o6T25Dk0i","email":"email@example-01.com"},
"repo.example-02.com":
{"auth":"fQ172MzEXAMPLEoF7225DU0j","email":"email@example-02.com"}}
```

Formato di autenticazione docker

Il formato `docker` utilizza una rappresentazione JSON del server di registro con il quale l'agente deve eseguire l'autenticazione. Include anche i parametri di autenticazione richiesti da tale registro (ad esempio il nome utente, la password e l'indirizzo e-mail per l'account). Per un account Docker Hub, la rappresentazione JSON sarà simile a quanto segue:

```
{
  "https://index.docker.io/v1/": {
    "username": "my_name",
    "password": "my_password",
    "email": "email@example.com"
  }
}
```

In questo esempio, è necessario aggiungere le seguenti variabili di ambiente al file delle variabili di ambiente (`/etc/ecs/ecs.config` per l'AMI ottimizzata per Amazon ECS) che l'agente del container di Amazon ECS carica in fase di runtime. Se non stai utilizzando l'AMI ottimizzata per Amazon ECS e stai avviando l'agente manualmente con il comando `docker run`, quando avvii l'agente specifica il file delle variabili d'ambiente con l'opzione `--env-file` *path_to_env_file*.

```
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"username":"my_name","password":"my_password","email":"email@example.com"}}
```

Puoi configurare più registri privati con la seguente sintassi:

```
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"repo.example-01.com":
{"username":"my_name","password":"my_password","email":"email@example-01.com"},
"repo.example-02.com":
{"username":"another_name","password":"another_password","email":"email@example-02.com"}}
```

Procedura

Utilizza la procedura seguente per attivare i registri privati per le istanze di container.

Come abilitare i registri privati nell'AMI ottimizzata per Amazon ECS

1. Accedi alla tua istanza di container con SSH.
2. Apri il file `/etc/ecs/ecs.config` e aggiungi i valori `ECS_ENGINE_AUTH_TYPE` ed `ECS_ENGINE_AUTH_DATA` per il registro e l'account:

```
sudo vi /etc/ecs/ecs.config
```

In questo esempio si esegue l'autenticazione di un account utente Docker Hub:

```
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"username":"my_name","password":"my_password","email":"email@example.com"}}
```

3. Controlla se il tuo agente utilizza la variabile di ambiente `ECS_DATADIR` per salvarne lo stato:

```
docker inspect ecs-agent | grep ECS_DATADIR
```

Output:

```
"ECS_DATADIR=/data",
```

 Important

Se il comando precedente non restituisce la variabile di ambiente ECS_DATADIR, è necessario arrestare qualsiasi attività in esecuzione in questa istanza di container prima di arrestare l'agente. Gli agenti più nuovi con la variabile di ambiente ECS_DATADIR salvano il proprio stato e puoi arrestarli e avviarli mentre le attività vengono eseguite senza problemi. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

4. Arresta il servizio ecs:

```
sudo stop ecs
```

Output:

```
ecs stop/waiting
```

5. Riavvia il servizio ecs.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS:

```
sudo stop ecs && sudo start ecs
```

6. (Facoltativo) Puoi verificare se l'agente è in esecuzione e visualizzare alcune informazioni sulla nuova istanza di container interrogando l'operazione API di introspezione dell'agente. Per ulteriori informazioni, consulta [the section called "Introspezione dei contenitori"](#).

```
curl http://localhost:51678/v1/metadata
```

Pulizia automatica delle attività e delle immagini di Amazon ECS

Ogni volta che si colloca un processo in un'istanza di container, l'agente del container di Amazon ECS verifica se le immagini a cui si fa riferimento nell'attività sono le più recenti del tag specificato nel repository. In caso contrario, il comportamento di default consente all'agente di estrarre le immagini dai rispettivi repository. Se aggiorni di frequente le immagini nelle tue attività e nei servizi, il tuo storage dell'istanza di container si può riempire rapidamente con le immagini Docker che non stai più utilizzando e probabilmente non riutilizzerai mai più. Ad esempio, puoi usare una pipeline per l'integrazione e l'implementazione continue (CI/CD).

Note

Il comportamento pull dell'immagine dell'agente Amazon ECS può essere personalizzato utilizzando il parametro `ECS_IMAGE_PULL_BEHAVIOR`. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Analogamente, i container che appartengono ad attività arrestate possono anche consumare lo storage dell'istanza di container con informazioni di log, volumi di dati e altri elementi. Questi elementi sono utili per il debug dei container che si sono arrestati inaspettatamente, ma la maggior parte di questo storage può essere liberata in sicurezza dopo un periodo di tempo.

Di default, l'agente del container di Amazon ECS elimina automaticamente le attività interrotte e le immagini Docker che non vengono utilizzate da nessun processo nelle tue istanze di container.

Note

La funzione di pulizia automatizzata delle immagini richiede almeno la versione 1.13.0 dell'agente del container di Amazon ECS. Per aggiornare il tuo agente all'ultima versione, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Le seguenti variabili di configurazione dell'agente sono disponibili per regolare la tua esperienza di attività automatica e pulizia delle immagini. Per ulteriori informazioni su come impostare queste variabili sulle tue istanze di container, consulta [Configurazione dell'agente del container Amazon ECS](#).

ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION

Questa variabile specifica il tempo di attesa prima di rimuovere qualsiasi container appartenente ad attività arrestate. Il processo di pulizia delle immagini non può eliminare un'immagine finché c'è un container che fa riferimento alla stessa. Quando nessun container (interrotto o in esecuzione) fa riferimento alle immagini, l'immagine diventa una possibile candidata per la pulizia. Per impostazione predefinita, questo parametro è impostato su 3 ore, ma puoi ridurre questo periodo fino a 1 secondo, se necessario per la tua applicazione. Il parametro viene ignorato se imposti il valore a meno di 1 secondo.

ECS_DISABLE_IMAGE_CLEANUP

Se imposti questa variabile su `true`, la pulizia automatizzata delle immagini viene disattivata sull'istanza di container e nessuna immagine viene rimossa automaticamente.

ECS_IMAGE_CLEANUP_INTERVAL

Questa variabile specifica la frequenza con cui il processo di pulizia automatizzata delle immagini deve verificare la presenza di immagini da eliminare. L'impostazione predefinita è ogni 30 minuti, ma puoi ridurre questo periodo a solo 10 minuti per rimuovere le immagini con maggiore frequenza.

ECS_IMAGE_MINIMUM_CLEANUP_AGE

Questa variabile specifica la quantità di tempo minima tra l'estrazione di un'immagine e il momento in cui diventa una candidata per la rimozione. Viene utilizzata per evitare la pulizia di immagini appena estratte. Il valore predefinito è 1 ora.

ECS_NUM_IMAGES_DELETE_PER_CYCLE

Questa variabile specifica il numero di immagini rimovibili durante un singolo ciclo di pulizia. Il valore predefinito è 5 e il valore minimo è 1.

Quando l'agente del container di Amazon ECS è in esecuzione e la pulizia automatizzata delle immagini non è disattivata, l'agente verifica le immagini Docker alle quali nessun container in esecuzione o arrestato fa riferimento a una frequenza determinata dalla variabile `ECS_IMAGE_CLEANUP_INTERVAL`. Se vengono rilevate immagini inutilizzate meno recenti del tempo di pulizia minimo specificato dalla variabile `ECS_IMAGE_MINIMUM_CLEANUP_AGE`, l'agente rimuove fino al numero massimo di immagini specificato con la variabile `ECS_NUM_IMAGES_DELETE_PER_CYCLE`. Le immagini a cui si fa riferimento meno di recente vengono eliminate per prime. Dopo aver rimosso le immagini, l'agente attende fino all'intervallo successivo e ripete il processo.

Pianifica i tuoi contenitori su Amazon ECS

Amazon Elastic Container Service (Amazon ECS) è un sistema di concorrenza ottimistico a stati condivisi che offre funzionalità di pianificazione flessibili per i carichi di lavoro containerizzati. I pianificatori di Amazon ECS utilizzano le stesse informazioni sullo stato del cluster fornite dall'API Amazon ECS per poter prendere decisioni di posizionamento adeguate.

Amazon ECS fornisce un pianificatore di servizi per processi e applicazioni di lunga durata. Offre inoltre la possibilità di eseguire attività autonome o attività pianificate per lavori in batch o attività a esecuzione singola. Puoi specificare le strategie di posizionamento dei processi e i vincoli per l'esecuzione dei processi più adatte alle tue esigenze. Ad esempio, puoi specificare se i processi vengono eseguiti in più zone di disponibilità o all'interno di una singola zona di disponibilità. Puoi inoltre integrare i processi con pianificatori di terze parti o personalizzati.

Opzione	Quando usare	Ulteriori informazioni
Servizio	Il service scheduler è adatto per servizi e applicazioni stateless a esecuzione prolungata. Il pianificatore di servizi, facoltativamente, assicura anche che i processi siano registrati nel load balancer Elastic Load Balancing. Puoi aggiornare i servizi gestiti dal pianificatore di servizi. Ciò potrebbe includere l'implementazione di una nuova definizione di attività o la modifica del numero di processi desiderati in esecuzione. Di default, il pianificatore di servizi distribuisce i processi su più zone di disponibilità. Puoi utilizzare vincoli e strategie di	Servizi Amazon ECS

Opzione	Quando usare	Ulteriori informazioni
	posizionamento dei processi per personalizzare le decisioni riguardo al posizionamento dei processi.	
Processo autonomo	Un'attività autonoma è adatta per processi come i lavori in batch che eseguono un lavoro e poi si interrompono. Ad esempio, puoi avere una chiamata di processo RunTask quando il lavoro è in una coda. L'attività recupera il lavoro dalla coda, esegue il lavoro e quindi si chiude. Grazie a RunTask, puoi consentire alla strategia di posizionamento dei processi di default di distribuire i processi in modo casuale in tutto il cluster. In questo modo si riduce al minimo la possibilità che una singola istanza ottenga un numero sproporzionato di processi.	Attività autonome di Amazon ECS

Opzione	Quando usare	Ulteriori informazioni
Processi pianificati	Un'attività pianificata è adatta quando si hanno attività da eseguire a intervalli prestabiliti nel cluster, è possibile utilizzare EventBridge Scheduler per creare una pianificazione. Puoi eseguire processi per un'operazione di backup o una scansione del log. La EventBridge pianificazione dello Scheduler che crei può eseguire una o più attività nel cluster in orari specifici. L'evento programmato può essere impostato su un intervallo specifico (eseguito ogni <i>N</i> minuto, ora o giorno). In caso contrario, per una pianificazione più complessa, puoi utilizzare un'espressione cron.	Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS

Opzioni di calcolo

Con Amazon ECS, puoi specificare l'infrastruttura su cui vengono eseguite le tue attività o i tuoi servizi. Puoi utilizzare una strategia di capacity provider o un tipo di lancio.

Per Fargate, i fornitori di capacità sono Fargate e Fargate Spot. Infatti EC2, il fornitore di capacità è il gruppo Auto Scaling con le istanze di container registrate.

La strategia dei provider di capacità distribuisce le attività tra i provider di capacità associati al cluster.

In una strategia di provider di capacità possono essere utilizzati solo i provider di capacità che sono già associati a un cluster e hanno uno stato ACTIVE o UPDATING. Puoi associare un provider di capacità a un cluster durante la creazione di un cluster.

In una strategia del provider di capacità, il valore di base opzionale indica il numero minimo di attività da eseguire su un provider di capacità specificato. Solo un provider di capacità in una strategia di provider di capacità può avere una base definita.

Il valore del peso indica la percentuale relativa del numero totale di attività avviate che utilizzano il provider di capacità specificato. Analizza l'esempio seguente. Ad esempio, supponiamo di avere una strategia che contiene due provider di capacità, ognuno con un peso pari a 1. Una volta raggiunta la percentuale di base, le attività si dividono equamente tra i due provider di capacità. Usando la stessa logica, assumiamo di specificare un peso pari a 1 per `capacityProviderA` e un peso pari a 4 per `capacityProviderB`. Quindi, per ogni attività che viene eseguita utilizzando `capacityProviderA`, quattro attività utilizzano `capacityProviderB`.

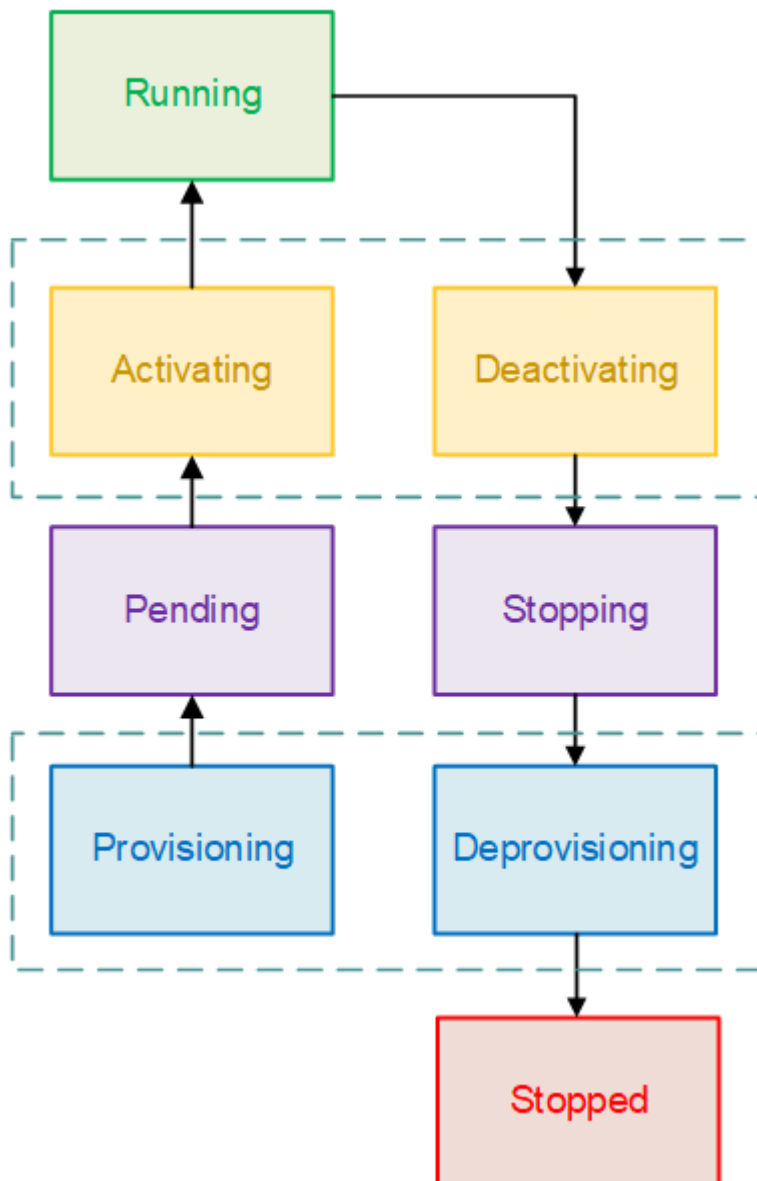
Il tipo di avvio avvia le tue attività direttamente su Fargate o sulle istanze EC2 Amazon che hai registrato manualmente nei tuoi cluster.

Ciclo di vita delle attività di Amazon ECS

Quando viene avviata un'attività, manualmente o come parte di un servizio, questa può attraversare diversi stati prima che termini da sola o venga interrotta manualmente. Alcune attività devono essere eseguite come processi in batch che passano in modo naturale da `PENDING` a `RUNNING` a `STOPPED`. Altre attività, che possono fare parte di un servizio, devono rimanere in esecuzione a tempo indeterminato o a essere ridimensionate in base alle esigenze.

Quando vengono richieste modifiche allo stato delle attività, ad esempio l'interruzione di un'attività o l'aggiornamento del conteggio desiderato di un servizio per aumentarne o ridurne le dimensioni, l'agente del container Amazon ECS traccia tali modifiche come ultimo stato noto dell'attività (`lastStatus`) e stato desiderato dell'attività (`desiredStatus`). È possibile scoprire sia l'ultimo stato noto che lo stato desiderato di un'attività nella console o descrivendo l'attività con l'API o l'AWS CLI.

Il diagramma di seguito mostra il flusso del ciclo di vita dell'attività.



Stati del ciclo di vita

Di seguito è descritto ciascuno stato del ciclo di vita dell'attività.

PROVISIONING

Amazon ECS deve eseguire operazioni aggiuntive prima che il processo venga avviato. Ad esempio, per le attività che utilizzano la modalità di rete `awsvpc`, deve essere predisposta l'interfaccia di rete elastica.

PENDING

Si tratta di uno stato di transizione in cui Amazon ECS è in attesa che l'agente del container effettui ulteriori operazioni. Un'attività rimane nello stato in sospeso fino a quando non sono disponibili risorse per l'attività.

ACTIVATING

Si tratta di uno stato di transizione in cui Amazon ECS deve eseguire operazioni aggiuntive dopo l'avvio dell'attività, ma prima che questa possa passare allo stato RUNNING. Questo è lo stato in cui Amazon ECS recupera le immagini dei container, li crea, configura il networking delle attività, registra i gruppi target dei sistemi di bilanciamento del carico e configura l'individuazione dei servizi.

RUNNING (ESECUZIONE IN CORSO)

L'attività è in esecuzione.

DEACTIVATING

Si tratta di uno stato di transizione in cui Amazon ECS deve eseguire operazioni aggiuntive prima che l'attività venga interrotta. Ad esempio, per le attività che fanno parte di un servizio configurato per utilizzare i gruppi target di Elastic Load Balancings, la cancellazione del gruppo target avviene durante questo stato.

STOPPING

Si tratta di uno stato di transizione in cui Amazon ECS è in attesa che l'agente del container effettui ulteriori operazioni.

Per i contenitori Linux, l'agente del contenitore invierà il SIGTERM segnale per notificare la necessità di terminare e chiudere l'applicazione, quindi invierà un messaggio SIGKILL dopo aver atteso la StopTimeout durata impostata nella definizione dell'attività.

DEPROVISIONING

Amazon ECS deve eseguire operazioni aggiuntive dopo l'arresto del processo, ma prima che questo passi allo stato STOPPED. Ad esempio, per le attività che utilizzano la modalità di rete awsvpc, deve essere scollegata e rimossa l'interfaccia di rete elastica.

STOPPED

L'arresto dell'attività è riuscito.

Se l'operazione è stata interrotta a causa di un errore, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

ELIMINATO

Si tratta di uno stato di transizione quando un'operazione viene interrotta. Questo stato non viene visualizzato nella console, ma viene visualizzato in `describe-tasks`.

In che modo Amazon ECS colloca le attività sulle istanze di container

Puoi utilizzare il posizionamento delle attività per configurare Amazon ECS in modo da collocare le tue attività su istanze di container che soddisfano determinati criteri, ad esempio una zona di disponibilità o un tipo di istanza.

Di seguito sono riportati i componenti per il posizionamento delle attività:

- Strategia di posizionamento delle attività: l'algoritmo per selezionare le istanze del contenitore per il posizionamento delle attività o le attività da terminare. Ad esempio, Amazon ECS può selezionare istanze di container a caso oppure può selezionare istanze di container in modo tale che le attività siano distribuite uniformemente su un gruppo di istanze.
- Gruppo di attività: un gruppo di attività correlate, ad esempio attività di database.
- Vincolo di posizionamento delle attività: si tratta di regole che devono essere soddisfatte per inserire un'attività su un'istanza del contenitore. Se il vincolo non viene soddisfatto, l'attività non viene inserita e rimane nello stato. PENDING Ad esempio, è possibile utilizzare un vincolo per posizionare le attività solo su un particolare tipo di istanza.

Amazon ECS ha diversi algoritmi per i tipi di lancio.

EC2 tipo di lancio

Per le attività che utilizzano il tipo di EC2 avvio, Amazon ECS deve determinare dove collocare l'attività in base ai requisiti specificati nella definizione dell'attività, come CPU e memoria. Analogamente, quando riduci orizzontalmente il conteggio di processi, Amazon ECS deve determinare quali processi terminare. Puoi applicare vincoli e strategie di posizionamento dei processi per personalizzare il modo in cui Amazon ECS posiziona e termina i processi.

Le strategie di posizionamento delle attività predefinite dipendono dal fatto che le attività vengano eseguite manualmente (attività autonome) o all'interno di un servizio. Per le attività eseguite come parte di un servizio Amazon ECS, la strategia di posizionamento delle attività è `spread` e utilizza `attribute:ecs.availability-zone`. Non esiste un vincolo predefinito di posizionamento delle attività per le attività non incluse nei servizi. Per ulteriori informazioni, consulta [Pianifica i tuoi contenitori su Amazon ECS](#).

Note

Le strategie di posizionamento dei processi si basano sul miglior tentativo. Amazon ECS prova a posizionare i processi anche quando l'opzione di posizionamento ottimale non è disponibile. Tuttavia, i vincoli di posizionamento delle attività sono vincolanti, per cui potrebbero impedire il posizionamento delle attività.

Puoi utilizzare strategie e vincoli di posizionamento delle attività contemporaneamente. Ad esempio, puoi utilizzare una strategia di posizionamento e un vincolo di posizionamento delle attività per distribuire le attività all'interno di zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, ma solo per quanto riguarda le istanze G2.

Quando Amazon ECS posiziona i processi, utilizza la seguente procedura per selezionare le istanze di container:

1. Identifica le istanze del contenitore che soddisfano i requisiti di CPU, GPU, memoria e porta nella definizione dell'attività.
2. Identifica le istanze del contenitore che soddisfano i vincoli di posizionamento delle attività.
3. Identifica le istanze del contenitore che soddisfano le strategie di posizionamento delle attività.
4. Seleziona le istanze del contenitore per il posizionamento delle attività.

Tipo di avvio di Fargate

Per processi con tipo di avvio Fargate, le strategie e i vincoli di posizionamento dei processi non sono supportati. Fargate farà del suo meglio per distribuire le attività tra zone di disponibilità accessibili. Se il provider di capacità include sia Fargate che Fargate Spot, il comportamento di distribuzione sarà indipendente per ogni provider.

Usa strategie per definire il posizionamento delle attività di Amazon ECS

Per le attività che utilizzano il tipo di EC2 avvio, Amazon ECS deve determinare dove collocare l'attività in base ai requisiti specificati nella definizione dell'attività, come CPU e memoria.

Analogamente, quando riduci orizzontalmente il conteggio di processi, Amazon ECS deve determinare quali processi terminare. Puoi applicare vincoli e strategie di posizionamento dei processi per personalizzare il modo in cui Amazon ECS posiziona e termina i processi.

Le strategie di posizionamento delle attività predefinite dipendono dal fatto che le attività vengano eseguite manualmente (attività autonome) o all'interno di un servizio. Per le attività eseguite come parte di un servizio Amazon ECS, la strategia di posizionamento delle attività è `spread` e utilizza `attribute:ecs.availability-zone`. Non esiste un vincolo predefinito di posizionamento delle attività per le attività non incluse nei servizi. Per ulteriori informazioni, consulta [Pianifica i tuoi contenitori su Amazon ECS](#).

Note

Le strategie di posizionamento dei processi si basano sul miglior tentativo. Amazon ECS prova a posizionare i processi anche quando l'opzione di posizionamento ottimale non è disponibile. Tuttavia, i vincoli di posizionamento delle attività sono vincolanti, per cui potrebbero impedire il posizionamento delle attività.

Puoi utilizzare strategie e vincoli di posizionamento delle attività contemporaneamente. Ad esempio, puoi utilizzare una strategia di posizionamento e un vincolo di posizionamento delle attività per distribuire le attività all'interno di zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, ma solo per quanto riguarda le istanze G2.

Quando Amazon ECS posiziona i processi, utilizza la seguente procedura per selezionare le istanze di container:

1. Identifica le istanze del contenitore che soddisfano i requisiti di CPU, GPU, memoria e porta nella definizione dell'attività.
2. Identifica le istanze del contenitore che soddisfano i vincoli di posizionamento delle attività.
3. Identifica le istanze del contenitore che soddisfano le strategie di posizionamento delle attività.
4. Seleziona le istanze del contenitore per il posizionamento delle attività.

Le strategie di posizionamento delle attività vengono specificate nella definizione del servizio o la definizione dell'attività utilizzando il `placementStrategy` parametro.

```
"placementStrategy": [  
  {  
    "field": "The field to apply the placement strategy against",  
    "type": "The placement strategy to use"  
  }  
]
```

È possibile specificare le strategie quando si esegue un'attività ([RunTask](#)), si crea un nuovo servizio ([CreateService](#)) o si aggiorna un servizio esistente ([UpdateService](#)).

La tabella seguente descrive i tipi e i campi disponibili.

tipo	Valori di campo validi	
binpack Le attività vengono posizionate sulle istanze di container in modo da lasciare la quantità minima di CPU o memoria inutilizzata. Questa strategia riduce al minimo il numero di istanze di container in uso. Quando viene utilizzata questa strategia e viene intrapresa un'operazione di riduzione orizzontale, Amazon ECS termina i processi. Esegui questa operazione e in base alla quantità di risorse lasciate sull'istanza di container dopo che il processo è stato terminato. L'istanza di container con il maggior numero di risorse disponibili	• cpu • memory	

tipo	Valori di campo validi	
li rimanenti dopo la fine del processo avrà il processo terminato.		
<code>random</code> Le attività vengono posizionate in modo casuale.	Non utilizzato	
<code>spread</code> Le attività vengono posizionate in modo uniforme in base al valore specificato. Le attività di servizio vengono distribuite in base alle attività di tale servizio. Attività standalone sono distribuite sulle attività dallo stesso gruppo di attività. Per ulteriori informazioni sui gruppi di processi, consulta Attività Amazon ECS relative al gruppo. Quando la strategia <code>spread</code> viene utilizzata e viene intrapresa un'operazione di riduzione orizzontale, Amazon ECS seleziona i processi da terminare che mantengono un equilibrio in tutte le zone di disponibilità. All'interno di una zona di disponibilità, i processi vengono selezionati in modo casuale.	• <code>instanceId</code> (oppure <code>host</code>, che ha lo stesso effetto) • qualsiasi piattaforma o attributo personalizzato applicato a un'istanza del contenitore, ad esempio <code>attribute:ecs.availability-zone</code>	

Le strategie di posizionamento delle attività possono essere aggiornate anche per i servizi esistenti. Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

Puoi creare una strategia di posizionamento delle attività che utilizzi più strategie tramite la creazione di una serie di strategie nell'ordine in cui desideri che vengano eseguite. Ad esempio, se desideri distribuire le attività tra le zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, specifica la strategia della zona di disponibilità seguita dalla strategia della memoria. Per le strategie di esempio, consulta [Esempi di strategie di collocamento delle attività di Amazon ECS](#).

Esempi di strategie di collocamento delle attività di Amazon ECS

Puoi specificare strategie di posizionamento delle attività con le seguenti azioni: [CreateService](#), [UpdateService](#), e [RunTask](#).

Esempi

- [Distribuire le attività in modo uniforme all'interno delle zone di disponibilità](#)
- [Distribuire le attività in modo uniforme su tutte le istanze](#)
- [Raggruppare le attività in bin packing in base alla memoria](#)
- [Posizionare le attività casualmente](#)
- [Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e distribuire le attività all'interno delle istanze in ciascuna zona di disponibilità](#)
- [Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e raggruppare le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità](#)
- [Distribuire le attività in modo uniforme tra le istanze e raggruppare le attività in bin packing in base alla memoria](#)

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zona di disponibilità.

```
"placementStrategy": [  
  {  
    "field": "attribute:ecs.availability-zone",  
    "type": "spread"
```

```
}  
]
```

Distribuire le attività in modo uniforme su tutte le istanze

La strategia seguente distribuisce le attività in modo uniforme all'interno di tutte le istanze.

```
"placementStrategy": [  
  {  
    "field": "instanceId",  
    "type": "spread"  
  }  
]
```

Raggruppare le attività in bin packing in base alla memoria

La strategia seguente raggruppa le attività in bin packing in base alla memoria.

```
"placementStrategy": [  
  {  
    "field": "memory",  
    "type": "binpack"  
  }  
]
```

Posizionare le attività casualmente

La strategia seguente posiziona le attività in modo casuale.

```
"placementStrategy": [  
  {  
    "type": "random"  
  }  
]
```

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e distribuire le attività all'interno delle istanze in ciascuna zona di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zone di disponibilità, quindi distribuisce le attività all'interno delle istanze in ciascuna zona di disponibilità.

```
"placementStrategy": [  
  {  
    "field": "attribute:ecs.availability-zone",  
    "type": "spread"  
  },  
  {  
    "field": "instanceId",  
    "type": "spread"  
  }  
]
```

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e raggruppare le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zone di disponibilità, quindi raggruppa le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità.

```
"placementStrategy": [  
  {  
    "field": "attribute:ecs.availability-zone",  
    "type": "spread"  
  },  
  {  
    "field": "memory",  
    "type": "binpack"  
  }  
]
```

Distribuire le attività in modo uniforme tra le istanze e raggruppare le attività in bin packing in base alla memoria

La seguente strategia distribuisce le attività in modo uniforme tra le istanze, quindi raggruppa le attività in bin packing in base alla memoria all'interno di ciascuna istanza.

```
"placementStrategy": [  
  {  
    "field": "instanceId",  
    "type": "spread"  
  },  
  {
```

```
        "field": "memory",  
        "type": "binpack"  
    }  
]
```

Attività Amazon ECS relative al gruppo

Puoi identificare una serie di attività correlate e inserirle in un gruppo di attività. Tutti i processi con lo stesso nome del gruppo di processi sono considerati come un set quando si utilizza la strategia di posizionamento dei processi spread. Ad esempio, supponiamo che stia eseguendo diverse applicazioni in un unico cluster, come database e server Web. Per assicurarti che i database siano bilanciati all'interno delle zone di disponibilità, aggiungili a un gruppo di processi denominato databases, quindi utilizza la strategia di posizionamento dei processi spread. Per ulteriori informazioni, consulta [Usa strategie per definire il posizionamento delle attività di Amazon ECS](#).

I gruppi di processi possono essere utilizzati anche come vincolo di posizionamento dei processi. Quando specifichi un gruppo di attività nel vincolo `memberOf`, le attività vengono inviate solo alle istanze di container presenti nel gruppo di attività specificato. Per vedere un esempio, consulta [Esempio di vincoli di posizionamento delle attività di Amazon ECS](#).

Di default, i processi autonomi utilizzano il nome della famiglia della definizione di attività (ad esempio, `family:my-task-definition`) come nome del gruppo di processi se non è specificato un nome di gruppo di processi personalizzato. I processi avviati come parte di un servizio utilizzano il nome del servizio come nome del gruppo di processi e non possono essere modificati.

Si applicano i seguenti requisiti per il gruppo di attività.

- Il nome di un gruppo di processi deve essere al massimo di 255 caratteri.
- Ciascuna attività può trovarsi in un solo gruppo.
- Dopo l'avvio di un processo, non puoi modificarne il gruppo di processi.

Definisci quali istanze di container Amazon ECS utilizza per le attività

Un vincolo di posizionamento delle attività è una regola relativa a un'istanza di contenitore che Amazon ECS utilizza per determinare se l'attività può essere eseguita sull'istanza. Almeno un'istanza di container deve corrispondere al vincolo. Se non sono presenti istanze corrispondenti al vincolo, l'attività rimane in uno stato PENDING. Quando crei un nuovo servizio o ne aggiorni uno esistente, puoi specificare i vincoli di posizionamento delle attività per le attività del servizio.

È possibile specificare i vincoli di posizionamento delle attività nella definizione del servizio, nella definizione dell'attività o nell'attività utilizzando il parametro. `placementConstraint`

```
"placementConstraints": [
  {
    "expression": "The expression that defines the task placement constraints",
    "type": "The placement constraint type to use"
  }
]
```

La tabella seguente descrive come utilizzare i parametri.

Constraint type (Tipo di vincolo)	Può essere specificato quando	
<code>distinctInstance</code> Posiziona ogni attività attiva su un'istanza di contenitore diversa. Amazon ECS esamina lo stato desiderato delle attività per il posizionamento delle attività. Ad esempio, se lo stato desiderato dell'attività esistente è <code>STOPPED</code>, (ma l'ultimo stato non lo è), una nuova attività in arrivo può essere inserita nella stessa istanza nonostante il vincolo di <code>distinctInstance</code> posizionamento. Pertanto, è possibile che vengano visualizzate 2 attività con l'ultimo stato di nella stessa <code>RUNNING</code> istanza.	- Esecuzione di un'attività RunTask - Creare un nuovo servizio CreateService,	

Constraint type (Tipo di vincolo)	Può essere specificato quando	
 Important Consigliamo ai clienti che cercano un forte isolamento per le proprie attività di utilizzare Fargate. Fargate esegue ogni attività in un ambiente di virtualizzazione hardware. Ciò garantisce che questi carichi di lavoro containerizzati non condividano interfacce di rete, storage temporaneo Fargate, CPU o memoria con altre attività. Per ulteriori informazioni, vedere Panoramic a sulla sicurezza di. AWS Fargate		

Constraint type (Tipo di vincolo)	Può essere specificato quando	
memberOf Posizionamento delle attività in istanze di container che soddisfano un'espressione.	• Esecuzione di un'attività RunTask • Creare un nuovo servizio CreateService, • Creazione di una nuova definizione di attività RegisterTaskDefinition • Creazione di una nuova revisione di una definizione di attività RegisterTaskDefinition • Aggiornamento di un servizio UpdateService	

Quando utilizzi il tipo di `memberOf` vincolo, puoi creare un'espressione utilizzando il linguaggio di interrogazione del cluster che definisce le istanze del contenitore in cui Amazon ECS può collocare le attività. L'espressione è un modo per raggruppare le istanze del contenitore in base agli attributi. L'espressione rientra nel `expression` parametro di `placementConstraint`

Attributi dell'istanza del contenitore Amazon ECS

Puoi aggiungere metadati personalizzati alle istanze di container, note come attributi. Ogni attributo ha un nome e un valore di stringa facoltativo. Puoi utilizzare gli attributi integrati offerti da Amazon ECS oppure definire attributi personalizzati.

Le sezioni seguenti contengono attributi incorporati, facoltativi e personalizzati di esempio.

Attributi integrati

Amazon ECS applica automaticamente i seguenti attributi alle istanze di container.

`ecs.ami-id`

L'ID dell'AMI utilizzato per avviare l'istanza. Un valore di esempio per questo attributo è `ami-1234abcd`.

`ecs.availability-zone`

La zona di disponibilità dell'istanza. Un valore di esempio per questo attributo è `us-east-1a`.

`ecs.instance-type`

Il tipo di istanza dell'istanza. Un valore di esempio per questo attributo è `g2.2xlarge`.

`ecs.os-type`

Il sistema operativo dell'istanza. I valori possibili per questo attributo sono `linux` e `windows`.

`ecs.os-family`

La versione del sistema operativo dell'istanza.

Per le istanze Linux, il valore valido è `LINUX`. Per le istanze Windows, ECS imposta il valore nel formato `WINDOWS_SERVER_<OS_Release>_<FULL or CORE>`. I valori validi sono `WINDOWS_SERVER_2022_FULL`, `WINDOWS_SERVER_2022_CORE`, `WINDOWS_SERVER_20H2_CORE`, `WINDOWS_SERVER_2019_FULL`, `WINDOWS_SERVER_2019_CORE` e `WINDOWS_SERVER_2016_FULL`.

Questo è importante per i contenitori Windows e Windows containers on AWS Fargate perché la versione del sistema operativo di ogni contenitore Windows deve corrispondere a quella dell'host. Se la versione Windows dell'immagine del container è diversa da quella dell'host, il container non si avvia. Per ulteriori informazioni, consulta [Compatibilità con la versione del container Windows](#) sul sito Web della documentazione Microsoft.

Se il tuo cluster esegue più versioni di Windows, puoi assicurarti che un'attività venga inserita su un' EC2 istanza in esecuzione sulla stessa versione utilizzando il vincolo di posizionamento: `memberOf(attribute:ecs.os-family == WINDOWS_SERVER_<OS_Release>_<FULL or CORE>)` Per ulteriori informazioni, consulta [the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”](#).

`ecs.cpu-architecture`

Architettura della CPU per l'istanza. I valori di esempio per questo attributo sono `x86_64` e `arm64`.

`ecs.vpc-id`

VPC in cui è stata avviata l'istanza. Un valore di esempio per questo attributo è `vpc-1234abcd`.

`ecs.subnet-id`

La sottorete utilizzata dall'istanza. Un valore di esempio per questo attributo è `subnet-1234abcd`.

Attributi facoltativi

Amazon ECS può aggiungere i seguenti attributi alle istanze di container.

`ecs.awsvpc-trunk-id`

Se questo attributo esiste, l'istanza dispone di un'interfaccia di rete trunk. Per ulteriori informazioni, consulta [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#).

`ecs.outpost-arn`

Se questo attributo esiste, contiene l'Amazon Resource Name (ARN) dell'Outpost. Per ulteriori informazioni, consulta [the section called “Amazon Elastic Container Service su AWS Outposts”](#).

`ecs.capability.external`

Se questo attributo esiste, l'istanza viene identificata come un'istanza esterna. Per ulteriori informazioni, consulta [Cluster Amazon ECS per il tipo di lancio esterno](#).

Attributi personalizzati

Puoi applicare attributi personalizzati alle istanze di container. Ad esempio, puoi definire un attributo con il nome "stack" e un valore di "prod".

Quando si specificano attributi personalizzati, è necessario considerare quanto segue.

- Il `name` può contenere un massimo di 128 caratteri e può contenere lettere (maiuscole e minuscole), numeri, trattini, caratteri di sottolineatura, barre, barre inverse o punti.
- Il `value` può contenere un massimo di 128 caratteri e può contenere lettere (maiuscole e minuscole), numeri, trattini, caratteri di sottolineatura, punti, chiocciola (@), barre, barre inverse, due punti o spazi. Il valore non può contenere spazi iniziali o finali.

Crea espressioni per definire istanze di container per le attività di Amazon ECS

Le query di cluster sono espressioni che consentono di raggruppare gli oggetti. Ad esempio, puoi raggruppare le istanze di container per attributi, ad esempio la zona di disponibilità, il tipo di istanza

o metadati personalizzati. Per ulteriori informazioni, consulta [Attributi dell'istanza del contenitore Amazon ECS](#).

Dopo aver definito un gruppo di istanze di container, potrai personalizzare Amazon ECS per posizionare i processi su istanze di container basate sul gruppo. Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#) e [Creazione di un servizio Amazon ECS utilizzando la console](#). Puoi inoltre applicare un filtro di gruppo per elencare le istanze di container.

Sintassi delle espressioni

Le espressioni presentano la sintassi seguente:

```
subject operator [argument]
```

Subject

L'attributo o il campo da valutare.

agentConnected

Seleziona le istanze di container in base allo stato di connessione dell'agente del container di Amazon ECS. È possibile usare questo filtro per cercare le istanze con agenti di container che sono disconnessi.

Operatori validi: equals (==), not_equals (!=), in, not_in (!in), matches (=~), not_matches (!~)

agentVersion

Seleziona le istanze di container in base alla versione dell'agente del container di Amazon ECS. È possibile usare questo filtro per trovare le istanze che eseguono le versioni obsolete dell'agente del container Amazon ECS.

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

attribute:*attribute-name*

Seleziona le istanze di container per attributo. Per ulteriori informazioni, consulta [Attributi dell'istanza del contenitore Amazon ECS](#).

ec2InstanceId

Seleziona le istanze del contenitore in base al relativo ID di EC2 istanza Amazon.

Operatori validi: equals (==), not_equals (!=), in, not_in (!in), matches (=~), not_matches (!~)

registeredAt

Seleziona le istanze di container in base alla loro data di registrazione dell'istanza di container. È possibile usare questo filtro per individuare le istanze appena registrate o le istanze molto vecchie.

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

Formati di data validi: 2018-06-18T22:28:28+00:00, 2018-06-18T22:28:28Z, 2018-06-18T22:28:28, 2018-06-18

runningTasksCount

Seleziona le istanze di container in base al numero di attività in esecuzione. È possibile usare questo filtro per trovare le istanze che sono vuote o quasi vuote (poche attività in esecuzione su di esse).

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

task:group

Seleziona le istanze di container per gruppo di attività. Per ulteriori informazioni, consulta [Attività Amazon ECS relative al gruppo](#).

Operatore

L'operatore di confronto. Sono supportati i seguenti operatori.

Operatore	Descrizione
==, equals	Uguaglianza stringhe
!=, not_equals	Disuguaglianza stringhe
>, greater_than	Maggiore di
>=, greater_than_equal	Maggiore di o uguale a
<, less_than	Minore di

Operatore	Descrizione
<code><=</code> , <code>less_than_equal</code>	Minore di o uguale a
<code>exists</code>	Il soggetto esiste
<code>!exists</code> , <code>not_exists</code>	Il soggetto non esiste
<code>in</code>	Il valore è nell'elenco di argomenti
<code>!in</code> , <code>not_in</code>	Il valore non è nell'elenco di argomenti
<code>=~</code> , <code>matches</code>	Corrispondenza modelli
<code>!~</code> , <code>not_matches</code>	Mancata corrispondenza modelli

Note

Una singola espressione non può contenere parentesi. Tuttavia, le parentesi possono essere utilizzate per specificare la precedenza in espressioni composte.

Argomento

Per molti operatori, l'argomento è un valore letterale.

Gli operatori `in` e `not_in` prevedono come argomento un elenco di argomenti. È necessario specificare un elenco di argomenti nel modo seguente:

```
[argument1, argument2, ..., argumentN]
```

Gli operatori `matches` e `not_matches` prevedono un argomento conforme alla sintassi di espressione regolare di Java. Per ulteriori informazioni, consulta [java.util.regex.Pattern](https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html).

Espressioni composte

Puoi combinare espressioni tramite gli operatori booleani seguenti:

- `&&`, e
- `||`, oppure

- !, not

Puoi specificare la precedenza utilizzando le parentesi:

```
(expression1 or expression2) and expression3
```

Espressioni di esempio

Vengono riportate di seguito espressioni di esempio.

Esempio: uguaglianza stringhe

L'espressione seguente seleziona le istanze con il tipo di istanza specificato.

```
attribute:ecs.instance-type == t2.small
```

Esempio: elenco di argomenti

L'espressione seguente seleziona le istanze nelle zone di disponibilità us-east-1a o us-east-1b.

```
attribute:ecs.availability-zone in [us-east-1a, us-east-1b]
```

Esempio: espressione composta

L'espressione composta seguente seleziona le istanze G2 che non si trovano nella zona di disponibilità us-east-1d.

```
attribute:ecs.instance-type =~ g2.* and attribute:ecs.availability-zone != us-east-1d
```

Esempio: affinità di attività

L'espressione seguente seleziona le istanze che non ospitano le attività nel gruppo `service:production`.

```
task:group == service:production
```

Esempio: non affinità di attività

L'espressione seguente seleziona le istanze che non ospitano i processi nel gruppo di database.

```
not(task:group == database)
```

Esempio: conteggio dell'attività in esecuzione

L'espressione seguente seleziona le istanze sulle quali è in esecuzione solo un'attività.

```
runningTasksCount == 1
```

Esempio: Versione dell'agente del container di Amazon ECS

L'espressione seguente seleziona le istanze che eseguono una versione dell' agente del container inferiore a 1.14.5.

```
agentVersion < 1.14.5
```

Esempio: tempo di registrazione dell'istanza

L'espressione seguente seleziona le istanze che sono state registrate prima del 13 febbraio 2018.

```
registeredAt < 2018-02-13
```

Esempio: ID EC2 istanza Amazon

L'espressione seguente seleziona le istanze con la seguente istanza Amazon EC2 . IDs

```
ec2InstanceId in ['i-abcd1234', 'i-wxyx7890']
```

Esempio di vincoli di posizionamento delle attività di Amazon ECS

Di seguito sono elencati esempi di vincoli del posizionamento delle attività.

Questo esempio utilizza il `memberOf` vincolo per inserire attività su istanze t2. Può essere specificato con le seguenti azioni: [CreateService](#), [UpdateService](#), [RegisterTaskDefinitionRunTask](#)

```
"placementConstraints": [  
  {  
    "expression": "attribute:ecs.instance-type =~ t2.*",  
    "type": "memberOf"  
  }  
]
```

L'esempio utilizza il vincolo `memberOf` per posizionare le attività di replica su istanze con attività nel gruppo di attività `daemon-service` del servizio `daemon`, rispettando tutte le strategie di posizionamento eventualmente specificate. Questo vincolo garantisce che le attività del servizio `daemon` vengano posizionate sull' EC2 istanza prima delle attività del servizio di replica.

Sostituisci `daemon-service` con il nome del servizio `daemon`.

```
"placementConstraints": [  
  {  
    "expression": "task:group == service:daemon-service",  
    "type": "memberOf"  
  }  
]
```

Nell'esempio è utilizzato il vincolo `memberOf` per posizionare i processi su istanze con altri processi nel gruppo di processi `databases`, rispettando tutte le strategie di posizionamento dei processi che sono eventualmente specificate. Per ulteriori informazioni sui gruppi di processi, consulta [Attività Amazon ECS relative al gruppo](#). Può essere specificato con le seguenti azioni: [CreateService](#), e [UpdateServiceRegisterTaskDefinitionRunTask](#)

```
"placementConstraints": [  
  {  
    "expression": "task:group == databases",  
    "type": "memberOf"  
  }  
]
```

La restrizione `distinctInstance` posiziona ciascuna attività nel gruppo su un'istanza diversa. Può essere specificato con le seguenti azioni: [CreateService](#), [UpdateService](#), e [RunTask](#)

Amazon ECS esamina lo stato desiderato delle attività per il posizionamento delle attività. Ad esempio, se lo stato desiderato dell'attività esistente è `STOPPED`, (ma l'ultimo stato non lo è), una nuova attività in arrivo può essere inserita nella stessa istanza nonostante il vincolo di `distinctInstance` posizionamento. Pertanto, è possibile che vengano visualizzate 2 attività con l'ultimo stato di nella stessa `RUNNING` istanza.

```
"placementConstraints": [  
  {  
    "type": "distinctInstance"  
  }  
]
```

]

Attività autonome di Amazon ECS

Puoi eseguire l'applicazione come attività quando hai un'applicazione che esegue alcune operazioni e poi interrompe, ad esempio un processo in batch. Se si desidera eseguire un'operazione una sola volta, è possibile utilizzare la console AWS CLI, APIs, o SDKs.

Se è necessario eseguire l'applicazione in base a una pianificazione basata sulla frequenza, su cron o una tantum, è possibile creare una pianificazione utilizzando Scheduler. EventBridge

Workflow delle attività

Quando avvii attività di Amazon ECS (attività autonome o tramite servizi Amazon ECS), un'attività viene creata e inizialmente spostata nello stato. PROVISIONING Quando un'attività è nello PROVISIONING stato, non esistono né l'attività né i contenitori perché Amazon ECS deve trovare la capacità di calcolo per collocare l'attività.

Amazon ECS seleziona la capacità di elaborazione appropriata per l'attività in base al tipo di avvio o alla configurazione del provider di capacità. Puoi utilizzare i provider di capacità e le strategie dei provider di capacità con i tipi di EC2 lancio di Fargate e Amazon. Con Fargate, non è necessario pensare al provisioning, alla configurazione e alla scalabilità della capacità del cluster. Fargate si occupa di tutta la gestione dell'infrastruttura per le tue attività. Per il tipo di EC2 avvio, puoi gestire la capacità del cluster registrando EC2 le istanze Amazon nel cluster oppure puoi utilizzare la scalabilità automatica del cluster per semplificare la gestione della capacità di calcolo. La scalabilità automatica del cluster si occupa della scalabilità dinamica della capacità del cluster, in modo che tu possa concentrarti sulle attività in esecuzione. Amazon ECS determina dove collocare l'attività in base ai requisiti specificati nella definizione dell'attività, come CPU e memoria, nonché ai vincoli e alle strategie di posizionamento. Per ulteriori informazioni, consultare [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

Se utilizzi un provider di capacità con scalabilità gestita abilitata, le attività che non possono essere avviate a causa della mancanza di capacità di elaborazione vengono spostate PROVISIONING nello stato anziché fallire immediatamente. Dopo aver trovato la capacità di collocare l'attività, Amazon ECS fornisce gli allegati necessari (ad esempio, Elastic Network Interfaces (ENIs) per le attività in awsvpc modalità). Utilizza l'agente container Amazon ECS per estrarre le immagini dei contenitori e quindi avviare i contenitori. Dopo il completamento del provisioning e il lancio dei contenitori

pertinenti, Amazon ECS sposta l'attività in stato. RUNNING Per informazioni sugli stati delle attività, consulta. [Ciclo di vita delle attività di Amazon ECS](#)

Ottimizza il tempo di avvio delle attività di Amazon ECS

Per velocizzare l'avvio delle attività, prendi in considerazione i seguenti consigli.

- Memorizza nella cache le immagini dei contenitori e le istanze binpack

Se utilizzi il tipo di EC2 avvio, puoi configurare il comportamento pull dell'agente container Amazon ECS su `ECS_IMAGE_PULL_BEHAVIOR:prefer-cached`. L'immagine viene recuperata in remoto se non è presente alcuna immagine memorizzata nella cache. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. La pulizia automatica delle immagini è disattivata per il contenitore per garantire che l'immagine memorizzata nella cache non venga rimossa. In questo modo si riduce il tempo di recupero delle immagini per i successivi avvii. L'effetto della memorizzazione nella cache è ancora maggiore quando nelle istanze del contenitore è presente un'elevata densità di attività, che è possibile configurare utilizzando la strategia di posizionamento. binpack La memorizzazione nella cache delle immagini dei container è particolarmente utile per i carichi di lavoro basati su Windows, che di solito hanno immagini container di grandi dimensioni (decine di). GBs Quando utilizzi la strategia di binpack posizionamento, puoi anche prendere in considerazione l'utilizzo del trunking Elastic Network Interface (ENI) per collocare più attività con la modalità di aws vpc rete su ogni istanza del contenitore. Il trunking ENI aumenta il numero di attività che è possibile eseguire in modalità. aws vpc Ad esempio, un'istanza c5.large che può supportare l'esecuzione simultanea di solo 2 attività, può eseguire fino a 10 attività con il trunking ENI.

- Scegli una modalità di rete ottimale

Sebbene ci siano molti casi in cui la modalità di aws vpc rete è ideale, questa modalità di rete può aumentare intrinsecamente la latenza di avvio delle attività, perché per ogni attività in modalità aws vpc, i flussi di lavoro di Amazon ECS devono fornire e collegare un ENI richiamando Amazon, EC2 APIs che aggiunge un sovraccarico di diversi secondi all'avvio delle attività. Al contrario, un vantaggio chiave dell'utilizzo della modalità di aws vpc rete è che ogni attività ha un gruppo di sicurezza che consente o nega il traffico. Ciò significa che hai una maggiore flessibilità per controllare le comunicazioni tra attività e servizi a un livello più granulare. Se la velocità di implementazione è la tua priorità, puoi prendere in considerazione l'utilizzo della bridge modalità per velocizzare l'avvio delle attività. Per ulteriori informazioni, consulta [the section called “AWSVPC modalità di rete”](#).

- Tieni traccia del ciclo di vita dell'avvio delle attività per trovare opportunità di ottimizzazione

Spesso è difficile conoscere il tempo necessario per avviare l'applicazione. L'avvio dell'immagine del contenitore, l'esecuzione di script di avvio e altre configurazioni durante l'avvio dell'applicazione possono richiedere una quantità di tempo sorprendente. Puoi utilizzare l'endpoint di metadati Task per pubblicare metriche per tenere traccia del tempo di avvio dell'applicazione, dalla risposta dei metadati del contenitore `StartedAt` all'ora dell'attività o del servizio. `StartedAt` Grazie a questi dati, è possibile comprendere in che modo l'applicazione contribuisce al tempo totale di avvio e individuare aree in cui è possibile ridurre il sovraccarico non necessario specifico dell'applicazione e ottimizzare le immagini del contenitore. Per ulteriori informazioni, consulta [Ottimizza la capacità e la disponibilità di Amazon ECS](#).

- Scegliete un tipo di istanza ottimale (per il tipo di avvio) EC2

La scelta del tipo di istanza corretto si basa sulla prenotazione di risorse (ad esempio CPU, memoria) configurata per l'attività. Pertanto, quando si ridimensiona l'istanza, è possibile calcolare quante attività possono essere posizionate su una singola istanza. Un semplice esempio di attività ben posizionata è l'hosting di 4 attività che richiedono 0,5 vCPU e 2 GB di riserve di memoria in un'istanza `m5.large` (che supporta 2 vCPU e 8 GB di memoria). Le prenotazioni di questa definizione di attività sfruttano appieno le risorse dell'istanza.

Esecuzione di un'applicazione come attività Amazon ECS

Puoi creare un'attività per un processo singolo utilizzando AWS Management Console

Per creare un'attività autonoma ()AWS Management Console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. La console Amazon ECS ti consente di creare un'attività autonoma dalla pagina dei dettagli del cluster o dall'elenco di revisione delle definizioni delle attività. Utilizza i seguenti passaggi per creare un'attività autonoma in base alla pagina delle risorse scelta.

Per avviare un servizio da	Fasi	
La pagina dei dettagli del cluster	a. Nella pagina Cluster, seleziona il cluster in cui creare il servizio. b. Dalla scheda Attività, scegli Esegui attività.	

Per avviare un servizio da	Fasi	
La pagina di revisione della definizione dell'attività	- Nella pagina Definizioni delle attività, scegli la famiglia di definizioni delle attività per visualizzare le revisioni per quella famiglia. - Seleziona la revisione che desideri utilizzare. - Dal menu Distribuisci, scegli Esegui attività.	

- Per Cluster esistente, scegli il cluster.

Scegli Crea cluster per eseguire l'attività su un nuovo cluster

- Scegli in che modo le tue attività sono distribuite nell'infrastruttura del cluster. In Configurazione di elaborazione, scegli la tua opzione. Per utilizzare una strategia di provider di capacità, devi configurare i provider di capacità a livello di cluster.

Se non hai configurato il cluster per utilizzare un provider di capacità, utilizza invece un tipo di avvio.

Metodo di distribuzione	Fasi	
Strategia del provider di capacità	- Nella sezione Compute options (Opzioni di calcolo), seleziona Capacity provider strategy (Strategia del provider di capacità). - Scegli una strategia: - Per utilizzare una strategia del provider di capacità predefinita del cluster, scegli Use	

Metodo di distribuzione	Fasi	
	cluster default (Usa impostazione predefinita del cluster. • Se il cluster non dispone di una strategia del provider di capacità predefinito o per utilizzare una strategia personalizzata, scegli Use custom (Usa personalizzato), Add capacity provider strategy (Aggiungi strategia del provider di capacità) e definisci la strategia del provider di capacità personalizzata specificando Base (Base), Capacity provider (Provider di capacità) e Weight (Peso).	
	 Note Per utilizzare un provider di capacità in una strategia , il provider di capacità deve essere associato al cluster.	

Metodo di distribuzione	Fasi	
Tipo di avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Launch type (Tipo di avvio), seleziona un tipo di avvio. c. (Facoltativo) Quando viene specificato il tipo di avvio Fargate, per Versione piattaforma specifica la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

5. In Configurazione di distribuzione, procedi come segue:

a. In Definizione dell'attività, inserisci la definizione dell'attività.

 Important

La console convalida la selezione per garantire che la famiglia di definizioni dell'attività e la revisione selezionate siano compatibili con la configurazione di calcolo definita.

b. Per Desired tasks (Attività desiderate), specifica il numero di attività da avviare.

c. Per Gruppo di attività, immettere il nome del gruppo di attività.

6. Se la tua definizione di attività utilizza la modalità di rete awsvpc, espandi Networking (Rete). Per specificare una configurazione personalizzata, completa la procedura riportata di seguito.

a. Per VPC seleziona il VPC da utilizzare.

- b. Per Subnets (Sottoreti), seleziona una o più sottoreti nel VPC che lo scheduler di attività deve prendere in considerazione quando posiziona le attività.

⚠ Important

Per la modalità di rete `aws-vpc` sono supportate solo sottoreti private. Le attività non ricevono indirizzi IP pubblici. Pertanto, è necessario un gateway NAT per l'accesso a Internet in uscita e il traffico Internet in entrata viene instradato attraverso un load balancer.

- c. Per Gruppi di sicurezza puoi scegliere un gruppo di sicurezza esistente o crearne uno nuovo. Per utilizzare un gruppo di sicurezza esistente, scegli il gruppo di sicurezza e passa alla fase successiva. Per creare un nuovo gruppo di sicurezza, scegliere `Create a new security group` (Crea un nuovo gruppo di sicurezza). È necessario specificare un nome e una descrizione del gruppo di sicurezza e aggiungere una o più regole in entrata per il gruppo di sicurezza.
- d. Per IP pubblico scegli se assegnare automaticamente un indirizzo IP pubblico all'interfaccia di rete elastica (ENI) del processo stesso.

AWS Fargate alle attività può essere assegnato un indirizzo IP pubblico quando vengono eseguite in una sottorete pubblica in modo che abbiano un percorso verso Internet. EC2 alle attività non può essere assegnato un IP pubblico utilizzando questo campo. Per ulteriori informazioni, consulta le [opzioni di rete delle attività di Amazon ECS per il tipo di lancio Fargate e Allocazione di un'interfaccia di rete per un'attività Amazon ECS](#).

7. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

Il nome del volume e il tipo di volume vengono configurati durante la creazione di una revisione della definizione di attività e non possono essere modificati quando si esegue un'attività autonoma. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione dell'attività ed eseguire un'attività utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per il tipo di volume EBS, scegli il tipo di volume	

Per configurare questo tipo di volume	Esegui questa operazione	
	EBS che desideri allegare all'attività. b. Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettete il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per io1 i tipi di volume e io2 gp3 d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). MiBps Questo valore è configurabile solo per il tipo di volume. gp3 e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno	

Per configurare questo tipo di volume	Esegui questa operazione	
	snapshot se desideri creare un volume da uno snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Se si specifica un ID snapshot, è possibile specificare una velocità di inizializzazione del volume (MiB/s). Immettete un valore compreso tra 100 e 300, in MiB/s, che determinerà la velocità di caricamento dei dati dall'istantanea specifica utilizzando Snapshot ID per la creazione del volume. g. Per la politica di terminazione, deseleziona la casella di controllo se desideri che il volume configurato per l'allegato all'attività venga preservato o dopo la fine dell'attività. Per impostazione predefinita, i volumi EBS allegati alle attività vengono eliminati quando l'attività viene terminata. h. Per Tipo di file system, scegli il tipo di file system	

Per configurare questo tipo di volume	Esegui questa operazione	
	che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico . L'impostazione predefinita per Linux è XFS. Per i volumi creati da un'istanza, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istanza. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. i. Per il ruolo Infrastruttura, scegli un ruolo IAM con le autorizzazioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy <code>AmazonECSInfrastructureRolePolicyForVolumes</code> gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con	

Per configurare questo tipo di volume	Esegui questa operazione	
	autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessari e, consulta. Ruolo IAM dell'infrastruttura Amazon ECS j. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specificata nell'impostazione. Se scegli Default e la crittografia predefinita di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittografia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà	

Per configurare questo tipo di volume	Esegui questa operazione	
	crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se crei un volume da un'istanza anea crittografata. k. Se hai scelto Personalizzato per la crittografia, devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazioni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. l. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione dell'attività o fornendo tag personalizzati.	

Per configurare questo tipo di volume	Esegui questa operazione	
	Se desideri propagare i tag dalla definizione dell'attività, scegli Definizione dell'attività per Propagate i tag da. Se scegli Non propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS.	

8. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Task Placement (Posizionamento attività), quindi scegli una tra le seguenti opzioni.

Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

- AZ Balanced Spread: distribuisce le attività tra le zone di disponibilità e tra le istanze di container nella zona di disponibilità.
- AZ Balanced BinPack: distribuisce le attività tra le zone di disponibilità e tra le istanze di container con la minima memoria disponibile.
- BinPack— Distribuisce le attività in base alla quantità minima disponibile di CPU o memoria.

- Un'attività per host: posiziona al massimo un'attività dal servizio su ogni istanza del contenitore.
- Personalizzato: definisci la tua strategia di posizionamento delle attività.

Se hai scelto Custom (Personalizzato), definisci l'algoritmo per il posizionamento delle attività e le regole che vengono prese in considerazione durante il posizionamento delle attività.

- In Strategy (Strategia), per Type (Tipo) e Field (Campo), scegli l'algoritmo e l'entità da utilizzare per l'algoritmo.

Puoi aggiungere un massimo di 5 strategie.

- In Vincolo, per Tipo ed Espressione, scegli la regola e l'attributo per il vincolo.

Ad esempio, per impostare il vincolo per posizionare le attività su istanze T2, per Expression (Espressione), immetti `attribute:ecs.instance-type =~ t2.*`.

Puoi aggiungere un massimo di 10 vincoli.

9. (Facoltativo) Per sovrascrivere il ruolo IAM dell'attività o il ruolo di esecuzione dell'attività definito nella definizione dell'attività, espandi Task overrides (Sostituzione dei processi), quindi completa la procedura seguente:

- a. Per Ruolo attività, scegli un ruolo IAM per questa attività. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Vengono visualizzati solo i ruoli con la relazione di attendibilità `ecs-tasks.amazonaws.com`. Per istruzioni su come creare manualmente un ruolo IAM, consulta [Creazione del ruolo IAM dell'attività](#).

- b. Per Ruolo di esecuzione attività, scegli un ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

10. (Facoltativo) Per sovrascrivere i comandi del container e le variabili di ambiente, espandi Container Overrides (Sostituzione dei container), quindi espandi il container.

- Per inviare al container un comando diverso dal comando di definizione dell'attività, inserisci il comando Docker in Sostituzione comando.
- Per aggiungere una variabile di ambiente, seleziona Add Environment Variable (Aggiungi variabile di ambiente). In Chiave, digita il nome della variabile di ambiente. Per Valore immetti un valore di stringa per il valore di ambiente (senza le virgolette doppie (" ")).

AWS racchiude le stringhe tra virgolette doppie (« ») e passa la stringa al contenitore nel seguente formato:

```
MY_ENV_VAR="This variable contains a string."
```

11. (Facoltativo) Per identificare la tua attività, espandi la sezione Tags (Tag), quindi configura i tag.

Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag di definizione delle attività, seleziona Turn on Amazon ECS managed tags (Attiva i tag gestiti da Amazon ECS), quindi seleziona Task definitions (Definizioni di attività).

Aggiungi o rimuovi un tag.

- [Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
- [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).

12. Scegli Create (Crea).

Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS

EventBridge Scheduler è uno scheduler senza server che consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Fornisce funzionalità di pianificazione una tantum e ricorrenti indipendentemente dai bus e dalle regole degli eventi. EventBridge Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole EventBridge pianificate, con un set più ampio di operazioni e servizi API target. AWS EventBridge Scheduler fornisce le seguenti pianificazioni che puoi configurare per le tue attività nella console Scheduler: EventBridge

- Basata sulla frequenza
- Basate su cron

Puoi configurare le pianificazioni basate su cron in qualsiasi fuso orario.

- Pianificazioni una tantum

Puoi configurare le pianificazioni una tantum in qualsiasi fuso orario.

Puoi pianificare Amazon ECS utilizzando Amazon EventBridge Scheduler.

Sebbene sia possibile creare un'attività pianificata nella console Amazon ECS, attualmente la console EventBridge Scheduler offre più funzionalità.

Prima di pianificare un'attività, completa la procedura seguente:

1. Usa la console VPC per ottenere la sottorete IDs in cui vengono eseguite le attività e il gruppo di sicurezza IDs per le sottoreti. Per ulteriori informazioni, consulta [Subnet per il tuo VPC](#) e [Controlla il traffico verso AWS le tue risorse utilizzando i gruppi di sicurezza nella](#) Amazon VPC User Guide.
2. Configura il ruolo di esecuzione di EventBridge Scheduler. Per ulteriori informazioni, consulta [Configurare il ruolo di esecuzione](#) nella Amazon EventBridge Scheduler User Guide.
3. Se desideri utilizzare una strategia di capacity provider per eseguire l'attività, devi avere un provider di capacità associato al cluster.

Per creare una nuova pianificazione utilizzando la console

1. Apri la console Amazon EventBridge Scheduler a <https://console.aws.amazon.com/scheduler/casa>.
2. Nella pagina Pianificazioni, scegli Crea pianificazione.
3. Nella pagina Specifica i dettagli della pianificazione, nella sezione Nome e descrizione della pianificazione, effettua le seguenti operazioni:
 - a. Per Nome pianificazione, inserisci un nome per la pianificazione. Ad esempio, **MyTestSchedule**.
 - b. (Facoltativo) Per Descrizione, inserisci una descrizione per la pianificazione. Ad esempio, **TestSchedule**.
 - c. Per Schedule group, scegli un gruppo di pianificazione. Se non hai un gruppo, scegli predefinito. Per creare un gruppo di pianificazioni, scegli crea la tua pianificazione.

I gruppi di pianificazione vengono utilizzati per aggiungere tag a gruppi di pianificazioni.

4. Scegli le opzioni di pianificazione.

Ricorrenza	Esegui questa operazione...	
Pianificazione una tantum	Per Data e ora, effettua le seguenti operazioni:	

Ricorrenza	Esegui questa operazione...	
Una pianificazione unica richiama una destinazione solo una volta alla data e all'ora specificate.	• Inserisci una data valida in formato YYYY/MM/DD . • Inserisci un timestamp in formato hh:mm 24 ore. • Per Fuso orario, scegli il fuso orario.	

Ricorrenza	Esegui questa operazione...	
Pianificazione ricorrente Una pianificazione ricorrent e richiama una destinazione con una frequenza specifica utilizzando un'espressione cron o un'espressione rate.	a. Per Tipo di pianificazione, esegui una delle seguenti operazioni: • Per utilizzare un'espressione Cron per definire la pianificazione, scegli Pianificazione basata su cron e immetti l'espressione Cron. • Per utilizzare un'espressione di frequenza per definire la pianificazione, scegli Pianificazione basata su frequenza e inserisci l'espressione di frequenza. Per ulteriori informazioni sulle espressioni cron e rate, consulta Schedule types on EventBridge Scheduler nella Amazon EventBridge Scheduler User Guide. b. Per Finestra temporale flessibile, scegli Disattiva ta per disattivare l'opzione o scegli una delle finestre temporali predefinite. Ad esempio, se scegli 15 minuti e imposti una pianificazione ricorrent e per il richiamo della	

Ricorrenza	Esegui questa operazione...	
	destinazione ogni ora, la pianificazione viene eseguita entro 15 minuti dall'inizio di ogni ora.	

5. (Facoltativo) Se hai scelto Pianificazione ricorrente nel passaggio precedente, nella sezione Intervallo di tempo effettua le seguenti operazioni:
 - a. Per Fuso orario, scegli un fuso orario.
 - b. Per Data e ora di inizio, inserisci una data valida in formato YYYY/MM/DD, quindi specifica un timestamp in formato hh:mm 24 ore.
 - c. Per Data e ora di fine, inserisci una data valida in formato YYYY/MM/DD, quindi specifica un timestamp in formato hh:mm 24 ore.
6. Scegli Next (Successivo).
7. Nella pagina Seleziona destinazione, procedi come segue:
 - a. Scegli Tutto APIs, quindi nella casella di ricerca inserisci ECS.
 - b. Seleziona Amazon ECS.
 - c. Nella casella di ricerca, immettete RunTask, quindi scegliete RunTask.
 - d. Per Cluster ECS, scegli il cluster.
 - e. Per Attività ECS, scegli la definizione dell'attività da utilizzare.
 - f. Scegli in che modo le tue attività sono distribuite nell'infrastruttura del cluster. Espandi le opzioni di elaborazione, quindi scegli una delle seguenti opzioni

Opzione di calcolo	Fasi	
Strategia del provider di capacità	a. Scegli la strategia Capacity Provider. b. Scegli una strategia: • Per utilizzare la strategia predefinita del provider di capacità, scegli Usa	

Opzione di calcolo	Fasi	
	impostazione predefinita del cluster. • Per utilizzare una strategia personalizzata, scegli Usa personalizzata. Quindi, inserisci Base, Capacity provider e Weight. Infatti EC2, il fornitore di capacità è il gruppo Auto Scaling.	
Tipo di avvio	a. Scegli il tipo di avvio. b. Per Launch type (Tipo di avvio), seleziona un tipo di avvio. c. Quando viene specificato il tipo di lancio Fargate, per Versione piattaforma, specificare la versione della piattaforma da utilizzare.	

- g. Per le sottoreti, inserite la sottorete in IDs cui eseguire l'operazione.
- h. Per Gruppi di sicurezza, immettere il gruppo di sicurezza IDs per la sottorete.
- i. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Vincolo di posizionamento, quindi inserisci i vincoli.

Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

- j. (Facoltativo) Per identificare le attività, configura i tag nella sezione Tag.

Per fare in modo che Amazon ECS assegni automaticamente i tag a tutte le attività appena avviate con i tag di definizione delle attività, seleziona **Abilita tag gestiti da Amazon ECS**.

8. Scegli **Next (Successivo)**.
9. Nella pagina **Settings (Impostazioni)**, eseguire le operazioni descritte di seguito.
 - a. Per attivare la pianificazione, in **Stato della pianificazione**, attiva **Abilita pianificazione**.
 - b. Per configurare una policy di ripetizione per la tua pianificazione, in **Policy di ripetizione e coda DLQ (Dead-Letter Queue)** effettua le seguenti operazioni:
 - Attiva/disattiva **Riprova**.
 - Per **Tempo massimo di conservazione dell'evento**, inserisci il numero massimo di ore e minuti in cui **EventBridge Scheduler** deve conservare un evento non elaborato.
 - La durata massima è 24 ore.
 - Per **Numero massimo di tentativi**, inserisci il numero massimo di volte in cui **EventBridge Scheduler** riprova la pianificazione se la destinazione restituisce un errore.

Il valore massimo è 185 tentativi.

Con le politiche di ripetizione dei tentativi, se una pianificazione non riesce a richiamare l'obiettivo, **EventBridge Scheduler** esegue nuovamente la pianificazione. Se configurato, è necessario impostare il tempo di conservazione massimo e i nuovi tentativi per la pianificazione.

- c. Scegli dove **EventBridge Scheduler** archivia gli eventi non consegnati.

Opzione Dead-letter queue (DLQ)	Esegui questa operazione...	
Non conservare	Scegliere None (Nessuno) .	
Archivia l'evento nello stesso spazio in Account AWS cui stai creando il programma	a. Scegli Seleziona una coda Amazon SQS in my Account AWS as a DLQ. b. Scegli il nome della risorsa Amazon (ARN)	

Opzione Dead-letter queue (DLQ)	Esegui questa operazione e...	
	della coda di Amazon SQS.	
Archivia l'evento in un luogo diverso Account AWS da quello in cui stai creando il programma	a. Scegli Specificare una coda Amazon SQS tra le altre Account AWS come DLQ. b. Inserisci il nome della risorsa Amazon (ARN) della coda di Amazon SQS.	

- d. Per utilizzare una chiave gestita dal cliente per crittografare l'input di destinazione, in Crittografia scegli Personalizza le impostazioni di crittografia (avanzate).

Se scegli questa opzione, inserisci l'ARN di una chiave KMS esistente scegli Crea una AWS KMS key per accedere alla console AWS KMS . Per ulteriori informazioni su come EventBridge Scheduler crittografa i dati inattivi, consulta [Encryption at rest](#) nella Amazon EventBridge Scheduler User Guide.

- e. Per Autorizzazioni, scegli Usa ruolo esistente, quindi seleziona il ruolo.

Per fare in modo che EventBridge Scheduler crei un nuovo ruolo di esecuzione per te, scegli Crea nuovo ruolo per questa pianificazione. Inserisci, quindi, un nome per Nome ruolo. Se scegli questa opzione, EventBridge Scheduler assegna al ruolo le autorizzazioni necessarie per la destinazione basata sul modello.

10. Scegli Next (Successivo).
11. Nella pagina Rivedi e crea pianificazione, rivedi i dettagli della pianificazione. In ogni sezione, scegli Modifica per tornare a tale passaggio e modificarne i dettagli.
12. Scegli Crea pianificazione.

Puoi visualizzare un elenco delle pianificazioni nuove ed esistenti nella pagina Pianificazioni. Nella colonna Stato, accertati che la nuova pianificazione sia Abilitata.

Passaggi successivi

È possibile utilizzare la console EventBridge Scheduler o gestire la pianificazione. AWS CLI Per ulteriori informazioni, consulta [Managing a planning](#) nella Amazon EventBridge Scheduler User Guide.

Interruzione di un'attività Amazon ECS

Se non hai più bisogno di mantenere in esecuzione un'attività autonoma, puoi interromperla. La console Amazon ECS semplifica l'interruzione di una o più attività.

Non è possibile riavviare un'attività interrotta autonoma.

Se desideri interrompere un servizio, consulta [Eliminazione di un servizio Amazon ECS tramite la console](#).

Per interrompere un'attività autonoma (AWS Management Console)

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Cluster, scegli il cluster per accedere alla pagina dei dettagli del cluster.
4. Nella pagina dei dettagli del cluster, scegli la scheda Attività.
5. Puoi filtrare le attività per tipo di avvio utilizzando l'elenco Filtra tipo di avvio.

Attività da interrompere	Fasi	
Una o più	a. Seleziona le attività, quindi scegli Stop, Stop selected. b. Nella pagina di conferma dell'interruzione dell'attività, scegli Interrompi	
Tutti	 Important Se scegli di interrompere tutte le attività utilizzando la console, Amazon	

Attività da interrompere	Fasi	
	ECS interrompe tutte le attività autonome e le attività che fanno parte di un servizio. Pertanto, raccomandiamo cautela quando si utilizza questa opzione. a. Seleziona Interrompi, Interrompi tutto. b. Nella pagina di conferma dell'interruzione dell'attività, inserisci Interrompi tutte le attività, quindi scegli Interrompi.	

Servizi Amazon ECS

Puoi utilizzare un servizio Amazon ECS per eseguire e mantenere simultaneamente un determinato numero di istanze di una definizione di attività in un cluster Amazon ECS. Se una delle tue attività non riesce o si interrompe, il pianificatore del servizio Amazon ECS avvia un'altra istanza della definizione di attività per sostituirla. Ciò consente di mantenere il numero desiderato di attività nel servizio.

A scelta, puoi eseguire il tuo servizio con un load balancer. Il load balancer distribuisce il traffico nelle attività associate al servizio.

Ti consigliamo di utilizzare il pianificatore di servizi per le applicazioni e i servizi stateless di lunga durata. Il pianificatore di servizi garantisce che venga seguita la strategia di pianificazione specificata e ripianifica i processi qualora un processo dia esito negativo. Ad esempio, se l'infrastruttura sottostante riporta un errore, il pianificatore del servizio può riprogrammare un'attività. Puoi applicare vincoli e strategie di posizionamento delle attività per personalizzare il modo in cui il pianificatore posiziona e termina le attività. Se un processo in un servizio viene arrestato, il pianificatore avvia un nuovo processo per sostituirlo. Questo processo continua finché il servizio non raggiunge il numero

di attività desiderato in base alla strategia di pianificazione utilizzata dal servizio. La strategia di pianificazione del servizio è indicata anche come tipo di servizio.

Il pianificatore di servizi sostituisce inoltre le attività ritenute non integre dopo l'esito negativo di un controllo dell'integrità del container o di un sistema di bilanciamento del carico del gruppo di destinazione. Questa sostituzione dipende dai parametri di definizione del servizio `maximumPercent` e `desiredCount`. Se un'attività è contrassegnata come non integra, il pianificatore di servizi avvierà innanzitutto un'attività di sostituzione. Quindi, accade quanto segue.

- Se lo stato di integrità dell'attività sostitutiva è pari a `HEALTHY`, l'utilità di pianificazione del servizio interrompe l'attività non integra
- Se lo stato di integrità dell'attività di sostituzione è `UNHEALTHY`, il pianificatore interromperà l'attività di sostituzione non integra o l'attività esistente non integra per far sì che il numero totale delle attività sia pari a `desiredCount`.

Se il parametro `maximumPercent` impedisce al pianificatore di avviare un'attività di sostituzione, il pianificatore interromperà un'attività non integra alla volta, in modo casuale, per liberare spazio, e poi avvierà un'attività di sostituzione. Il processo di avvio e arresto continua fino a quando tutte le attività non integre vengono sostituite con attività integre. Dopo aver sostituito tutte le attività non integre e aver avviato solo quelle integre, se il numero totale delle attività supera `desiredCount`, le attività integre vengono interrotte casualmente fino a quando il numero totale delle attività è pari a `desiredCount`. Per ulteriori informazioni sui parametri `maximumPercent` e `desiredCount`, consulta [Parametri di definizione del servizio](#).

Il pianificatore del servizio include una logica che limita la frequenza dei tentativi di riavvio delle attività se queste non riescono a riavviarsi ripetutamente. Se un'attività si interrompe senza essere entrata in uno stato `RUNNING`, il pianificatore del servizio inizia a rallentare in modo incrementale i tentativi di avvio ed emette un messaggio di evento del servizio. Questo comportamento previene l'utilizzo di risorse non necessarie per le attività non riuscite prima che sia risolto il problema. Dopo l'aggiornamento del servizio, il pianificatore del servizio riprende il normale comportamento. Per ulteriori informazioni, consultare [Logica di accelerazione del servizio Amazon ECS](#) e [Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS](#).

Sono disponibili due strategie del pianificatore del servizio:

- **REPLICA**: la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le

decisione riguardo al posizionamento delle attività. Per ulteriori informazioni, consulta [Strategia di replica](#).

- **DAEMON:** la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Quando si utilizza questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta [Strategia daemon](#).

Note

I processi Fargate non supportano la strategia di pianificazione DAEMON

Strategia daemon

La strategia di pianificazione daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Il service scheduler valuta i vincoli di posizionamento delle attività per le attività in esecuzione e interrompe le attività che non soddisfano i vincoli di posizionamento. Quando si utilizza questa strategia, non è necessario specificare il numero desiderato di attività, una strategia di posizionamento delle attività o utilizzare le politiche di Service Auto Scaling.

Amazon ECS riserva risorse di calcolo dell'istanza di container tra cui CPU, memoria e interfacce di rete per i processi del daemon. Quando avvii un servizio daemon in un cluster con altri servizi di replica, Amazon ECS assegna la priorità all'attività del daemon. Ciò significa che l'attività daemon è la prima attività da avviare sulle istanze e l'ultima attività da interrompere dopo l'interruzione di tutte le attività di replica. La strategia garantisce che le risorse non vengano utilizzate da attività di replica in sospeso e che siano disponibili per le attività del daemon. Tieni presente che a volte i servizi di strategia daemon non possono essere avviati perché il servizio di strategia di replica si avvia più velocemente e le risorse di CPU e memoria non sono disponibili per il servizio strategico.

Il pianificatore del servizio del daemon non posiziona alcuna attività sulle istanze con stato DRAINING. Se un'istanza di container passa a uno stato DRAINING, le attività daemon in esecuzione su di essa vengono arrestate. Il pianificatore del servizio inoltre monitora l'aggiunta di nuove istanze di container al cluster, alle quali aggiunge le attività daemon.

Quando si specifica una configurazione di distribuzione, il valore del `maximumPercent` parametro deve essere 100 (specificato in percentuale), che è il valore predefinito utilizzato se non impostato. Il valore predefinito per il `minimumHealthyPercent` parametro è 0 (specificato come percentuale).

Quando modifichi i vincoli di posizionamento per il servizio daemon devi riavviare il servizio. Amazon ECS aggiorna dinamicamente le risorse riservate sulle istanze idonee per l'attività del daemon. Per le istanze esistenti, il pianificatore prova a posizionare l'attività sull'istanza.

Una modifica della dimensione del processo o della prenotazione della risorsa del container nella definizione di attività avvia una nuova implementazione. Una nuova distribuzione inizia anche quando si aggiorna un servizio o si imposta una revisione diversa della definizione dell'attività. Amazon ECS raccoglie la CPU e le prenotazioni di memoria aggiornate per il daemon e quindi blocca tale capacità per il processo del daemon.

Se ci sono risorse insufficienti per uno dei casi precedenti, si verifica quanto segue:

- Il posizionamento del processo ha esito negativo.
- Viene generato un CloudWatch evento.
- Amazon ECS continua a provare a pianificare il processo sull'istanza attendendo che le risorse diventino disponibili.
- Amazon ECS libera tutte le istanze riservate che non soddisfano più i criteri di vincolo di posizionamento e interrompe i processi del daemon corrispondenti.

La strategia di pianificazione del daemon può essere utilizzata nei seguenti casi:

- Esecuzione di container di applicazioni
- Esecuzione di container di supporto per processi di registrazione, monitoraggio e traccia

Le attività che utilizzano il tipo di avvio Fargate oppure i tipi di controller di implementazione `CODE_DEPLOY` o `EXTERNAL` non supportano la strategia di pianificazione del daemon.

Quando il pianificatore di servizi arresta i processi in esecuzione, prova a mantenere il bilanciamento tra le zone di disponibilità nel cluster. Il pianificatore utilizza la logica seguente:

- Se è stata definita una strategia di posizionamento, utilizzare tale strategia per selezionare le attività da terminare. Ad esempio, se per un servizio è definita una strategia di distribuzione tra zone di disponibilità, viene selezionata un'attività che lascia le attività rimanenti con la migliore distribuzione.

- Se non è stata definita una strategia di posizionamento, mantieni il bilanciamento tra le zone di disponibilità nel cluster con la logica seguente:
 - Ordina le istanze di container valide. Dai priorità alle istanze che hanno il maggior numero di attività in esecuzione per questo servizio nella rispettiva zona di disponibilità. Ad esempio, se nella zona A è presente un'attività del servizio in esecuzione e nelle zone B e C ne sono presenti due, le istanze di container nella zona B o C sono considerate ottimali per l'arresto.
 - Arresta l'attività del servizio in un'istanza di container in una zona di disponibilità ottimale in base alle fasi precedenti. Dai la priorità alle istanze di container con il maggior numero di attività in esecuzione per questo servizio.

Strategia di replica

La strategia di pianificazione delle repliche posiziona e gestisce il numero desiderato di attività nel cluster.

Per un servizio che esegue attività su Fargate, quando il pianificatore di servizi avvia nuove attività o arresta quelle in esecuzione, il pianificatore fa del suo meglio per mantenere il bilanciamento tra le zone di disponibilità. Non è necessario specificare strategie di posizionamento delle attività o vincoli.

Quando crei un servizio che esegue attività su EC2 istanze, puoi facoltativamente specificare strategie e vincoli di posizionamento delle attività per personalizzare le decisioni di posizionamento delle attività. Se non vengono specificati vincoli o strategie di posizionamento delle attività, per impostazione predefinita il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Il pianificatore di servizi utilizza la logica seguente:

- Determina quale delle istanze di container nel cluster può supportare la definizione di attività del servizio (in termini, ad esempio, di CPU, memoria, porte e attributi dell'istanza di container necessari).
- Determina quali istanze di container soddisfano gli eventuali vincoli di posizionamento definiti per il servizio.
- Se disponi di un servizio di replica che dipende da un servizio daemon (ad esempio, un'attività del daemon log router che deve essere eseguita prima che le attività possano utilizzare la registrazione), crea un vincolo di posizionamento delle attività che assicuri che le attività del servizio daemon vengano posizionate sull' EC2 istanza prima delle attività del servizio di replica. Per ulteriori informazioni, consulta [Esempio di vincoli di posizionamento delle attività di Amazon ECS](#).

- Se hai definito una strategia di posizionamento, utilizzala per selezionare un'istanza tra le candidate rimanenti.
- Se non è stata definita una strategia di posizionamento, bilancia le attività tra le zone di disponibilità nel cluster con la logica seguente:
 - Ordina le istanze di container valide. Dai priorità alle istanze che hanno il minor numero di attività in esecuzione per questo servizio nella rispettiva zona di disponibilità. Ad esempio, se nella zona A è presente un'attività del servizio in esecuzione e nelle zone B e C nessuna, le istanze di container valide nella zona B o C sono considerate ottimali per il posizionamento.
 - Posiziona la nuova attività del servizio in un'istanza di container valida in una zona di disponibilità ottimale in base alle fasi precedenti. Dai la priorità alle istanze di container con il minor numero di attività in esecuzione per questo servizio.

Ti consigliamo di utilizzare la funzionalità di ribilanciamento del servizio quando utilizzi la REPLICA strategia perché aiuta a garantire un'elevata disponibilità del servizio.

Bilanciamento di un servizio Amazon ECS tra le zone di disponibilità

Per aiutare le tue applicazioni a raggiungere un'elevata disponibilità, ti consigliamo di configurare i tuoi servizi multi-task in modo che vengano eseguiti su più zone di disponibilità. Per i servizi che prevedono come prima strategia di collocamento la distribuzione in zone di disponibilità, AWS si adopera al massimo per distribuire in modo uniforme le attività di servizio tra le zone di disponibilità disponibili. Tuttavia, a volte il numero di attività in esecuzione in una zona di disponibilità potrebbe non essere lo stesso di quello in altre zone di disponibilità, ad esempio dopo un'interruzione della zona di disponibilità. Per risolvere questo squilibrio tra le attività, è possibile abilitare la funzionalità di ribilanciamento delle zone di disponibilità. Con il ribilanciamento delle zone di disponibilità, Amazon ECS monitora continuamente la distribuzione delle attività tra le zone di disponibilità per ciascuno dei tuoi servizi. Quando Amazon ECS rileva una distribuzione irregolare delle attività, interviene automaticamente per riequilibrare il carico di lavoro tra le zone di disponibilità. Ciò comporta l'avvio di nuove attività nelle zone di disponibilità con il minor numero di attività e l'interruzione delle attività nelle zone di disponibilità sovraccariche. Questa ridistribuzione garantisce che nessuna singola zona di disponibilità diventi un punto di errore, contribuendo a mantenere la disponibilità complessiva delle applicazioni containerizzate. Il processo di ribilanciamento automatizzato elimina la necessità di un intervento manuale, accelerando i tempi di ripristino dopo un evento.

Di seguito è riportata una panoramica del processo di ribilanciamento della zona di disponibilità:

1. Amazon ECS inizia a monitorare un servizio dopo che ha raggiunto lo stato stazionario e analizza il numero di attività in esecuzione in ciascuna zona di disponibilità.
2. Amazon ECS esegue le seguenti operazioni quando rileva uno squilibrio nel numero di attività in esecuzione in ciascuna zona di disponibilità:
 - Invia un evento di servizio che indica che il ribilanciamento della zona di disponibilità è in corso.
 - Avvia le attività nelle zone di disponibilità con il minor numero di attività in esecuzione
 - Interrompe le attività nelle zone di disponibilità con il maggior numero di attività in esecuzione.
 - Lo scheduler attende che le attività appena avviate siano terminate HEALTHY e RUNNING prima di interromperle nella zona di disponibilità sovradimensionata.
 - Invia un evento di servizio con l'esito del ribilanciamento della zona di disponibilità.

Il ribilanciamento della zona di disponibilità supporta i tipi Fargate e Launch. EC2 Per Fargate, Amazon ECS ridistribuirà automaticamente le attività tra le zone di disponibilità disponibili per mantenere l'equilibrio. Per il tipo di EC2 lancio, Amazon ECS ribilancia le attività tra le istanze di container esistenti nel miglior modo possibile, rispettando le strategie e i vincoli di posizionamento definiti. Tuttavia, Amazon ECS non può lanciare nuove istanze in zone di disponibilità sottoutilizzate come parte del processo di ribilanciamento, limitando il ribilanciamento alle istanze di container esistenti.

Il ribilanciamento delle zone di disponibilità funziona nelle seguenti configurazioni:

- Servizi che utilizzano la strategia Replica
- I servizi che specificano la zona di disponibilità si diffondono come prima strategia di collocamento delle attività oppure non specificano una strategia di posizionamento.

Non è possibile utilizzare il ribilanciamento della zona di disponibilità con servizi che soddisfano uno dei seguenti criteri:

- Utilizza la strategia Daemon
- Utilizza il tipo di EXTERNAL avvio (ECS Anywhere)
- Utilizza il 100% per il valore maximumPercent
- Utilizza un Classic Load Balancer
- Utilizza `attribute:ecs.availability-zone` come vincolo di posizionamento delle attività

Strategie di posizionamento e vincoli di posizionamento con il ribilanciamento delle zone di disponibilità

Le strategie di posizionamento determinano il modo in cui Amazon ECS seleziona le istanze di container e le zone di disponibilità per l'interruzione delle attività. I vincoli di posizionamento delle attività sono regole che determinano se un'attività può essere eseguita su una specifica istanza del contenitore. Per il tipo di EC2 avvio, è possibile utilizzare strategie di posizionamento e vincoli di posizionamento insieme al ribilanciamento della zona di disponibilità. Tuttavia, affinché il ribilanciamento delle zone di disponibilità funzioni, la strategia di posizionamento degli spread nelle zone di disponibilità deve essere la prima strategia specificata. Il ribilanciamento delle zone di disponibilità è compatibile con varie combinazioni di strategie di posizionamento. Ad esempio, è possibile creare una strategia che distribuisca prima le attività in modo uniforme tra le zone di disponibilità e quindi comprime le attività in base alla memoria all'interno di ciascuna zona di disponibilità. In questo caso, il ribilanciamento delle zone di disponibilità funziona perché la strategia di diffusione delle zone di disponibilità viene specificata per prima. È importante notare che il ribilanciamento delle zone di disponibilità non funzionerà se la prima strategia nell'array della strategia di posizionamento non è un componente di diffusione delle zone di disponibilità. Questo requisito garantisce che l'obiettivo principale della distribuzione delle attività sia il mantenimento dell'equilibrio tra le zone di disponibilità, fondamentale per un'elevata disponibilità. Per ulteriori informazioni sulle strategie e sui vincoli di collocamento delle attività, vedere. [In che modo Amazon ECS colloca le attività sulle istanze di container](#)

Per processi con tipo di avvio Fargate, le strategie e i vincoli di posizionamento dei processi non sono supportati. Fargate farà del suo meglio per distribuire le attività tra zone di disponibilità accessibili. Se il provider di capacità include sia Fargate che Fargate Spot, il comportamento di distribuzione sarà indipendente per ogni provider.

La strategia di esempio seguente distribuisce le attività in modo uniforme tra le zone di disponibilità, quindi raggruppa le attività in base alla memoria all'interno di ciascuna zona di disponibilità. Il ribilanciamento delle zone di disponibilità è compatibile con il servizio perché la spread strategia è la prima.

```
"placementStrategy": [  
  {  
    "field": "attribute:ecs.availability-zone",  
    "type": "spread"  
  },  
  {  
    "field": "memory",
```

```

    "type": "binpack"
  }
]
```

Attiva il ribilanciamento della zona di disponibilità

È necessario abilitare il ribilanciamento della zona di disponibilità per i servizi nuovi ed esistenti.

È possibile abilitare e disabilitare il ribilanciamento della zona di disponibilità utilizzando la console o il APIs. AWS CLI

Tipo di servizio	API	Console	CLI
Esistente	UpdateService	Aggiornamento di un servizio Amazon ECS tramite la console	servizio di aggiornamento
Novità	CreateService	Creazione di un servizio Amazon ECS utilizzando la console	create-service

Tieni traccia del ribilanciamento della zona di disponibilità di Amazon ECS

Puoi verificare se il ribilanciamento della zona di disponibilità è abilitato per un servizio nella console o chiamando. `describe-services` L'esempio seguente può essere utilizzato per visualizzare lo stato con la CLI.

La risposta sarà `ENABLED` o `DISABLED`.

```
aws ecs describe-services \
  --services service-name \
  --cluster cluster-name \
  --query services[0].availabilityZoneRebalancing
```

Eventi di servizio

Amazon ECS invia eventi di azione del servizio per aiutarti a comprendere il ciclo di vita del ribilanciamento della zona di disponibilità.

Evento	Scenario	Tipo	Ulteriori informazioni
SERVICE_REBALANCING_STARTED	Amazon ECS avvia un'operazione di ribilanciamento della zona di disponibilità	INFO	service (service-name) non è bilanciato in AZ con number-tasks le attività number-tasks in Availability Zone 1Availability Zone 2 , in e number-tasks inAvailability Zone 3. Ribilanciamento AZ in corso.
SERVICE_REBALANCING_COMPLETED	L'operazione di ribilanciamento della zona di disponibilità è completata	INFO	service (service-name) è bilanciato da AZ con number-tasks number-tasks task in Availability Zone 1Availability Zone 2 ,tasks in e number-tasks tasks inAvailability Zone 3 .
TASKS_STARTED	Amazon ECS avvia correttamente le attività nell'ambito dell'operazione di ribilanciamento della zona di disponibilità	INFO	service-name ha avviato number-tasks attività in Availability Zone AZ Rebalance:. task-ids
TASKS_STOPPED	Amazon ECS interrompe correttamente le attività	INFO	service-name ha interrotto l'number-tasks esecuzione

Evento	Scenario	Tipo	Ulteriori informazioni
	nell'ambito dell'operazione di ribilanciamento della zona di disponibilità		e delle attività a <i>Availability Zone</i> causa del ribilanciamento AZ: <i>task-id</i>
SERVICE_TASK_PLACE MENT_FAILURE	Amazon ECS non è riuscito ad avviare un'attività nell'ambito dell'operazione di ribilanciamento della zona di disponibilità	ERRORE	Per il tipo di EC2 avvio, vedi service (<i>service-name</i>) non è in grado di inserire un'attività Availability Zone perché nessuna istanza del contenuto re soddisfa tutti i requisiti. Per il tipo di lancio di Fargate, vedi service (<i>service-name</i>) non è in grado di inserire un'attività in Availability Zone.
TASKSET_SCALE_IN_FAILURE_BY _TASK_PROTECTION	L'operazione di ribilanciamento della zona di disponibilità è bloccata perché è in uso la protezione delle attività.	INFO	service (<i>service-name</i>) non è riuscito a eseguire AZ Rebalance perché non task-set-name è riuscito a scalare a causa di reason

Evento	Scenario	Tipo	Ulteriori informazioni
SERVICE_REBALANCING_STOPPED	L'operazione di ribilanciamento della zona di disponibilità è stata interrotta. Amazon ECS invia eventi aggiuntivi che forniscono ulteriori informazioni.	INFO	service (<i>service-name</i>) ha interrotto AZ Rebalancing.

Eventi di modifica dello stato dei processi

Amazon ECS invia un evento di modifica dello stato dell'attività (START) per ogni attività avviata come parte del processo di ribilanciamento.

Amazon ECS invia un evento di modifica dello stato dell'attività (STOPPED) per ogni attività che interrompe come parte del processo di ribilanciamento. Il motivo è impostato su `Availability-zone rebalancing initiated by (deployment ecs-svc/deployment-id)`

Per ulteriori informazioni sugli eventi, vedere [Eventi di modifica dello stato delle attività di Amazon ECS](#).

Creazione di un servizio Amazon ECS utilizzando la console

Crea un servizio per eseguire e gestire contemporaneamente un numero specifico di istanze di una definizione di attività in un cluster. Se una delle tue attività non riesce o si interrompe, il pianificatore del servizio Amazon ECS avvia un'altra istanza della definizione di attività per sostituirla. Ciò consente di mantenere il numero desiderato di attività nel servizio.

Decidi i seguenti parametri di configurazione prima di creare un servizio:

- Esistono due opzioni di calcolo che distribuiscono le attività.
 - Una strategia per provider di capacità fa sì che Amazon ECS distribuisca le attività in uno o più provider di capacità.
 - Un tipo di avvio fa sì che Amazon ECS avvii le nostre attività direttamente su Fargate o sulle istanze registrate EC2 nei tuoi cluster.

- Le definizioni di processo che utilizzano la modalità di rete `awsvpc` o i servizi configurati per l'utilizzo di un load balancer devono disporre di una configurazione di rete. Di default, la console seleziona l'Amazon VPC di default insieme a tutte le sottoreti e il gruppo di sicurezza di default all'interno dell'Amazon VPC di default.
- La strategia di posizionamento, La strategia di posizionamento delle attività predefinita distribuisce le attività in modo uniforme tra le zone di disponibilità.

Ti consigliamo di utilizzare il ribilanciamento delle zone di disponibilità per garantire un'elevata disponibilità del servizio. Per ulteriori informazioni, consulta [Bilanciamento di un servizio Amazon ECS tra le zone di disponibilità](#).

- Quando utilizzi il tipo di avvio per l'implementazione del servizio, per impostazione predefinita il servizio viene avviato nelle sottoreti del cluster VPC.
- Per la strategia per provider di capacità, la console seleziona un'opzione di calcolo di default. Di seguito viene descritto l'ordine utilizzato dalla console per selezionare un valore di default:
 - Se il cluster dispone di una strategia di provider di capacità definita, questa è selezionata.
 - Se nel cluster non è stata definita una strategia predefinita per i provider di capacità, ma al cluster sono stati aggiunti i provider di capacità Fargate, viene selezionata una strategia personalizzata per i provider di capacità che utilizza il provider di FARGATE capacità.
 - Se nel cluster non è stata definita una strategia predefinita per i provider di capacità, ma al cluster sono stati aggiunti uno o più provider di capacità di gruppo Auto Scaling, viene selezionata l'opzione Usa personalizzato (avanzato) ed è necessario definire manualmente la strategia.
 - Se nel cluster non è presente una strategia di provider di capacità di default definita e non sono stati aggiunti provider di capacità al cluster, è selezionato il tipo di avvio Fargate.
- Le opzioni predefinite di rilevamento degli errori di distribuzione prevedono l'utilizzo dell'opzione di interruzione del circuito di distribuzione di Amazon ECS con l'opzione Rollback on failures.

Per ulteriori informazioni, consulta [In che modo l'interruttore di distribuzione di Amazon ECS rileva i guasti](#).

- Se desideri utilizzare l'opzione di distribuzione blu/verde, determina come si spostano le applicazioni. CodeDeploy Sono disponibili le seguenti opzioni:
 - CodeDeployDefault. ECSAllAtOnce: sposta tutto il traffico verso il contenitore Amazon ECS aggiornato contemporaneamente
 - CodeDeployDefault. ECSLinear10 PercentEvery 1Minuti: Sposta il 10% del traffico ogni minuto fino a quando tutto il traffico non viene spostato.

- CodeDeployDefault. ECSLinear10PercentEvery3Minutes: sposta il 10 per cento del traffico ogni 3 minuti fino a quando tutto il traffico non viene spostato.
- CodeDeployDefault. ECSCanary10Percent5Minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato cinque minuti più tardi.
- CodeDeployDefault. ECSCanary10Percent15Minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato 15 minuti più tardi.
- Decidi se desideri che Amazon ECS aumenti o diminuisca automaticamente il numero desiderato di attività nel tuo servizio. Per informazioni, consulta [Ridimensiona automaticamente il tuo servizio Amazon ECS](#).
- Se hai bisogno di un'applicazione per connetterti ad altre applicazioni in esecuzione su Amazon ECS, determina l'opzione più adatta alla tua architettura. Per ulteriori informazioni, consulta [Interconnetti i servizi Amazon ECS](#).
- Quando crei un servizio che utilizza Amazon ECS circuit breaker, Amazon ECS crea una distribuzione e una revisione del servizio. Queste risorse consentono di visualizzare informazioni dettagliate sulla cronologia dei servizi. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS](#).

Per informazioni su come creare un servizio utilizzando il AWS CLI, vedere [create-service](#) nel AWS Command Line Interface riferimento.

Per informazioni su come creare un servizio utilizzando AWS CloudFormation, vedere [AWS::ECS::Service](#) nella Guida per l'utente di AWS CloudFormation .

Creare un servizio con le opzioni predefinite

Puoi utilizzare la console per creare e implementare rapidamente un servizio. Il servizio ha la seguente configurazione:

- Si implementa nel VPC e nelle sottoreti associate al cluster
- Implementa un'attività
- Utilizza l'implementazione in sequenza
- Utilizza la strategia del provider di capacità con il tuo provider di capacità predefinito
- Utilizza l'interruttore automatico di implementazione per rilevare i guasti e imposta l'opzione per ripristinare automaticamente l'implementazione in caso di errore

Per implementare un servizio utilizzando i parametri predefiniti, completa la seguente procedura.

Per creare un servizio (console Amazon ECS)

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella pagina di navigazione, scegli Cluster.
3. Nella pagina Cluster, scegli il cluster in cui creare il servizio.
4. Nella scheda Services (Servizi), scegli Create (Crea).

Viene visualizzata la pagina Crea servizio.

5. In Dettagli del servizio, procedi come segue:
 - a. Per Definizione dell'attività, immettete la famiglia di definizioni delle attività e la revisione da utilizzare.
 - b. In Service name (Nome servizio), specifica un nome per il servizio.
6. In Configurazione di distribuzione, procedi come segue:
 - Per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e gestire nel servizio.
7. (Facoltativo) Per identificare il servizio e le attività, espandi la sezione Tags (Tag), quindi configura i tag.

Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag di definizione delle attività, seleziona Turn on Amazon ECS managed tags (Attiva i tag gestiti da Amazon ECS), quindi seleziona Task definitions (Definizioni di attività).

Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag del servizio, seleziona Turn on Amazon ECS managed tags (Attiva i tag gestiti da Amazon ECS), quindi seleziona Service (Servizio).

Aggiungi o rimuovi un tag.

- [Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
- [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).

Creazione di un servizio utilizzando parametri definiti

Per creare un servizio utilizzando parametri definiti, segui questi passaggi.

Per creare un servizio (console Amazon ECS)

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Determina la risorsa da cui avviare il servizio.

Per avviare un servizio da	Fasi	
Cluster	a. Nella pagina Cluster, seleziona il cluster in cui creare il servizio. b. Nella scheda Services (Servizi), scegli Create (Crea).	
Definizione di attività	a. Nella pagina Definizioni delle attività, seleziona il pulsante di opzione accanto alla definizione dell'attività. b. Nel menu Distribuisci, scegli Crea servizio.	

Viene visualizzata la pagina Crea servizio.

3. In Dettagli del servizio, procedi come segue:
 - a. In Definizione dell'attività, immettere la definizione dell'attività da utilizzare. Quindi, per Revisione, scegli la revisione da utilizzare.
 - b. In Service name (Nome servizio), specifica un nome per il servizio.
4. Per Cluster esistente, scegli il cluster.

Scegli Crea cluster per eseguire l'attività su un nuovo cluster

5. Scegli in che modo le tue attività sono distribuite nell'infrastruttura del cluster. In Configurazione di elaborazione, scegli la tua opzione.

Opzione di calcolo	Fasi	
Strategia del provider di capacità	a. In Opzioni di calcolo, scegli Strategia del provider di capacità. b. Scegli una strategia: • Per utilizzare una strategia del provider di capacità predefinita del cluster, scegli Use cluster default (Usa impostazione predefinita del cluster). • Se il cluster non dispone di una strategia del provider di capacità predefinita o per utilizzare una strategia personalizzata, scegli Usa personalizzato, Aggiungi strategia del provider di capacità e definisci la strategia personalizzata specificando Base, Provider di capacità e Peso.	
 Note Per utilizzare un provider di capacità in una strategia , il provider di		

Opzione di calcolo	Fasi	
	capacità deve essere associato al cluster.	
Tipo di avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Launch type (Tipo di avvio), seleziona un tipo di avvio. c. (Facoltativo) Quando viene specificato il tipo di avvio Fargate, per Versione piattaforma specifica la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

6. In Configurazione di distribuzione, procedi come segue:

- a. Per Service type (Tipo di servizio), scegli la strategia di pianificazione del servizio.
 - Perché lo scheduler implementi esattamente una attività su ciascuna istanza di container che risponda a tutti i vincoli di posizionamento dell'attività, scegli Daemon.
 - Perché lo scheduler posizioni e mantenga il numero di attività desiderato nel cluster, scegli Replica.
- b. Se hai scelto Replica, per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e mantenere nel servizio.
- c. Se hai scelto Replica, per consentire ad Amazon ECS di monitorare la distribuzione delle attività tra le zone di disponibilità e ridistribuirle in caso di squilibrio, in Availability Zone service rebalancing, seleziona Availability Zone service rebalancing.

- d. Per il periodo di tolleranza del controllo dell'integrità, inserisci la quantità di tempo (in secondi) per inserire la quantità di tempo (in secondi) in cui lo scheduler del servizio ignora i controlli non integri di Elastic Load Balancing, VPC Lattice e dello stato dei container dopo il primo avvio di un'attività. Se non si specifica un valore del periodo di tolleranza per il controllo dello stato di salute, viene utilizzato il valore predefinito 0.
- e. Determina il tipo di implementazione per il servizio. Espandi le opzioni di distribuzione, quindi specifica i seguenti parametri.

Il tipo di distribuzione	Fasi	
Aggiornamento in sequenza	a. Per Min running tasks (Numero minimo di attività in esecuzione), specifica il limite inferiore per il numero di attività nel servizio che devono rimanere nello stato RUNNING durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Per ulteriori informazioni, consulta Configurazione dell'implementazione. b. Per Max running tasks (Numero massimo di attività in esecuzione), specifica il limite superiore per il numero di attività del servizio consentite nello stato RUNNING o PENDING durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per difetto al numero intero più vicino).	

Il tipo di distribuzione	Fasi	
Implementazione blu/verde	a. Per la configurazione della distribuzione, scegliete come CodeDeploy indirizzare il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione. b. Per Service role for CodeDeploy, scegli il ruolo IAM utilizzato dal servizio per autorizzare le richieste API Servizi AWS.	

- f. Per configurare il modo in cui Amazon ECS rileva e gestisce gli errori di implementazione, espandi Deployment failure detection (Rilevamento degli errori di implementazione), quindi scegli le tue opzioni.
- i. Per interrompere un'implementazione quando le attività non possono essere avviate, seleziona Use the Amazon ECS deployment circuit breaker (Usa l'interruttore automatico di implementazione di Amazon ECS).

Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di implementazione completato quando l'interruttore di distribuzione imposta la distribuzione su uno stato fallito, seleziona Rollback on failures.

- ii. Per interrompere una distribuzione in base alle metriche dell'applicazione, seleziona Usa CloudWatch allarmi. Quindi, dal nome CloudWatch dell'allarme, scegli gli allarmi. Per creare un nuovo allarme, vai alla CloudWatch console.

Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di distribuzione completato quando un CloudWatch allarme imposta la distribuzione su uno stato fallito, seleziona Rollback in caso di errori.

7. Se la definizione dell'attività utilizza la modalità di `aws_vpc` rete, è possibile specificare una configurazione di rete personalizzata, espandere Networking, quindi eseguire le seguenti operazioni per.
 - a. Per VPC seleziona il VPC da utilizzare.
 - b. Per Subnets (Sottoreti), seleziona una o più sottoreti nel VPC che lo scheduler di attività deve prendere in considerazione quando posiziona le attività.

 Important

Per la modalità di rete `aws_vpc` sono supportate solo sottoreti private. I processi non ricevono indirizzi IP pubblici. Pertanto, è necessario un gateway NAT per l'accesso a Internet in uscita e il traffico Internet in entrata viene instradato attraverso un load balancer.

- c. Per Gruppi di sicurezza è possibile selezionare un gruppo di sicurezza esistente o crearne uno nuovo. Per utilizzare un gruppo di sicurezza esistente, seleziona il gruppo di sicurezza e passa alla fase successiva. Per creare un nuovo gruppo di sicurezza, scegliere Create a new security group (Crea un nuovo gruppo di sicurezza). È necessario specificare un nome e una descrizione del gruppo di sicurezza e aggiungere una o più regole in entrata per il gruppo di sicurezza.
- d. Per IP pubblico scegli se assegnare automaticamente un indirizzo IP pubblico all'interfaccia di rete elastica (ENI) del processo stesso.

AWS Fargate alle attività può essere assegnato un indirizzo IP pubblico quando vengono eseguite in una sottorete pubblica in modo che abbiano un percorso verso Internet. EC2 alle attività non può essere assegnato un IP pubblico utilizzando questo campo. Per ulteriori informazioni, consulta le [opzioni di rete delle attività di Amazon ECS per il tipo di lancio Fargate e Allocazione di un'interfaccia di rete per un'attività Amazon ECS](#).

8. (Facoltativo) Per interconnettere il servizio tramite Service Connect, espandi Service Connect, quindi specifica quanto segue:
 - a. Seleziona Attiva Service Connect.
 - b. In Service Connect configuration (Configurazione Service Connect), specifica la modalità client.

- Se il tuo servizio esegue un'applicazione client di rete che deve solo connettersi ad altri servizi nel namespace, scegli Solo lato client.
 - Se il servizio esegue un'applicazione di rete o di servizio Web, deve fornire endpoint per questo servizio e si connette ad altri servizi nello spazio dei nomi, scegli Client and server (Client e server).
- c. Per utilizzare uno spazio dei nomi differente da quello del cluster predefinito, per Namespace (Spazio dei nomi), scegli lo spazio dei nomi del servizio.
- d. (Facoltativo) Specificate una configurazione di registro. Seleziona Usa la raccolta dei registri. L'opzione predefinita invia i log dei contenitori a CloudWatch Logs. Le altre opzioni del driver di registro sono configurate utilizzando. AWS FireLens Per ulteriori informazioni, consulta [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

Di seguito sono riportate descrizioni più dettagliate per ogni destinazione di log di container.

- Amazon CloudWatch: configura l'attività per inviare i log dei container a CloudWatch Logs. Vengono fornite le opzioni predefinite dei driver di registro, che creano un gruppo di CloudWatch log per tuo conto. Per specificare un nome del gruppo di log diverso, modifica i valori dell'opzione del driver.
 - Amazon Data Firehose: configura l'attività per inviare i log dei container a Firehose. Vengono fornite le opzioni predefinite del driver di registro, che inviano i log a un flusso di distribuzione Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
 - Amazon Kinesis Data Streams: configura l'attività per inviare i log dei container a Kinesis Data Streams. Vengono fornite le opzioni predefinite del driver di registro, che inviano i log a un flusso Kinesis Data Streams. Per specificare un nome del flusso diverso, modifica i valori dell'opzione del driver.
 - Amazon OpenSearch Service: configura l'attività per inviare i log dei container a un dominio OpenSearch di servizio. Devono essere fornite le opzioni del driver di log.
 - Amazon S3: configura l'attività per inviare i log dei container a un bucket Amazon S3. Vengono fornite le opzioni di driver di registro predefinite, ma è necessario specificare un nome di bucket Amazon S3 valido.
9. (Facoltativo) Per interconnettere il servizio tramite Service Discovery, espandi Service discovery, quindi procedi come segue.
- a. Seleziona Usa service discovery.

- b. Per utilizzare un nuovo spazio dei nomi, scegli Crea un nuovo spazio dei nomi in Configura lo spazio dei nomi, quindi fornisci un nome e una descrizione del namespace. Per utilizzare uno spazio dei nomi esistente, scegli Seleziona uno spazio dei nomi esistente, quindi scegli lo spazio dei nomi che desideri utilizzare.
- c. Fornisci informazioni sul servizio Service Discovery come il nome e la descrizione del servizio.
- d. Per fare in modo che Amazon ECS esegua controlli periodici dello stato di salute a livello di contenitore, seleziona Abilita la propagazione dello stato delle attività di Amazon ECS.
- e. Per Tipo di record DNS seleziona il tipo di record DNS da creare per il servizio. Amazon ECS service discovery supporta solo i record A e SRV, a seconda della modalità di rete specificata dalla definizione dell'attività. Per informazioni su questi tipi di record, consulta [Tipi di record DNS supportati](#) nella Guida per gli sviluppatori di Amazon Route 53.
 - Se la definizione di attività specificata dalla tua attività di servizio usa la modalità di rete `bridge` o `host`, sono supportati solo i record di tipo SRV. Scegli un nome di container e una combinazione di porte da associare al record.
 - Se la definizione di attività specificata dalla tua attività di servizio usa la modalità di rete `awsvpc`, seleziona il tipo di record A o SRV. Se scegli A, vai al passaggio successivo. Se scegli SRV, specifica la porta sulla quale si trova il servizio oppure un nome di container e una combinazione di porte da associare al record.

Per TTL, inserisci il tempo in secondi per cui un set di record viene memorizzato nella cache dai resolver DNS e dai browser Web.

10. (Facoltativo) Per interconnettere il servizio utilizzando VPC Lattice, xrp e VPC Lattice, quindi procedi come segue:

- a. Seleziona Attiva VPC Lattice
- b. Per il ruolo dell'infrastruttura, scegli il ruolo dell'infrastruttura.

Se non hai creato un ruolo, scegli Crea ruolo di infrastruttura.

- c. In Target Groups scegli il gruppo o i gruppi target. Devi scegliere almeno un gruppo target e puoi averne al massimo cinque. Scegli Aggiungi gruppo target per aggiungere altri gruppi target. Scegli il nome della porta, il protocollo e la porta per ogni gruppo target che hai scelto.

Per eliminare un gruppo target, scegli Rimuovi.

 Note

- Se desideri aggiungere gruppi target esistenti, devi utilizzare il AWS CLI. Per istruzioni su come aggiungere gruppi target utilizzando il AWS CLI, consulta [register-targets](#) nel riferimento. AWS Command Line Interface
- Sebbene un servizio VPC Lattice possa avere più gruppi target, ogni gruppo target può essere aggiunto a un solo servizio.

- d. Per completare la configurazione VPC Lattice, includendo i nuovi gruppi target nell'azione predefinita del listener o nelle regole di un servizio VPC Lattice esistente nella console VPC Lattice. Per ulteriori informazioni, consulta le [regole del listener per il servizio VPC](#) Lattice.
11. (Facoltativo) Per configurare un load balancer per il servizio, espandi Load balancing (Bilanciamento del carico).

Scegli il load balancer.

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
Application Load Balancer	a. Per Seleziona tipo di load balancer, scegli Applicati on Load Balancer. b. Scegli Creazione di un nuovo load balancer per creare un nuovo servizio di Application Load Balancer o Utilizzo di un load balancer esistente per selezionare un Application Load Balancer esistente. c. Per Load balancer name (Nome load balancer), immetti un nome univoco.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	d. Per Choose container to load balance (Scegli il container per il bilanciamento del carico), scegli il container che ospita il servizio. e. Per Listener, specifica una porta e un protocollo per l'Application Load Balancer su cui ascoltare le richieste di connessione. Di default, il load balancer sarà configurato per utilizzare la porta 80 e HTTP. f. Per Target group name (Nome del gruppo di destinazione), specifica un nome e un protocollo per il gruppo di destinazione a cui l'Application Load Balancer instraderà le richieste. Per impostazione predefinita, il gruppo di destinazione instraderà le richieste al primo container definito nella definizione di attività. g. Per il ritardo di deregistrazione, inserite il numero di secondi in cui il load balancer modificherà lo stato di destinazione.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	UNUSED Il valore predefinito è 300 secondi. h. Per Health check path (Percorso del controllo dell'integrità), specifica un percorso esistente all'interno del container in cui l'Application Load Balancer deve inviare periodicamente le richieste per verificare l'integrità della connessione tra l'Application Load Balancer e il container. L'opzione predefinita è la directory root (/).	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
Network Load Balancer	a. Per Load balancer type (Tipo di load balancer) , scegli Network Load Balancer. b. Per Load Balancer, scegli un Network Load Balancer esistente. c. Per Choose container to load balance (Scegli il container per il bilanciamento del carico), scegli il container che ospita il servizio. d. Per Target group name (Nome del gruppo di destinazione), specifica un nome e un protocollo per il gruppo di destinazione a cui il Network Load Balancer instraderà le richieste. Per impostazione predefinita, il gruppo di destinazione instraderà le richieste al primo container definito nella definizione di attività. e. Per Ritardo di deregistrazione, immettete il numero di secondi in cui il sistema di bilanciamento del carico modificherà lo stato di destinazione.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	UNUSED Il valore predefinito è 300 secondi. f. Per Health check path (Percorso del controllo dell'integrità), specifica un percorso esistente all'interno del container in cui Network Load Balancer deve inviare periodicamente le richieste per verificare l'integrità della connessione tra l'Application Load Balancer e il container. L'opzione predefinita è la directory root (/).	

12. (Facoltativo) Per configurare il servizio Auto Scaling, espandi Service auto scaling e quindi specifica i seguenti parametri. Per utilizzare lo scaling automatico predittivo, che esamina i dati di caricamento precedenti provenienti dai flussi di traffico, configuralo dopo aver creato il servizio. Per ulteriori informazioni, consulta [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).
- Per utilizzare la scalabilità automatica del servizio, seleziona Service auto scaling (Scalabilità automatica del servizio).
 - In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
 - In Numero massimo di attività, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
 - Scegli il tipo di policy. In Tipo di politica di scalabilità, scegli una delle seguenti opzioni.

Per utilizzare questo tipo di politica	Esegui questa operazione	
Monitoraggio degli obiettivi	- In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. - In Policy name (Nome policy), immetti il nome della policy. - Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. - ECSServiceMedioCPUUtilization: utilizzo medio della CPU del servizio. - ECSServiceAverageMemoryUtilization—Utilizzo medio della memoria del servizio. - ALBRequestCountPerTarget— Numero di richieste completate e per destinazione in un gruppo target di Application Load Balancer. - Per Target value (Valore di destinazione), inserisci il valore mantenuto dal servizio per il parametro selezionato.	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	e. Per il periodo di recupero con scalabilità orizzontale, immettete il periodo di tempo, in secondi, dopo un'attività di scalabilità orizzontale (aggiunta di attività) che deve trascorrere prima che possa iniziare un'altra attività di scalabilità orizzontale. f. Per il periodo di recupero scalabile, inserite la quantità di tempo, in secondi, che deve trascorrere dopo un'attività di scalabilità (rimozione delle attività) che deve trascorrere prima che possa iniziare un'altra attività scalabile. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Seleziona Disattiva la scalabilità orizzontale se desideri che la tua politica di scalabilità venga ridimensionata in base	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	all'aumento del traffico, ma non è necessario che venga adattata quando il traffico diminuisce.	

Per utilizzare questo tipo di politica	Esegui questa operazione	
Dimensionamento a fasi	a. In Tipo di policy di dimensionamento, scegli Dimensionamento a fasi. b. In Nome policy, immetti un nome per la policy. c. Per Alarm name (Nome allarme), immetti un nome univoco per l'allarme. d. Per Parametro del servizio Amazon ECS, scegli il parametro da utilizzare per l'allarme. e. Per Statistica, scegli la statistica dell'allarme. f. In Periodo, scegli il periodo dell'allarme. g. Per Condizione di allarme, scegli come confrontare il parametro selezionato con la soglia definita. h. In Soglia per confrontare i parametri e Periodo di valutazione per avviare l'allarme, inserisci la soglia utilizzata per l'allarme e il tempo di valutazione della soglia. i. In Operazioni di dimensionamento,	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	esegui queste operazioni: • In Azione, seleziona se aggiungere, rimuovere o impostare un conteggio specifico desiderato per il servizio. • Se hai scelto di aggiungere o rimuovere attività, in Valore inserisci il numero di attività (o la percentuale di attività esistenti) da aggiungere e o rimuovere quando viene avviata l'azione di ridimensionamento. Se hai scelto di impostare il conteggio desiderato, inserisci il numero di attività. Per Tipo, scegli se Valore è un numero intero o un valore percentuale del conteggio desiderato esistente. • Per Limite inferiore e Limite superiore, inserisci i relativi limiti della regolazione di dimensionamento per fasi. Per impostazioni	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	one predefinita, il limite inferiore per una policy di aggiunta è la soglia di allarme e il limite superiore è positivo (+) infinito. Per impostazione predefinita, il limite superiore per una policy di rimozione è la soglia di allarme e il limite inferiore è negativo (-) infinito. • (Facoltativo) Aggiungi opzioni di dimensionamento aggiuntive. Scegli Aggiungi nuova azione di ridimensionamento, quindi ripeti i passaggi delle azioni di ridimensionamento. • Per Periodo di recupero, inserite la quantità di tempo, in secondi, per attendere che una precedent e attività di ridimensionamento abbia effetto. Per una politica di aggiunta, si tratta del periodo successivo a un'attività di scalabilità	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	orizzontale in cui la politica di scalabilità blocca le attività di scalabilità orizzontale e limita il numero di attività che possono essere scalate orizzontalmente alla volta. Per quanto riguarda una politica di rimozione, si tratta del periodo successivo a un'attività di scalabilità che deve trascorrere prima che possa iniziare un'altra attività scalabile.	

13. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Task Placement (Posizionamento attività), quindi scegli una tra le seguenti opzioni.

Per ulteriori informazioni, consulta [In che modo Amazon ECS colloca le attività sulle istanze di container](#).

- AZ Balanced Spread: distribuisce le attività tra le zone di disponibilità e tra le istanze di container nella zona di disponibilità.
- AZ Balanced BinPack: distribuisce le attività tra le zone di disponibilità e tra le istanze di container con la minima memoria disponibile.
- BinPack— Distribuisce le attività in base alla quantità minima disponibile di CPU o memoria.
- Un'attività per host: posiziona al massimo un'attività dal servizio su ogni istanza del contenitore.
- Personalizzato: definisci la tua strategia di posizionamento delle attività.

Se hai scelto Custom (Personalizzato), definisci l'algoritmo per il posizionamento delle attività e le regole che vengono prese in considerazione durante il posizionamento delle attività.

- In Strategy (Strategia), per Type (Tipo) e Field (Campo), scegli l'algoritmo e l'entità da utilizzare per l'algoritmo.

Puoi aggiungere un massimo di 5 strategie.

- In Vincolo, per Tipo ed Espressione, scegli la regola e l'attributo per il vincolo.

Ad esempio, per impostare il vincolo per posizionare le attività su istanze T2, per Expression (Espressione), immetti `attribute:ecs.instance-type =~ t2.*`.

Puoi aggiungere un massimo di 10 vincoli.

14. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

Il nome del volume e il tipo di volume vengono configurati quando si crea una revisione della definizione dell'attività e non possono essere modificati durante la creazione di un servizio. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione di attività e creare un servizio utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	- Per il tipo di volume EBS, scegli il tipo di volume EBS che desideri allegare all'attività. - Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB.	

Per configurare questo tipo di volume	Esegui questa operazione	
	Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettete il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per <code>io1</code> i tipi di volume e <code>io2 gp3</code> d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). MiBps Questo valore è configurabile solo per il tipo di volume. <code>gp3</code> e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno snapshot se desideri creare un volume da uno snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Se si specifica un ID snapshot, è possibile specificare una velocità di inizializzazione del	

Per configurare questo tipo di volume	Esegui questa operazione	
	volume (MiB/s). Immettete un valore compreso tra 100 e 300, in MiB/s, che determinerà la velocità di caricamento dei dati dall'istantanea specifica utilizzando Snapshot ID per la creazione del volume. g. Per Tipo di file system, scegli il tipo di file system che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico. L'impostazione predefinita per Linux è XFS. Per i volumi creati da un'istantanea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. h. Per il ruolo Infrastruttura, scegli un ruolo IAM con le autorizzazioni necessarie che consentan	

Per configurare questo tipo di volume	Esegui questa operazione	
	o ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy <code>AmazonECSInfrastructureRolePolicyForVolumes</code> gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessari e, consulta Ruolo IAM dell'infrastruttura Amazon ECS. i. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specificata nell'impostazione. Se scegli Default e la crittografia predefini	

Per configurare questo tipo di volume	Esegui questa operazione	
	ta di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittografia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se crei un volume da un'istanza crittografata. j. Se hai scelto Personalizzato per la crittografia, devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazi	

Per configurare questo tipo di volume	Esegui questa operazione	
	oni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. k. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione o dal servizio dell'attività o fornendo tag personalizzati. Se desideri propagare i tag dalla definizione dell'attività, scegli Definizione attività per Propagare i tag da. Se desideri propagare i tag dal servizio, scegli Servizio da cui propaga i tag. Se scegli Non propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS,	

Per configurare questo tipo di volume	Esegui questa operazione	
	consulta Tagging dei volumi Amazon EBS .	

15. (Facoltativo) Per identificare il servizio e le attività, espandi la sezione Tags (Tag), quindi configura i tag.

Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag di definizione delle attività, seleziona Attiva i tag gestiti di Amazon ECS, quindi in Propaga i tag da, scegli Definizioni di attività.

Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag del servizio, seleziona Attiva i tag gestiti di Amazon ECS, quindi in Propaga i tag da, scegli Servizio.

Aggiungi o rimuovi un tag.

- [Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
- [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).

16. Scegli Create (Crea).

Passaggi successivi

Di seguito sono riportate le azioni aggiuntive dopo la creazione di un servizio.

- Configura la scalabilità automatica delle previsioni, che analizza i dati di caricamento precedenti provenienti dai flussi di traffico. Per ulteriori informazioni, consulta [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).
- Tieni traccia della tua implementazione e visualizza la cronologia dei servizi che Amazon ECS ha interrotto. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS](#).

Aggiornamento di un servizio Amazon ECS tramite la console

Puoi aggiornare la definizione dell'attività, il numero di attività desiderato, la strategia del fornitore di capacità, la versione della piattaforma e la configurazione di distribuzione o qualsiasi combinazione di questi elementi. La configurazione corrente del servizio è precompilata.

Per informazioni su come aggiornare la configurazione di implementazione blu/verde, consulta [Aggiornamento di una distribuzione blu/verde di Amazon ECS tramite la console](#).

Quando utilizzi la console, considera i seguenti aspetti:

Se desideri interrompere temporaneamente il servizio, imposta Attività desiderate su 0. Quindi, quando sei pronto per avviare il servizio, aggiorna il servizio con il conteggio originale delle attività desiderate.

Quando utilizzi la console, considera i seguenti aspetti:

- È necessario utilizzare il AWS Command Line Interface per aggiornare un servizio che utilizza uno dei seguenti parametri:
 - Distribuzioni blu/verde
 - Service Discovery: è possibile visualizzare solo la configurazione di Service Discovery.
 - Policy di monitoraggio con un parametro personalizzato
 - Servizio di aggiornamento: non è possibile aggiornare la configurazione di awsvpc rete e il periodo di tolleranza per il controllo dello stato di salute.

Per informazioni su come aggiornare un servizio utilizzando il AWS CLI, vedere [update-service](#) nel AWS Command Line Interface riferimento.

- Se stai modificando le porte utilizzate dai container in una definizione di attività, potrebbe essere necessario aggiornare i gruppi di sicurezza per le istanze di container in modo che funzionino con le porte aggiornate.
- Amazon ECS non aggiorna automaticamente i gruppi di sicurezza associati ai sistemi di bilanciamento del carico Elastic Load Balancing o alle istanze di container di Amazon ECS.
- Se il servizio utilizza un sistema di bilanciamento del carico, la configurazione di quest'ultimo definita per il servizio al momento della creazione non può essere modificata. Puoi invece utilizzare AWS CLI o SDK per modificare la configurazione del load balancer. Per informazioni su come modificare la configurazione, consulta [UpdateService](#) nel riferimento all'API di riferimento di Amazon Elastic Container Service.

- Se aggiorni la definizione di attività per il servizio, il nome e la porta del container specificati nella configurazione del sistema di bilanciamento del carico devono rimanere nella definizione di attività.

Puoi aggiornare un servizio esistente per modificarne alcuni parametri di configurazione, ad esempio il numero di processi gestiti dal servizio, specificare quale definizione di attività viene utilizzata dai processi oppure, se i processi utilizzano il tipo di avvio Fargate, modificare la versione della piattaforma utilizzata dal servizio. Un servizio che utilizza una versione della piattaforma Linux non può essere aggiornato per utilizzare una versione della piattaforma Windows e viceversa. Se hai un'applicazione che ha bisogno di capacità maggiore, puoi aumentare la capacità del servizio. Se hai capacità superflua, puoi ridurre il numero di attività desiderate nel servizio e liberare risorse.

Il comportamento di estrazione dell'immagine del contenitore è diverso a seconda del tipo di avvio. Per ulteriori informazioni, consultare uno dei seguenti argomenti:

- [Tipo di lancio Fargate per Amazon ECS](#)
- [EC2 tipo di avvio per Amazon ECS](#)
- [Tipo di lancio esterno \(Amazon ECS Anywhere\) per Amazon ECS](#)

Il pianificatore del servizio utilizza i parametri di percentuale minima di attività integre e di percentuale massima (nella configurazione di implementazione del servizio) per determinare la strategia di distribuzione.

Se il servizio utilizza il tipo di distribuzione aggiornamento in sequenza (ECS), la percentuale minima di attività integre rappresenta il numero minimo di attività del servizio che devono rimanere nello stato RUNNING durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Il parametro si applica anche in presenza di istanze di container in stato DRAINING, se il servizio contiene attività che utilizzano il tipo di lancio EC2. Utilizza questo parametro per eseguire l'implementazione senza impiegare capacità aggiuntiva del cluster. Ad esempio, se il servizio ha un numero desiderato di quattro attività e una percentuale minima di attività integre del 50%, lo scheduler può arrestare due attività esistenti per liberare capacità del cluster prima di avviare due nuove attività. Le attività dei servizi che non utilizzano un sistema di bilanciamento del carico vengono considerate integre se sono in stato RUNNING. Le attività per i servizi che utilizzano un sistema di bilanciamento del carico vengono considerate integre se sono in stato RUNNING e se sono considerate integre da tale sistema. Il valore predefinito per la percentuale minima di integrità è 100%.

Se il servizio utilizza il tipo di implementazione aggiornamento in sequenza (ECS), il parametro di percentuale massima rappresenta il numero massimo di attività del servizio che possono restare in stato PENDING, RUNNING o STOPPING durante un'implementazione, espresso come percentuale del numero di attività desiderato (arrotondata per difetto al numero intero più vicino). Il parametro si applica anche quando tutte le istanze del contenitore sono nello DRAINING stato se il servizio contiene attività che utilizzano il tipo di EC2 avvio. Utilizza questo parametro per definire le dimensioni del batch di implementazione. Ad esempio, se il servizio ha un numero desiderato di quattro attività e un valore percentuale massimo del 200%, lo scheduler può avviare quattro nuove attività prima di arrestare le quattro precedenti, Questo a condizione che le risorse del cluster necessarie per questa operazione siano disponibili. Il valore predefinito per la percentuale massima è 200%.

Quando il pianificatore del servizio sostituisce un'attività durante un aggiornamento, se il servizio utilizza un load balancer, rimuove prima l'attività da tale sistema e attende la fine delle connessioni. Quindi, viene emesso l'equivalente del comando docker stop ai container in esecuzione nell'attività. Questo determina un segnale SIGTERM e un timeout della durata di 30 secondi, dopo il quale viene inviato un segnale SIGKILL e i container vengono forzatamente arrestati. Se il container gestisce il segnale SIGTERM normalmente ed esce entro 30 secondi dalla ricezione, non viene inviato un segnale SIGKILL. Il pianificatore del servizio avvia e arresta le attività secondo quanto definito dalle impostazioni di percentuale minima di attività integre e percentuale massima.

Il pianificatore di servizi sostituisce inoltre le attività ritenute non integre dopo l'esito negativo di un controllo dell'integrità del container o di un sistema di bilanciamento del carico del gruppo di destinazione. Questa sostituzione dipende dai parametri di definizione del servizio `maximumPercent` e `desiredCount`. Se un'attività è contrassegnata come non integra, il pianificatore di servizi avvierà innanzitutto un'attività di sostituzione. Quindi, accade quanto segue.

- Se lo stato di integrità dell'attività sostitutiva è pari a `HEALTHY`, l'utilità di pianificazione del servizio interrompe l'attività non integra
- Se lo stato di integrità dell'attività di sostituzione è `UNHEALTHY`, il pianificatore interromperà l'attività di sostituzione non integra o l'attività esistente non integra per far sì che il numero totale delle attività sia pari a `desiredCount`.

Se il parametro `maximumPercent` impedisce al pianificatore di avviare un'attività di sostituzione, il pianificatore interromperà un'attività non integra alla volta, in modo casuale, per liberare spazio, e poi avvierà un'attività di sostituzione. Il processo di avvio e arresto continua fino a quando tutte le attività non integre vengono sostituite con attività integre. Dopo aver sostituito tutte le attività non

integre e aver avviato solo quelle integre, se il numero totale delle attività supera `desiredCount`, le attività integre vengono interrotte casualmente fino a quando il numero totale delle attività è pari a `desiredCount`. Per ulteriori informazioni sui parametri `maximumPercent` e `desiredCount`, consulta [Parametri di definizione del servizio](#).

Important

Se stai modificando le porte utilizzate dai container in una definizione di attività, potrebbe essere necessario aggiornare i gruppi di sicurezza per le istanze di container in modo che funzionino con le porte aggiornate.

Se aggiorni la definizione di attività per il servizio, il nome del container e la porta del container specificati al momento della creazione del servizio devono rimanere nella definizione di attività.

Amazon ECS non aggiorna automaticamente i gruppi di sicurezza associati ai sistemi di bilanciamento del carico Elastic Load Balancing o alle istanze di container di Amazon ECS.

Quando aggiorni un servizio che utilizza Amazon ECS circuit breaker, Amazon ECS crea una distribuzione e una revisione del servizio. Queste risorse consentono di visualizzare informazioni dettagliate sulla cronologia dei servizi. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS](#).

Procedura

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, seleziona la casella di controllo accanto al servizio, quindi scegli Aggiorna.
4. Per fare in modo che il tuo servizio inizi una nuova implementazione, seleziona Force new deployment (Forza una nuova implementazione).
5. Per Definizione dell'attività, scegli la famiglia di definizioni dell'attività e la revisione.

Important

La console verifica che la famiglia e la revisione delle definizioni delle attività selezionate siano compatibili con la configurazione di elaborazione definita. Se ricevi un avviso,

verifica sia la compatibilità della definizione dell'attività che la configurazione di elaborazione selezionata.

6. Se hai scelto Replica, per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e mantenere nel servizio.
7. Se hai scelto Replica, per consentire ad Amazon ECS di monitorare la distribuzione delle attività tra le zone di disponibilità e ridistribuirle in caso di squilibrio, in Availability Zone service rebalancing, seleziona Availability Zone service rebalancing.
8. Per Min running tasks (Numero minimo di attività in esecuzione), specifica il limite inferiore per il numero di attività nel servizio che devono rimanere nello stato RUNNING durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Per ulteriori informazioni, consulta [Configurazione dell'implementazione](#).
9. Per Max running tasks (Numero massimo di attività in esecuzione), specifica il limite superiore per il numero di attività del servizio consentite nello stato RUNNING o PENDING durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per difetto al numero intero più vicino).
10. Per configurare il modo in cui Amazon ECS rileva e gestisce gli errori di implementazione, espandi Deployment failure detection (Rilevamento degli errori di implementazione), quindi scegli le tue opzioni.
 - a. Per interrompere un'implementazione quando le attività non possono essere avviate, seleziona Use the Amazon ECS deployment circuit breaker (Usa l'interruttore automatico di implementazione di Amazon ECS).

Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di distribuzione completato quando l'interruttore di distribuzione imposta la distribuzione su uno stato fallito, seleziona Rollback on failures.

- b. Per interrompere una distribuzione in base alle metriche dell'applicazione, seleziona Usa CloudWatch allarmi. Quindi, dal nome CloudWatch dell'allarme, scegli gli allarmi. Per creare un nuovo allarme, vai alla CloudWatch console.

Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di distribuzione completato quando un CloudWatch allarme imposta la distribuzione su uno stato fallito, seleziona Rollback in caso di errori.

11. Per modificare le opzioni di elaborazione, espandi Compute configuration, quindi procedi come segue:

- a. Per i servizi attivi AWS Fargate, per la versione della piattaforma, scegli la nuova versione.
- b. Per i servizi che utilizzano una strategia per i provider di capacità, per la strategia dei fornitori di capacità, procedi come segue:
 - Per aggiungere un provider di capacità aggiuntivo, scegli Aggiungi altro. Quindi, scegli il provider in Provider di capacità.
 - Per rimuovere un provider di capacità, scegli Rimuovi a destra del provider.

Un servizio che utilizza un provider di capacità di gruppo Auto Scaling non può essere aggiornato per utilizzare un provider di capacità Fargate. Un servizio che utilizza un provider di capacità Fargate non può essere aggiornato per utilizzare un provider di capacità di gruppo Auto Scaling.

12. (Facoltativo) Per configurare il servizio Auto Scaling, espandi Service auto scaling e quindi specifica i seguenti parametri. Per utilizzare lo scaling automatico predittivo, che esamina i dati di caricamento precedenti provenienti dai flussi di traffico, configuralo dopo aver creato il servizio. Per ulteriori informazioni, consulta [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).

- a. Per utilizzare la scalabilità automatica del servizio, seleziona Service auto scaling (Scalabilità automatica del servizio).
- b. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
- c. In Numero massimo di attività, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
- d. Scegli il tipo di policy. In Tipo di politica di scalabilità, scegli una delle seguenti opzioni.

Per utilizzare questo tipo di politica	Esegui questa operazione	
Monitoraggio degli obiettivi	- In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. - In Policy name (Nome policy), immetti il nome della policy. - Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. - ECSServiceMedioCPUUtilization: utilizzo medio della CPU del servizio. - ECSServiceAverageMemoryUtilization—Utilizzo medio della memoria del servizio. - ALBRequestCountPerTarget— Numero di richieste completate e per destinazione in un gruppo target di Application Load Balancer. - Per Target value (Valore di destinazione), inserisci il valore mantenuto dal servizio per il parametro selezionato.	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	e. Per il periodo di recupero con scalabilità orizzontale, immettete il periodo di tempo, in secondi, dopo un'attività di scalabilità orizzontale (aggiunta di attività) che deve trascorrere prima che possa iniziare un'altra attività di scalabilità orizzontale. f. Per il periodo di recupero scalabile, inserite la quantità di tempo, in secondi, che deve trascorrere dopo un'attività di scalabilità (rimozione delle attività) che deve trascorrere prima che possa iniziare un'altra attività scalabile. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Seleziona Disattiva la scalabilità orizzontale se desideri che la tua politica di scalabilità venga ridimensionata in base	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	all'aumento del traffico, ma non è necessario che venga adattata quando il traffico diminuisce.	

Per utilizzare questo tipo di politica	Esegui questa operazione	
Dimensionamento a fasi	a. In Tipo di policy di dimensionamento, scegli Dimensionamento a fasi. b. In Nome policy, immetti un nome per la policy. c. Per Alarm name (Nome allarme), immetti un nome univoco per l'allarme. d. Per Parametro del servizio Amazon ECS, scegli il parametro da utilizzare per l'allarme. e. Per Statistica, scegli la statistica dell'allarme. f. In Periodo, scegli il periodo dell'allarme. g. Per Condizione di allarme, scegli come confrontare il parametro selezionato con la soglia definita. h. In Soglia per confrontare i parametri e Periodo di valutazione per avviare l'allarme, inserisci la soglia utilizzata per l'allarme e il tempo di valutazione della soglia. i. In Operazioni di dimensionamento,	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	esegui queste operazioni: • In Azione, seleziona se aggiungere, rimuovere o impostare un conteggio specifico desiderato per il servizio. • Se hai scelto di aggiungere o rimuovere attività, in Valore inserisci il numero di attività (o la percentuale di attività esistenti) da aggiungere e o rimuovere quando viene avviata l'azione di ridimensionamento. Se hai scelto di impostare il conteggio desiderato, inserisci il numero di attività. Per Tipo, scegli se Valore è un numero intero o un valore percentuale del conteggio desiderato esistente. • Per Limite inferiore e Limite superiore, inserisci i relativi limiti della regolazione di dimensionamento per fasi. Per impostazioni	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	one predefinita, il limite inferiore per una policy di aggiunta è la soglia di allarme e il limite superiore è positivo (+) infinito. Per impostazione predefinita, il limite superiore per una policy di rimozione è la soglia di allarme e il limite inferiore è negativo (-) infinito. • (Facoltativo) Aggiungi opzioni di dimensionamento aggiuntive. Scegli Aggiungi nuova azione di ridimensionamento, quindi ripeti i passaggi delle azioni di ridimensionamento. • Per Periodo di recupero, inserite la quantità di tempo, in secondi, per attendere che una precedent e attività di ridimensionamento abbia effetto. Per una politica di aggiunta, si tratta del periodo successivo a un'attività di scalabilità	

Per utilizzare questo tipo di politica	Esegui questa operazione	
	orizzontale in cui la politica di scalabilità blocca le attività di scalabilità orizzontale e limita il numero di attività che possono essere scalate orizzontalmente alla volta. Per quanto riguarda una politica di rimozione, si tratta del periodo successivo a un'attività di scalabilità che deve trascorrere prima che possa iniziare un'altra attività scalabile.	

13. (Facoltativo) Per utilizzare Service Connect, seleziona Turn on Service Connect (Attiva Service Connect), quindi specifica quanto segue:
 - a. In Service Connect configuration (Configurazione Service Connect), specifica la modalità client.
 - Se il servizio esegue un'applicazione client di rete che deve connettersi solo ad altri servizi nello spazio dei nomi, scegli Client side only (Solo lato client).
 - Se il servizio esegue un'applicazione di rete o di servizio Web, deve fornire endpoint per questo servizio e si connette ad altri servizi nello spazio dei nomi, scegli Client and server (Client e server).
 - b. Per utilizzare uno spazio dei nomi differente da quello del cluster predefinito, per Namespace (Spazio dei nomi), scegli lo spazio dei nomi del servizio.
14. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

Il nome del volume e il tipo di volume vengono configurati quando si crea una revisione della definizione dell'attività e non possono essere modificati quando si aggiorna un servizio. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione di attività e aggiornare il servizio utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per il tipo di volume EBS, scegli il tipo di volume EBS che desideri allegare all'attività. b. Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettere il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per <code>io1</code> e <code>io2 gp3</code> i tipi di volume. d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). Questo valore è configurabile solo per il tipo di volume <code>gp3</code>.	

Per configurare questo tipo di volume	Esegui questa operazione	
	e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno snapshot se desideri creare un volume da uno snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Se si specifica un ID snapshot, è possibile specificare una velocità di inizializzazione del volume (MiB/s). Immettete un valore compreso tra 100 e 300, in MiB/s, che determinerà la velocità di caricamento dei dati dall'istantanea specifica utilizzando Snapshot ID per la creazione del volume. g. Per Tipo di file system, scegli il tipo di file system che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico. L'impostazione predefini	

Per configurare questo tipo di volume	Esegui questa operazione	
	ta per Linux èXFS. Per i volumi creati da un'istanza, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istanza. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. h. Per il ruolo Infrastruttura, scegli un ruolo IAM con le autorizzazioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy <code>AmazonECSInfrastructureRolePolicyForVolumes</code> gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Ruolo IAM dell'infrastruttura Amazon ECS.	

Per configurare questo tipo di volume	Esegui questa operazione	
	i. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specificata nell'impostazione. Se scegli Default e la crittografia predefinita di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittografia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se crei un volume da un'istanza crittografata. j. Se hai scelto Personalizzato per la crittografia,	

Per configurare questo tipo di volume	Esegui questa operazione	
	devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazioni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. k. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione o dal servizio dell'attività o fornendo tag personalizzati. Se desideri propagare i tag dalla definizione dell'attività, scegli Definizione attività per Propagare i tag da. Se desideri propagare i tag dal servizio, scegli Servizio da cui propaga i tag. Se scegli Non	

Per configurare questo tipo di volume	Esegui questa operazione	
	propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS.	

15. (Facoltativo) Per identificare il tuo servizio, espandi la sezione Tags (Tag), quindi configura i tag.

- [Aggiungi un tag] Scegli Aggiungi tag ed esegui le seguenti operazioni:
 - In Chiave, immetti il nome della chiave.
 - In Valore, immetti il valore della chiave.
- [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).

16. Scegli Aggiorna.

Passaggi successivi

Tieni traccia della tua implementazione e visualizza la cronologia dei servizi che Amazon ECS ha interrotto. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS](#).

Aggiornamento di una distribuzione blu/verde di Amazon ECS tramite la console

Puoi aggiornare le opzioni di distribuzione: blue/green deployment configuration using the Amazon ECS console. The current blue/green deployment configuration is pre-populated. You can update the following blue/green

- Nome del gruppo di distribuzione: le impostazioni CodeDeploy di distribuzione
- Nome dell'applicazione: il gruppo CodeDeploy di distribuzione
- Configurazione di distribuzione: in che modo CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione
- Listener di test sul load balancer: CodeDeploy utilizza il listener di test per indirizzare il traffico di test verso l'attività sostitutiva impostata durante una distribuzione

È necessario configurare la nuova opzione prima di aggiornare la configurazione.

Per aggiornare una configurazione di implementazione blu/verde (console Amazon ECS)

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster) seleziona il cluster.
3. Nella pagina Cluster overview (Panoramica del cluster), seleziona il servizio, quindi scegli Update (Aggiorna).
4. Espandi le opzioni di distribuzione - Powered by CodeDeploy, quindi scegli le opzioni da aggiornare:
 - Per modificare il gruppo CodeDeploy di distribuzione, per Nome applicazione, scegli il gruppo di distribuzione.
 - Per modificare le impostazioni CodeDeploy di distribuzione, per Nome del gruppo di distribuzione, scegli il gruppo.
 - Per modificare il modo in cui CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione, per la configurazione di Deployment, scegli l'opzione.
5. Seleziona gli hook degli eventi del ciclo di vita dell'implementazione e le relative funzioni Lambda da eseguire nell'ambito della nuova revisione dell'implementazione del servizio. Gli hook del ciclo di vita disponibili sono:

- **BeforeInstall**— Utilizzate questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda prima della creazione del set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita non attiva un ripristino dello stato precedente.
 - **AfterInstall**— Utilizza questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda dopo la creazione del set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.
 - **BeforeAllowTraffic**— Utilizzate questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda prima che il traffico di produzione venga reindirizzato al set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.
 - **AfterAllowTraffic**— Utilizzate questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda dopo che il traffico di produzione è stato reindirizzato al set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.
6. Per modificare il listener di test, espandi Load balancing, quindi per Test listener for deployment, scegli il listener di test. CodeDeploy
 7. Scegli Aggiorna.

Eliminazione di un servizio Amazon ECS tramite la console

Di seguito sono riportati alcuni dei motivi per cui dovresti eliminare un servizio:

- L'applicazione non è più necessaria
- Stai migrando il servizio in un nuovo ambiente
- L'applicazione non viene utilizzata attivamente
- L'applicazione utilizza più risorse del necessario e state cercando di ottimizzare i costi

Prima dell'eliminazione, la capacità del servizio viene automaticamente ridotta a zero. Le risorse del sistema di bilanciamento del carico o le risorse di rilevamento del servizio che sono associate al servizio non sono interessate dall'eliminazione del servizio. Per eliminare le risorse di Elastic Load Balancing, consulta uno dei seguenti argomenti, a seconda del tipo di sistema di bilanciamento del carico: [Eliminazione di un Application Load Balancer](#) o [Eliminazione di Network Load Balancer](#).

Quando elimini un servizio, Amazon ECS elimina tutte le distribuzioni e le revisioni del servizio.

Quando elimini un servizio, se ci sono ancora attività in esecuzione che richiedono la pulizia, lo stato del servizio passa da ACTIVE a DRAINING e il servizio non è più visibile nella console o nel funzionamento dell'API. `ListServices` Dopo che tutte le attività sono passate allo stato STOPPING o STOPPED, lo stato del servizio passa da DRAINING a INACTIVE. I servizi nello INACTIVE stato DRAINING o possono ancora essere visualizzati con l'operazione `DescribeServices` API.

Important

Se tenti di creare un nuovo servizio con lo stesso nome di un servizio esistente in uno ACTIVE dei due stati, riceverai un errore. DRAINING

Procedura

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Clusters (Cluster), seleziona il cluster per il servizio.
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella *name* pagina Cluster:, scegli la scheda Servizi.
5. Selezionare i servizi e quindi seleziona Delete (Elimina).
6. Per eliminare un servizio anche se non è stato ridotto a zero attività, seleziona Force delete service (Forza eliminazione servizio).
7. Alla richiesta di conferma, inserisci delete, quindi scegli Elimina.

Implementa i servizi Amazon ECS sostituendo le attività

Quando crei un servizio che utilizza il tipo di distribuzione rolling update (ECS), lo scheduler del servizio Amazon ECS sostituisce le attività attualmente in esecuzione con nuove attività. Il numero di attività che Amazon ECS aggiunge o rimuove durante un aggiornamento in sequenza è controllato dalla configurazione dell'implementazione del servizio.

Amazon ECS utilizza i seguenti parametri per determinare il numero di attività:

- La `minimumHealthyPercent` rappresenta il limite inferiore del numero di processi che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento, come percentuale del numero desiderato di processi per il servizio.

Questo valore viene arrotondato per eccesso. Ad esempio, se la percentuale minima di integrità è 50, il numero di processi desiderato è quattro, il pianificatore può interrompere due processi esistenti prima di avviare due nuovi processi. Allo stesso modo, se la percentuale di integrità minima è 75% e il numero di processi desiderato è due, il pianificatore non può interrompere alcun processo a causa del valore risultante che è anche due.

Se le attività non funzionano correttamente, lo strumento di pianificazione dei servizi Amazon ECS avvierà prima le attività sostitutive e le manterrà finché le `minimumHealthyPercent` attività sostitutive non diventeranno integre. Man mano che le attività sostitutive vengono avviate e diventano corrette, quelle non sane verranno gradualmente interrotte.

- La `maximumPercent` rappresenta il limite superiore del numero di processi che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento, come percentuale del numero desiderato di processi per il servizio. Questo valore viene arrotondato per difetto. Ad esempio, se la percentuale massima è 200 e il numero di processi desiderato è quattro, il pianificatore può avviare quattro nuovi processi prima di avviare arrestare i quattro processi esistenti. Allo stesso modo, se la percentuale di integrità massima è 125 e il numero di processi desiderato è tre, il pianificatore non può interrompere alcun processo a causa del valore risultante che è esso stesso tre.

Important

Quando si imposta una percentuale di integrità minima o massima, è necessario assicurarsi che lo scheduler possa arrestare o avviare almeno un'attività quando viene attivata un'implementazione. Se il servizio dispone di un'implementazione bloccata a causa di una configurazione di distribuzione non valida, verrà inviato un messaggio di evento del servizio. Per ulteriori informazioni, consulta [service \(*service-name*\) non è stato in grado di interrompere o avviare le attività durante una distribuzione a causa della configurazione della distribuzione del servizio. Aggiorna il valore `minimumHealthyPercent` o `MaximumPercent` e riprova..](#)

Le distribuzioni in sequenza hanno due metodi che consentono di identificare rapidamente quando l'implementazione di un servizio non è riuscita:

- [the section called “In che modo l'interruttore di distribuzione rileva i guasti”](#)
- [the section called “In che modo gli CloudWatch allarmi rilevano gli errori di distribuzione”](#)

I metodi possono essere utilizzati separatamente o insieme. Quando vengono utilizzati entrambi i metodi, la distribuzione viene impostata come non riuscita non appena vengono soddisfatti i criteri di errore per entrambi i metodi di errore.

Utilizza le seguenti linee guida per determinare quale metodo usare:

- Interruttore: utilizza questo metodo quando desideri interrompere un'implementazione quando le attività non possono essere avviate.
- CloudWatch allarmi: utilizzate questo metodo quando desiderate interrompere una distribuzione in base ai parametri dell'applicazione.

Entrambi i metodi supportano il rollback alla versione precedente del servizio.

Risoluzione dell'immagine del contenitore

Per impostazione predefinita, Amazon ECS risolve i tag di immagine del contenitore specificati nella definizione dell'attività in digest di immagini del contenitore. Se crei un servizio che esegue e gestisce una singola attività, tale attività viene utilizzata per stabilire digest di immagini per i contenitori inclusi nell'attività. Se si crea un servizio che esegue e gestisce più attività, la prima attività avviata dal service scheduler durante la distribuzione viene utilizzata per stabilire i digest di immagini per i contenitori delle attività.

Se tre o più tentativi di stabilire gli image digest del contenitore falliscono, la distribuzione continua senza la risoluzione dell'immagine digest. Se l'interruttore automatico di distribuzione è abilitato, la distribuzione viene inoltre fallita e ripristinata.

Dopo aver stabilito i digest delle immagini del contenitore, Amazon ECS li utilizza per avviare qualsiasi altra attività desiderata e per eventuali futuri aggiornamenti del servizio. Ciò comporta che tutte le attività di un servizio eseguano sempre immagini di container identiche, con conseguente coerenza delle versioni del software.

È possibile configurare questo comportamento per ogni contenitore dell'attività utilizzando il `versionConsistency` parametro nella definizione del contenitore. Per ulteriori informazioni, consulta [versionConsistency](#).

Note

- Le versioni di Amazon ECS Agent precedenti a 1.31.0 non supportano la risoluzione Image Digest. Versioni Agent 1.31.0 che 1.69.0 supportano la risoluzione Image Digest

solo per le immagini inviate ai repository Amazon ECR. Le versioni Agent 1.70.0 o successive supportano la risoluzione Image Digest per tutte le immagini.

- La versione minima della piattaforma Fargate Linux per la risoluzione dell'immagine digest è 1.3.0. La versione minima della piattaforma Fargate Windows per la risoluzione di Image Digest è 1.0.0.
- Amazon ECS non acquisisce digest di container sidecar gestiti da Amazon ECS, come l'agente di GuardDuty sicurezza Amazon o il proxy Service Connect.
- Per ridurre la potenziale latenza associata alla risoluzione delle immagini dei container nei servizi con più attività, esegui la versione dell'agente Amazon ECS 1.83.0 o superiore sulle istanze di EC2 container. Per evitare una potenziale latenza, specifica i digest delle immagini del contenitore nella definizione dell'attività.
- Se crei un servizio con un numero di attività desiderato pari a zero, Amazon ECS non può stabilire i container digest finché non attivi un'altra distribuzione del servizio con un numero di attività desiderato maggiore di zero.
- Per stabilire digest di immagini aggiornati, puoi forzare una nuova distribuzione. I digest aggiornati verranno utilizzati per avviare nuove attività e non influiranno sulle attività già in esecuzione. Per ulteriori informazioni su come forzare nuove distribuzioni, consulta il riferimento alle [forceNewDeployment](#) API di Amazon ECS.

In che modo l'interruttore di distribuzione di Amazon ECS rileva i guasti

L'interruttore di implementazione è il meccanismo di aggiornamento in sequenza che determina se le attività raggiungono uno stato stazionario. L'interruttore di implementazione dispone di un'opzione che consente di eseguire automaticamente il rollback di un'implementazione con esito negativo a un'implementazione nello stato COMPLETED.

Quando la distribuzione di un servizio cambia stato, Amazon ECS invia un evento di modifica dello stato di distribuzione del servizio a EventBridge. Ciò fornisce un modo programmatico per monitorare lo stato delle implementazioni dei servizi. Per ulteriori informazioni, consulta [Eventi di modifica dello stato di implementazione del servizio Amazon ECS](#). Ti consigliamo di creare e monitorare una EventBridge regola con un eventName of SERVICE_DEPLOYMENT_FAILED in modo da poter intraprendere azioni manuali per avviare la distribuzione. Per ulteriori informazioni, consulta la sezione Guida [introduttiva EventBridge alla](#) Amazon EventBridge User Guide.

Quando l'interruttore di implementazione determina il fallimento di un'implementazione, cerca quella più recente con lo stato COMPLETED. Si tratta dell'implementazione che viene utilizzata

come implementazione di rollback. Quando inizia il rollback, l'implementazione cambia da COMPLETED a IN_PROGRESS. Ciò significa che la distribuzione non è idonea per un altro rollback finché non raggiunge uno COMPLETED stato. Quando l'interruttore di implementazione non rileva un'implementazione nello stato COMPLETED, non avvia nuove attività e l'implementazione si blocca.

Quando si crea un servizio, lo scheduler tiene traccia delle attività che non sono state avviate in due fasi.

- Fase 1: lo scheduler monitora le attività per vedere se passano allo stato RUNNING.
 - Operazione riuscita: la distribuzione ha la possibilità di passare allo stato COMPLETED perché più di un'attività è passata allo stato RUNNING. Il criterio di errore viene ignorato e l'interruttore passa alla fase 2.
 - Errore: alcune attività consecutive non sono passate allo stato RUNNING e la distribuzione potrebbe passare allo stato FAILED.
- Fase 2: la distribuzione entra in questa fase quando è presente almeno un'attività nello stato RUNNING. L'interruttore automatico controlla i controlli di integrità relativi alle attività della distribuzione corrente oggetto di valutazione. I controlli di integrità convalidati sono Elastic Load Balancing AWS Cloud Map , controlli dello stato del servizio e controlli dello stato dei container.
 - Operazione riuscita: almeno un'attività è in corso e i controlli di integrità sono stati superati.
 - Fallimento: le attività sostituite a causa di errori nei controlli di integrità hanno raggiunto la soglia di errore.

Considerate quanto segue quando utilizzate il metodo dell'interruttore automatico di distribuzione su un servizio. EventBridge genera la regola.

- La risposta DescribeServices fornisce informazioni sullo stato di un'implementazione, rolloutState e rolloutStateReason. Quando viene avviata una nuova implementazione, lo stato di implementazione inizia in IN_PROGRESS. Quando il servizio raggiunge uno stato stazionario, lo stato di implementazione passa a COMPLETED. Se il servizio non riesce a raggiungere uno stato stazionario e l'interruttore automatico è abilitato, l'implementazione passerà a uno stato FAILED. Una implementazione in uno stato FAILED non avvierà nuove attività.
- Oltre agli eventi di modifica dello stato dell'implementazione del servizio che Amazon ECS invia per le implementazioni che sono state avviate e completate, Amazon ECS invia anche un evento quando un'implementazione con l'interruttore automatico attivato non riesce. Questi eventi forniscono dettagli sul motivo per cui un'implementazione non è riuscita o se un'implementazione

è stata avviata a causa di un ripristino dello stato precedente. Per ulteriori informazioni, consulta [Eventi di modifica dello stato di implementazione del servizio Amazon ECS](#).

- Se viene avviata una nuova implementazione perché un'implementazione precedente non è riuscita e si è verificato il ripristino dello stato precedente, il campo `reason` dell'evento di modifica dello stato di implementazione del servizio indicherà che l'implementazione è stata avviata a causa di un ripristino dello stato precedente.
- L'interruttore automatico di implementazione è supportato solo per i servizi Amazon ECS che utilizzano il controller di implementazione (ECS) dell'aggiornamento in sequenza.
- È necessario utilizzare la console Amazon ECS o, AWS CLI quando si utilizza, l'interruttore automatico di distribuzione con l' `CloudWatch` opzione. Per ulteriori informazioni, consulta [the section called “Creazione di un servizio utilizzando parametri definiti”](#) e [create-service](#) in Informazioni di riferimento sull'AWS Command Line Interface .

L'`create-service` AWS CLI esempio seguente mostra come creare un servizio Linux quando l'interruttore di distribuzione viene utilizzato con l'opzione `rollback`.

```
aws ecs create-service \
  --service-name MyService \
  --deployment-controller type=ECS \
  --desired-count 3 \
  --deployment-configuration "deploymentCircuitBreaker={enable=true,rollback=true}" \
  --task-definition sample-fargate:1 \
  --launch-type FARGATE \
  --platform-family LINUX \
  --platform-version 1.4.0 \
  --network-configuration \
    "awsVpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=ENABLED}"
```

Esempio:

L'implementazione 1 è in uno stato `COMPLETED`.

L'implementazione 2 non può essere avviata, quindi l'interruttore esegue il rollback all'implementazione 1. L'implementazione 1 passa allo stato `IN_PROGRESS`.

L'implementazione 3 viene avviata e non è presente alcuna implementazione nello stato `COMPLETED`, quindi l'implementazione non è in grado di eseguire il rollback o avviare attività.

Soglia di errore

L'interruttore automatico di implementazione calcola il valore di soglia e lo utilizza per determinare quando modificare lo stato della distribuzione in FAILED.

L'interruttore automatico di distribuzione ha una soglia minima di 3 e una soglia massima di 200 e utilizza i valori della formula seguente per determinare l'errore di distribuzione.

$$\text{Minimum threshold} \leq 0.5 * \text{desired task count} \Rightarrow \text{maximum threshold}$$

Quando il risultato del calcolo è superiore al minimo di 3, ma inferiore al massimo di 200, la soglia di errore viene impostata sulla soglia calcolata (arrotondata per eccesso).

Note

Non è possibile modificare nessuno dei valori di soglia.

Esistono due fasi per il controllo dello stato dell'implementazione.

1. L'interruttore automatico di implementazione monitora i processi che fanno parte dell'implementazione e verifica la presenza di attività con stato RUNNING. Il pianificatore ignora i criteri di errore quando un processo nell'implementazione corrente si trova nello stato RUNNING e procede alla fase successiva. Quando i processi non riescono a raggiungere lo stato RUNNING, l'interruttore automatico di implementazione aumenta di uno il numero di errori. Quando il numero di errori è uguale alla soglia, l'implementazione viene contrassegnata come FAILED.
2. Questa fase viene inserita quando RUNNING nello stato sono presenti una o più attività. L'interruttore automatico di implementazione esegue controlli dell'integrità delle seguenti risorse per i processi nell'implementazione corrente:
 - Load balancer Elastic Load Balancing
 - AWS Cloud Map servizio
 - Controlli dell'integrità dei container di Amazon ECS

Quando un controllo dell'integrità di un processo HA ESITO NEGATIVO, l'interruttore automatico di implementazione aumenta di uno il numero di errori. Quando il numero di errori è uguale alla soglia, l'implementazione viene contrassegnata come FAILED.

La tabella seguente mostra alcuni esempi.

Conteggio attività desiderato	Calcolo	Threshold
1	$3 \leq 0.5 * 1 \Rightarrow 200$	3 (il valore calcolato è inferiore al minimo)
25	$3 \leq 0.5 * 25 \Rightarrow 200$	13 (il valore viene arrotondato per eccesso)
400	$3 \leq 0.5 * 400 \Rightarrow 200$	200
800	$3 \leq 0.5 * 800 \Rightarrow 200$	200 (il valore calcolato è superiore al massimo)

Ad esempio, quando la soglia è 3, l'interruttore si avvia con il conteggio dei guasti impostato su 0. Quando un'attività non riesce a raggiungere RUNNING lo stato, l'interruttore automatico di dispiegamento aumenta il numero di guasti di uno. Quando il numero di errori è uguale a 3, la distribuzione viene contrassegnata come. FAILED

Per altri esempi sull'utilizzo dell'opzione di rollback, consulta [Annuncio dell'interruttore automatico di implementazione di Amazon ECS](#).

In che modo CloudWatch gli allarmi rilevano gli errori di distribuzione di Amazon ECS

Puoi configurare Amazon ECS in modo che la distribuzione non sia riuscita quando rileva che uno specifico CloudWatch allarme è entrato nello ALARM stato.

Facoltativamente, è possibile impostare la configurazione per ripristinare un'implementazione non riuscita all'ultima implementazione completata.

L'create-service AWS CLI esempio seguente mostra come creare un servizio Linux quando gli allarmi di distribuzione vengono utilizzati con l'opzione rollback.

```
aws ecs create-service \
  --service-name MyService \
  --deployment-controller type=ECS \
  --desired-count 3 \
  --deployment-configuration
    "alarms={alarmNames=[alarm1Name,alarm2Name],enable=true,rollback=true}" \
```

```
--task-definition sample-fargate:1 \  
--launch-type FARGATE \  
--platform-family LINUX \  
--platform-version 1.4.0 \  
--network-configuration  
"awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=EM
```

Considera quanto segue quando utilizzi il metodo Amazon CloudWatch alarms su un servizio.

- Il tempo di incorporamento è un periodo di tempo dopo che una nuova versione del servizio è stata aumentata e la vecchia versione del servizio è stata ridotta, durante il quale Amazon ECS continua a monitorare l'allarme associato all'implementazione. Amazon ECS calcola questo periodo di tempo in base alla configurazione degli allarmi associata all'implementazione. Non puoi impostare questo valore.
- Il parametro della richiesta `deploymentConfiguration` ora contiene il tipo di dati `alarms`. È possibile specificare i nomi degli allarmi, se utilizzare il metodo `startDeployment` e se avviare un rollback quando gli allarmi indicano un errore di implementazione. Per ulteriori informazioni, consulta il riferimento [CreateService](#) all'API di Amazon Elastic Container Service.
- La risposta `DescribeServices` fornisce informazioni sullo stato di un'implementazione, `rolloutState` e `rolloutStateReason`. Quando viene avviata una nuova implementazione, lo stato di rollout inizia come `IN_PROGRESS`. Quando il servizio raggiunge uno stato stazionario ed è stato raggiunto il tempo di incorporamento, lo stato di rollout passa a `COMPLETED`. Se il servizio non riesce a raggiungere uno stato stazionario e l'allarme è passato allo stato `ALARM`, l'implementazione passerà a uno stato `FAILED`. Una implementazione in uno stato `FAILED` non avvierà nuovi processi.
- Oltre agli eventi di modifica dello stato dell'implementazione del servizio che Amazon ECS invia per le implementazioni che sono state avviate e completate, Amazon ECS invia un evento anche quando un'implementazione che utilizza gli allarmi non riesce. Questi eventi forniscono dettagli sul motivo per cui un'implementazione non è riuscita o se un'implementazione è stata avviata a causa di un ripristino dello stato precedente. Per ulteriori informazioni, consulta [Eventi di modifica dello stato di implementazione del servizio Amazon ECS](#).
- Se una nuova implementazione viene avviata perché un'implementazione precedente non è riuscita ed è stato abilitato il ripristino dello stato precedente, il campo `reason` dell'evento di modifica dello stato di implementazione del servizio indicherà che l'implementazione è stata avviata a causa di un ripristino dello stato precedente.
- Se utilizzi l'interruttore di distribuzione e gli CloudWatch allarmi Amazon per rilevare i guasti, entrambi possono avviare un errore di distribuzione non appena vengono soddisfatti i criteri per

entrambi i metodi. Un rollback si verifica quando si utilizza l'opzione di rollback per il metodo che ha restituito l'errore di implementazione.

- Gli CloudWatch allarmi Amazon sono supportati solo per i servizi Amazon ECS che utilizzano il controller di distribuzione rolling update (ECS).
- Puoi configurare questa opzione utilizzando la console Amazon ECS o il AWS CLI. Per ulteriori informazioni, consulta [the section called “Creazione di un servizio utilizzando parametri definiti” e create-service](#) in Informazioni di riferimento sull'AWS Command Line Interface .
- Potresti notare che lo stato di implementazione rimane IN_PROGRESS per un periodo di tempo prolungato. Questo perché Amazon ECS non modifica lo stato finché non ha eliminato l'implementazione attiva e ciò avviene solo dopo il tempo di incorporamento. A seconda della configurazione degli allarmi, l'implementazione potrebbe richiedere diversi minuti in più rispetto a quando non sono utilizzati gli allarmi (anche se il nuovo set di attività principali è stato aumentato e la vecchia implementazione è stata ridotta). Se utilizzi i CloudFormation timeout, valuta la possibilità di aumentarli. Per ulteriori informazioni, consulta [Creazione delle condizioni di attesa in un modello](#) nella Guida per l'utente di AWS CloudFormation .
- Amazon ECS chiama DescribeAlarms per eseguire il polling degli allarmi. Le chiamate verranno DescribeAlarms conteggiate ai fini delle quote di CloudWatch servizio associate al tuo account. Se disponi di altri AWS servizi che effettuano chiamateDescribeAlarms, Amazon ECS potrebbe avere un impatto sulla necessità di interrogare gli allarmi. Ad esempio, se un altro servizio effettua un numero di DescribeAlarms chiamate sufficiente a raggiungere la quota, tale servizio viene limitato e anche Amazon ECS lo è e non è in grado di generare allarmi. Se viene generato un allarme durante il periodo di limitazione, Amazon ECS potrebbe non ricevere l'allarme e il rollback potrebbe non verificarsi. Non ci sono altri impatti sull'implementazione. Per ulteriori informazioni sulle quote di CloudWatch servizio, consulta le quote di [CloudWatch servizio](#) nella Guida per l'utente. CloudWatch
- Se all'inizio di una implementazione un allarme si trova nello stato ALARM, Amazon ECS non monitorerà gli allarmi per la durata di tale implementazione (Amazon ECS ignora la configurazione degli allarmi). Questo comportamento risolve il caso in cui si desideri avviare una nuova distribuzione per correggere un errore di distribuzione iniziale.

Allarmi consigliati

È consigliabile utilizzare i seguenti parametri di allarme:

- Se utilizzi un Application Load Balancer, utilizza i parametri HTTPCode_ELB_5XX_Count e HTTPCode_ELB_4XX_Count di Application Load Balancer. Questi parametri controllano i picchi

HTTP. Per ulteriori informazioni sulle metriche di Application Load Balancer, consulta CloudWatch le metriche per il tuo Application Load [Balancer nella User Guide for Application Load Balancer](#).

- Per un'applicazione esistente, utilizza i parametri `CPUUtilization` e `MemoryUtilization`. Questi parametri controllano la percentuale di CPU e memoria utilizzate dal cluster o dal servizio. Per ulteriori informazioni, consulta [the section called "Considerazioni"](#).
- Se utilizzi le Amazon Simple Queue Service code nelle tue attività, utilizza la metrica di `ApproximateNumberOfMessagesNotVisible` Amazon SQS. Questo parametro controlla il numero dei messaggi nella coda che vengono differiti e non sono disponibili per la lettura immediata. Per ulteriori informazioni sui parametri di Amazon SQS, consulta Parametri [disponibili per CloudWatch Amazon SQS nella Amazon Simple Queue Service Developer Guide](#).

Interruzione delle distribuzioni dei servizi Amazon ECS

Puoi interrompere manualmente una distribuzione quando l'interruttore automatico o gli allarmi non hanno rilevato un'implementazione difettosa. CloudWatch Sono disponibili i seguenti tipi di interruzione:

- Rollback: questa opzione ripristina la distribuzione del servizio alla revisione precedente del servizio.

Puoi utilizzare questa opzione anche se non hai configurato la distribuzione del servizio per l'opzione di rollback.

È possibile interrompere una distribuzione che si trova in uno dei seguenti stati. Per ulteriori informazioni sugli stati di distribuzione dei servizi, vedere [Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS](#).

- IN SOSPESO: la distribuzione del servizio passa allo stato `ROLLBACK_REQUESTED`, quindi viene avviata l'operazione di rollback.
- IN_PROGRESS - La distribuzione del servizio passa allo stato `ROLLBACK_REQUESTED`, quindi inizia l'operazione di rollback.
- STOP_REQUESTED - La distribuzione del servizio continua a interrompersi.
- ROLLBACK_REQUESTED: la distribuzione del servizio continua l'operazione di rollback.
- ROLLBACK_IN_PROGRESS - La distribuzione del servizio continua l'operazione di rollback.

Procedura

Prima di iniziare, configura le autorizzazioni richieste per la visualizzazione delle distribuzioni dei servizi. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per visualizzare le distribuzioni dei servizi Amazon ECS](#).

Amazon ECS Console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegli Implementazioni.

Viene visualizzata la pagina delle distribuzioni.

5. In Distribuzione continua, scegli Ripristina. Quindi, nella finestra di conferma, scegli Ripristina.

AWS CLI

1. Esegui `list-service-deployments` per recuperare l'ARN di distribuzione del servizio.

Sostituiscili *user-input* con i tuoi valori.

```
aws ecs list-service-deployments --cluster cluster-name --service service-name
```

Prendi nota della `serviceDeploymentArn` distribuzione che desideri interrompere.

```
{
  "serviceDeployments": [
    {
      "serviceDeploymentArn": "arn:aws:ecs:us-west-2:123456789012:service-deployment/cluster-name/service-name/NCWGC2ZR-taawPAYrIaU5",
      "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/cluster-name/service-name",
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/cluster-name",
      "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:123456789012:service-revision/cluster-name/service-name/4980306466373577095",
    }
  ]
}
```



```

        "status": "SUCCESSFUL"
      }
    ]
  }

```

2. Esegui `stop-service-deployments`. Usa quello da `serviceDeploymentArn` cui è stato restituito `list-service-deployments`.

user-input Sostituiscili con i tuoi valori.

```

aws ecs stop-service-deployment --service-deployment-arn
arn:aws:ecs:region:123456789012:service-deployment/cluster-name/service-
name/NCWGC2ZR-taawPAYrIaU5 --stop-type ROLLBACK

```

Passaggi successivi

Decidi quali modifiche apportare al servizio, quindi aggiorna il servizio. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Best practice per i parametri del servizio Amazon ECS

Per garantire che non si verifichino tempi di inattività delle applicazioni, il processo di distribuzione è il seguente:

1. Avvia i nuovi contenitori di applicazioni mantenendo attivi i contenitori esistenti.
2. Verifica che i nuovi contenitori siano integri.
3. Fermate i vecchi contenitori.

A seconda della configurazione di distribuzione e della quantità di spazio libero e non riservato nel cluster, potrebbero essere necessari più cicli per completare la procedura, sostituire tutte le vecchie attività con nuove attività.

Esistono due opzioni di configurazione del servizio che è possibile utilizzare per modificare il numero:

- `minimumHealthyPercent`: 100% (impostazione predefinita)

Il limite inferiore al numero di attività del servizio che devono rimanere RUNNING invariate durante una distribuzione. Si tratta di una percentuale dell'`desiredCount` arrotondato al numero intero più

vicino. Questo parametro consente di eseguire la distribuzione senza utilizzare capacità aggiuntiva del cluster.

- `maximumPercent`: 200% (impostazione predefinita)

Il limite massimo al numero di attività del servizio consentite PENDING nello stato RUNNING o durante una distribuzione. Si tratta di una percentuale del numero `desiredCount` arrotondato per difetto al numero intero più vicino.

Esempio: opzioni di configurazione predefinite

Prendiamo in considerazione il seguente servizio con sei attività, distribuito in un cluster che può ospitare otto attività in totale. Le opzioni di configurazione del servizio predefinite non consentono alla distribuzione di superare il 100% delle sei attività desiderate.

Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire le sei attività.
2. L'utilità di pianificazione avvia due nuove attività perché le impostazioni predefinite richiedono che vi siano sei attività in esecuzione.

Ora ci sono sei attività esistenti e due nuove attività.

3. Lo scheduler interrompe due delle attività esistenti.

Ora ci sono quattro attività esistenti e due nuove.

4. Lo scheduler avvia due nuove attività aggiuntive.

Ora ci sono quattro attività esistenti e quattro nuove attività.

5. L'utilità di pianificazione chiude due delle attività esistenti.

Ora ci sono due attività esistenti e quattro nuove.

6. Lo scheduler avvia due nuove attività aggiuntive.

Ora ci sono due attività esistenti e sei nuove attività.

7. L'utilità di pianificazione chiude le ultime due attività esistenti.

Ora ci sono sei nuove attività.

Nell'esempio precedente, se si utilizzano i valori predefiniti per le opzioni, c'è un'attesa di 2,5 minuti per ogni nuova attività che inizia. Inoltre, il sistema di bilanciamento del carico potrebbe dover attendere 5 minuti prima che la vecchia attività si interrompa.

Esempio: Modifica **minimumHealthyPercent**

È possibile velocizzare l'implementazione impostando il **minimumHealthyPercent** valore al 50%.

Prendiamo in considerazione il seguente servizio con sei attività, distribuito in un cluster che può ospitare otto attività in totale. Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire sei attività.
2. Lo scheduler interrompe tre delle attività esistenti.

Sono ancora in esecuzione tre attività esistenti che soddisfano il **minimumHealthyPercent** valore.

3. Lo scheduler avvia cinque nuove attività.

Esistono tre attività esistenti e cinque nuove attività.

4. L'utilità di pianificazione interrompe le tre attività esistenti rimanenti.

Sono presenti cinque nuove attività

5. Lo scheduler avvia le nuove attività finali.

Sono disponibili sei nuove attività.

Esempio: modificare lo spazio libero del cluster

È inoltre possibile aggiungere altro spazio libero in modo da poter eseguire attività aggiuntive.

Prendiamo in considerazione il seguente servizio con sei attività, distribuito in un cluster che può ospitare dieci attività in totale. Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire le attività esistenti.
2. Lo scheduler interrompe tre delle attività esistenti,

Esistono tre attività esistenti.

3. Lo scheduler avvia sei nuove attività.

Ci sono le attività esistenti e sei nuove attività

4. Lo scheduler interrompe le tre attività esistenti.

Sono disponibili sei nuove attività.

Raccomandazioni

Utilizza i seguenti valori per le opzioni di configurazione del servizio quando le attività sono inattive da qualche tempo e non hanno un tasso di utilizzo elevato.

- `minimumHealthyPercent`: 50%
- `maximumPercent`: 200%

Visualizza la cronologia dei servizi utilizzando le distribuzioni di servizi Amazon ECS

Le implementazioni dei servizi offrono una visione completa delle tue implementazioni. Le implementazioni del servizio forniscono le seguenti informazioni sul servizio:

- La configurazione del carico di lavoro attualmente distribuita (la revisione del servizio di origine)
- La configurazione del carico di lavoro in fase di implementazione (la revisione del servizio di destinazione)
- Lo stato della distribuzione
- Il numero di attività non riuscite rilevate dall'interruzione del circuito
- Gli CloudWatch allarmi che sono in allarme
- Quando l'implementazione del servizio è iniziata e completata
- I dettagli di un eventuale rollback

Per informazioni sulle proprietà di distribuzione del servizio, vedere [Proprietà incluse in una distribuzione di servizi Amazon ECS](#).

Le distribuzioni dei servizi sono di sola lettura e ciascuna ha un ID univoco.

Esistono tre fasi di implementazione del servizio:

Stage	Definizione	Stati associati
In attesa	È stata creata una distribuzione del servizio, ma non è stata avviata	PENDING
Continua	È in corso l'implementazione di un servizio	• IN_PROGRESS • STOP_REQUIRED • ROLLBACK_RICHIESTO • ROLLBACK_IN_PROGRESS
Completato	L'implementazione di un servizio è terminata (con successo o meno)	• SUCCESSFUL • STOPPED • ROLLBACK_SUCCESSFUL • ROLLBACK_FAILED

Le implementazioni dei servizi vengono utilizzate per comprendere il ciclo di vita del servizio e determinare se vi sono azioni da intraprendere. Ad esempio, se si verifica un rollback, potrebbe essere necessario esaminare la distribuzione del servizio e esaminare gli eventi del servizio.

Puoi visualizzare la cronologia di 90 giorni più recente per le distribuzioni create il o dopo il 25 ottobre 2024 utilizzando la console, l'API e il AWS CLI

È possibile interrompere una distribuzione non completata. Per ulteriori informazioni, consulta [Interruzione delle distribuzioni dei servizi Amazon ECS](#).

Ciclo di vita dell'implementazione del servizio

Amazon ECS crea automaticamente una nuova distribuzione di servizi quando si verifica una delle seguenti azioni:

- Un utente crea un servizio.
- Un utente aggiorna il servizio e utilizza l'opzione Force New Deployment.
- Un utente aggiorna una o più proprietà del servizio che richiedono una distribuzione.

Mentre una distribuzione è in corso, Amazon ECS aggiorna le seguenti proprietà di distribuzione del servizio per riflettere i progressi della distribuzione del servizio:

- Lo stato
- Il numero di attività in esecuzione

Il numero di attività in esecuzione indicato nella revisione del servizio potrebbe non corrispondere al numero effettivo di attività in esecuzione. Questo numero rappresenta il numero di attività in esecuzione al termine della distribuzione. Ad esempio, se le attività sono state avviate indipendentemente dalla distribuzione del servizio, tali attività non sono incluse nel conteggio delle attività in esecuzione per la revisione del servizio.

- Rilevamento di guasti all'interruttore automatico:
 - Il numero di attività che non sono state avviate
- CloudWatch rilevamento dei guasti degli allarmi
 - Gli allarmi attivi
- Informazioni sul rollback:
 - L'ora di inizio
 - Il motivo del rollback
 - L'ARN della revisione del servizio utilizzata per il rollback
- Il motivo dello stato

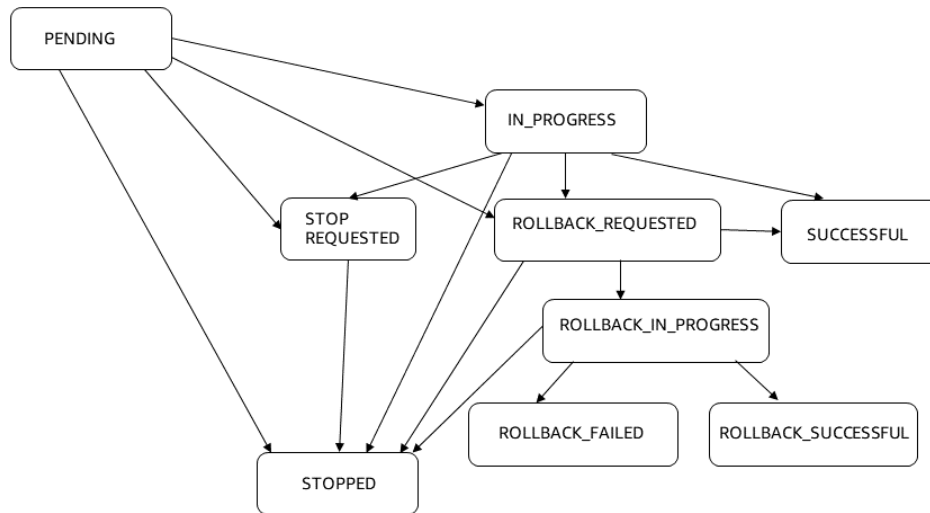
Amazon ECS elimina la distribuzione del servizio quando elimini un servizio.

Stati di distribuzione del servizio

L'implementazione di un servizio inizia nello PENDING stato.

La figura seguente mostra gli stati di distribuzione del servizio che possono verificarsi dopo lo PENDING

stato:IN_PROGRESS,ROLLBACK_REQUESTED,SUCCESSFUL,STOP_REQUESTED,ROLLBACK_IN_PROGRESS,ROLLBACK_FAILED,ROLLBACK_SUCCESSFUL, eSTOPPED.



Le seguenti informazioni forniscono dettagli sugli stati di distribuzione del servizio:

- **PENDING**- La distribuzione del servizio è stata creata, ma non è stata avviata.

Lo stato può passare a **IN_PROGRESS**, **ROLLBACK_REQUESTED**, **STOP_REQUESTED**, o **STOPPED**.

- **IN_PROGRESS**- L'implementazione del servizio è in corso.

Lo stato può passare a

SUCCESSFUL, **STOP_REQUESTED**, **ROLLBACK_REQUESTED**, **ROLLBACK_IN_PROGRESS**, e **STOPPED**.

- **STOP_REQUESTED**- Lo stato di distribuzione del servizio passa a **STOP_REQUESTED** quando si verifica una delle seguenti condizioni:

- Un utente avvia una nuova distribuzione del servizio.
- L'opzione di rollback non è utilizzata per il meccanismo di rilevamento dei guasti (basato sull'interruttore automatico o sull'allarme) e il servizio non raggiunge lo stato. **SUCCESSFUL**

Lo stato passa a. **STOPPED**

- **ROLLBACK_REQUESTED**- Lo stato di distribuzione del servizio passa a **ROLLBACK_REQUESTED** quando un utente richiede un rollback tramite la console, l'API o la CLI.

Lo stato può passare a `SUCCESSFULROLLBACK_IN_PROGRESS`, e. `STOPPED`

- `SUCCESSFUL`- Lo stato di distribuzione del servizio passa a `SUCCESSFUL` quando la distribuzione del servizio viene completata correttamente.
- `ROLLBACK_IN_PROGRESS`- Lo stato di distribuzione del servizio passa al `ROLLBACK_IN_PROGRESS` momento in cui viene utilizzata l'opzione di rollback per il meccanismo di rilevamento dei guasti (basato sull'interruttore o sull'allarme) e il servizio fallisce.

Lo stato passa a, o. `ROLLBACK_SUCCESSFUL` `ROLLBACK_FAILED`

Proprietà incluse in una distribuzione di servizi Amazon ECS

Le seguenti proprietà sono incluse in una distribuzione di servizi.

Proprietà	Descrizione	
ARN per l'implementazione del servizio	L'ARN dell'implementazione del servizio.	
ARN del servizio	L'ARN del servizio per questa distribuzione del servizio.	
ARN del cluster	L'ARN per il cluster che ospita il servizio.	
Ora di creazione della distribuzione del servizio	L'ora in cui è stata creata la distribuzione del servizio.	
Ora di inizio della distribuzione del servizio	L'ora in cui è iniziata la distribuzione del servizio.	
Ora di fine della distribuzione del servizio	L'ora in cui è terminata l'implementazione del servizio.	
Ora di interruzione della distribuzione del servizio	L'ora in cui è stata interrotta la distribuzione del servizio.	

Proprietà	Descrizione	
Ora di aggiornamento della distribuzione del servizio	L'ora dell'ultimo aggiornamento della distribuzione del servizio.	
Revisioni del servizio di origine	Le revisioni del servizio attualmente in esecuzione. Per informazioni sulle proprietà incluse, vedere Proprietà incluse in una revisione del servizio Amazon ECS.	

Proprietà	Descrizione	
Configurazione dell'implementazione	I parametri di implementazione, inclusa la configurazione dell'interruttore automatico, gli allarmi che determinano. • Il limite inferiore al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo , come percentuale del numero desiderato di attività per il servizio. • Il limite massimo al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo , come percentuale del numero desiderato di attività per il servizio. • La configurazione dell'interruttore automatico. • Gli allarmi utilizzati per il rilevamento dei guasti.	

Proprietà	Descrizione	
Revisione del servizio Target	La revisione del servizio da distribuire. Una volta completata correttamente la distribuzione, la revisione del servizio di destinazione è la revisione del servizio in esecuzione.	
Stato di distribuzione del servizio	Lo stato di distribuzione del servizio. I valori validi sono PENDING, SUCCESSFUL, STOP_REQUESTED, STOP_IN_PROGRESS, IN_PROGRESS, ROLLBACK_IN_PROGRESS, ROLLBACK_SUCCESSFUL e ROLLBACK_FAILED.	
Informazioni sullo stato della distribuzione del servizio	Informazioni sul motivo per cui la distribuzione del servizio si trova nello stato attuale. Ad esempio, l'interruttore ha rilevato un guasto.	
Informazioni sul rollback	Le opzioni di rollback utilizzate dalla distribuzione del servizio in caso di errore della distribuzione.	
Opzioni di interruzione del circuito di distribuzione del servizio	L'interruttore automatico che determina l'implementazione del servizio non è riuscito.	

Proprietà	Descrizione	
CloudWatch allarmi per l'implementazione del servizio	Gli CloudWatch allarmi che determinano quando l'implementazione di un servizio fallisce.	

Autorizzazioni necessarie per visualizzare le distribuzioni dei servizi Amazon ECS

Se segui la best practice di concessione del privilegio minimo, devi aggiungere autorizzazioni aggiuntive per visualizzare le distribuzioni dei servizi nella console.

È necessario accedere alle seguenti azioni:

- `ListServiceDeployments`
- `DescribeServiceDeployments`
- `DescribeServiceRevisions`

È necessario accedere alle seguenti risorse:

- Servizio
- Distribuzione del servizio
- Revisione del servizio

La seguente politica di esempio contiene le autorizzazioni richieste e limita le azioni a un servizio specificato.

Sostituisci `accountcluster-name`, e `service-name` con i tuoi valori.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:ListServiceDeployments",
        "ecs:DescribeServiceDeployments",
        "ecs:DescribeServiceRevisions"
      ],
      "Resource": [
```

```
        "arn:aws:ecs:us-east-1:123456789012:service/cluster-name/service-name",
        "arn:aws:ecs:us-east-1:123456789012:service-deployment/cluster-name/
service-name/*",
        "arn:aws:ecs:us-east-1:123456789012:service-revision/cluster-name/service-
name/*"
    ]
}
]
```

Visualizzazione delle distribuzioni dei servizi Amazon ECS

Puoi visualizzare la cronologia di 90 giorni più recente delle distribuzioni create a partire dal 25 ottobre 2024. Le distribuzioni dei servizi possono avvenire in uno dei seguenti stati:

- In corso
- In attesa
- Completato

È possibile utilizzare queste informazioni per determinare se è necessario aggiornare la modalità di distribuzione del servizio o la revisione del servizio. Per informazioni sulle proprietà incluse, vedere.

[Proprietà incluse in una distribuzione di servizi Amazon ECS](#)

Prima di iniziare, configura le autorizzazioni richieste per la visualizzazione delle distribuzioni dei servizi. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per visualizzare le distribuzioni dei servizi Amazon ECS](#).

Amazon ECS Console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegli Implementazioni.
5. Scegli la distribuzione del servizio da visualizzare.

Per visualizzare le distribuzioni dei servizi per questo tipo di distribuzione	Esegui questa operazione	
Implementazione continua	In Distribuzioni in corso, scegli l'ID di distribuzione.	
Ultima distribuzione	In Ultima distribuzione, scegli l'ID di distribuzione.	
Distribuzioni completate	In Distribuzioni dei servizi, scegli l'ID di distribuzione.	

Viene visualizzata la pagina dei dettagli sulla distribuzione del servizio.

6. (Facoltativo) Confronta le revisioni del servizio per visualizzare le differenze.

In Revisioni del servizio, scegli Confronta revisioni, quindi seleziona 2 revisioni da confrontare.

Le revisioni del servizio vengono visualizzate side-by-side con le differenze evidenziate.

AWS CLI

1. Esegui `list-service-deployments` per recuperare l'ARN di distribuzione del servizio.

Sostituisci le variabili con i tuoi valori.

```
aws ecs list-service-deployments --cluster cluster-name --service service-name
```

Prendi nota della `serviceDeploymentArn` distribuzione che desideri visualizzare.

```
{
  "serviceDeployments": [
    {
      "serviceDeploymentArn": "arn:aws:ecs:us-west-2:123456789012:service-deployment/example/sd-example/NCWGC2ZR-taawPAYrIaU5",

```

```

        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/example/
sd-example",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/example",
        "targetServiceRevisionArn": "arn:aws:ecs:us-
west-2:123456789012:service-revision/example/sd-example/4980306466373577095",
        "status": "SUCCESSFUL"
    }
]
}

```

2. Esegui `describe-service-deployments`. Usa quello da `serviceDeploymentArn` cui è stato restituito `list-service-deployments`.

Sostituisci le variabili con i tuoi valori.

```

aws ecs describe-service-deployments --service-deployment-arns
arn:aws:ecs:region:123456789012:service-deployment/cluster-name/service-
name/NCWGC2ZR-taawPAYrIaU5

```

Passaggi successivi

È possibile visualizzare i dettagli delle revisioni dei servizi nella distribuzione. Per ulteriori informazioni, consulta [Visualizzazione dei dettagli di revisione del servizio Amazon ECS](#)

Revisioni dei servizi Amazon ECS

Una revisione del servizio contiene un record della configurazione del carico di lavoro che Amazon ECS sta tentando di implementare. Ogni volta che crei o distribuisce un servizio, Amazon ECS crea e acquisisce automaticamente la configurazione che stai cercando di implementare nella revisione del servizio.

Le revisioni dei servizi sono di sola lettura e hanno identificatori univoci. Per informazioni sulle proprietà incluse, vedere. [Proprietà incluse in una revisione del servizio Amazon ECS](#)

Le revisioni del servizio offrono i seguenti vantaggi:

- Durante l'implementazione di un servizio, è possibile confrontare la revisione del servizio attualmente distribuita (origine) con quella distribuita (destinazione).
- Quando utilizzi l'opzione di rollback per la distribuzione di un servizio, Amazon ECS ripristina automaticamente la distribuzione del servizio all'ultima revisione del servizio distribuita con successo.

- Le revisioni del servizio contengono il record della configurazione del carico di lavoro in un'unica risorsa.

Ciclo di vita della revisione del servizio

Amazon ECS crea automaticamente una nuova revisione del servizio quando crei un servizio o aggiorni una proprietà del servizio che avvia una distribuzione.

Amazon ECS non crea una nuova revisione del servizio per un'operazione di rollback. Amazon ECS utilizza l'ultima revisione del servizio riuscita per il rollback.

Una revisione del servizio è immutabile.

Amazon ECS elimina la revisione del servizio quando elimini un servizio.

Puoi visualizzare le revisioni dei servizi create il o dopo il 25 ottobre 2024 utilizzando la console, l'API e la CLI.

Proprietà incluse in una revisione del servizio Amazon ECS

Le seguenti proprietà sono incluse in una revisione del servizio.

Risorsa	Descrizione	
ARN del servizio	L'ARN che identifica il servizio.	
ARN del cluster	L'ARN per il cluster che ospita il servizio.	
ARN della definizione dell'attività	L'ARN della definizione dell'attività utilizzata per le attività di servizio.	
Registri dei servizi	I dettagli dei registri di servizio utilizzati per l'individuazione dei servizi. • Il nome del contenitore Il valore che indica il nome del container da utilizzare	

Risorsa	Descrizione	
	e per l'individuazione del servizio. • La porta del container Il valore che indica la porta da utilizzare per l'individuazione del servizio. Questo valore è specificato anche nella definizione dell'attività. • La porta Il valore della porta utilizzato se l'individuazione dei servizi ha specificato un record SRV. • Il registro ARN L'Amazon Resource Name (ARN) del registro del servizio.	

Risorsa	Descrizione	
Provider di capacità	I dettagli della strategia del fornitore di capacità. • Il nome del fornitore di capacità Il nome del fornitore di capacità utilizzato per il servizio. • Il peso del fornitore di capacità La percentuale relativa del numero totale di attività avviate che dovrebbero o utilizzare il fornitore di capacità specificato. • La base dei fornitori di capacità Il numero minimo di attività da eseguire sul provider di capacità specificato.	
Immagini di container	I dettagli sulle immagini del contenitore. • Il nome del contenitore • L'immagine digest • L'immagine del contenitore	

Risorsa	Descrizione	
Rete	La configurazione di rete per il servizio. • Se esiste un indirizzo IP pubblico • Le sottoreti in cui vengono eseguite le attività • I gruppi di sicurezza che controllano il traffico del servizio	
Tipo di avvio	Il tipo di avvio utilizzato per il servizio.	
Proprietà specifiche di Fargate	Quando il tipo di lancio è Fargate, si tratta di informazioni sulla versione di Fargate. • La versione della piattaforma e la famiglia di piattaforma • Archiviazione temporanea • La quantità di storage temporaneo da allocare per l'implementazione. • La chiave KMS da utilizzare per la crittografia dello storage.	

Risorsa	Descrizione	
Volumi Amazon EBS configurati al momento della distribuzione	La configurazione per un volume specificato nella definizione dell'attività come volume configurato al momento dell'avvio. • Crittografato Indica se il volume è crittografato. • Tipo di file system Il tipo di file system Linux per il volume. • IOPS Il numero di operazioni I/O al secondo (IOPS). • Chiave KMS • Ruolo IAM L'ARN del ruolo IAM da associare a questo volume. • Size La dimensione del volume in GiB. • ID snapshot L'ID dello snapshot del volume. • Specifiche del tag La configurazione dei tag per il volume.	

Risorsa	Descrizione	
	- Prestazioni La velocità effettiva prevista per il volume. - Tipo di volume Tipo di volume.	
Service Connect	La configurazione di Service Connect. - L'indicazione se Service Connect è in uso - Lo spazio dei nomi - L'elenco dei servizi interconnessi - La configurazione di registrazione di Service Connect	
Service Load Balancer	I sistemi di bilanciamento del carico che indirizzano il traffico di servizio. - Il nome - Il nome del contenitore - La porta del container - Il gruppo target ARN	
Monitoraggio del runtime	Indica se il monitoraggio del runtime è attivo.	
Data di creazione	La data di creazione della revisione del servizio.	

Risorsa	Descrizione	
VPC Lattice	La configurazione VPC Lattice per la revisione del servizio.	

Visualizzazione dei dettagli di revisione del servizio Amazon ECS

Puoi visualizzare informazioni sui seguenti tipi di revisione del servizio che sono stati creati a partire dal 25 ottobre 2024:

- Fonte: la configurazione del carico di lavoro attualmente distribuita
- Target: la configurazione del carico di lavoro in fase di implementazione

Amazon ECS Console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegli Implementazioni.
5. Scegli la revisione del servizio da visualizzare.

Per visualizzare le revisioni del servizio per questo tipo di distribuzione	Esegui questa operazione	
Implementazioni in corso	In Distribuzioni in corso, procedi come segue: • Per visualizzare la revisione del servizio di origine, in Revisioni del servizio, scegli l'ID di revisione del servizio per	

Per visualizzare le revisioni del servizio per questo tipo di distribuzione	Esegui questa operazione	
	il tipo di revisione dell'origine. Per visualizzare la revisione del servizio di destinazione, in Revisioni del servizio, scegli l'ID di revisione del servizio per il tipo di revisione di destinazione.	
Ultima distribuzione	In Ultima distribuzione, scegli la revisione del servizio Target.	
Distribuzioni completate	In Distribuzioni di servizi, procedi come segue: In Revisione del servizio Target, scegli l'ID.	

AWS CLI

1. Esegui `describe-service-deployments` per recuperare l'ARN della revisione del servizio.

Sostituisci le variabili con i tuoi valori.

```
aws ecs describe-service-deployments --service-deployment-arns
arn:aws:ecs:region:account-id:service/cluster-name/service-name/NCWGC2ZR-
taawPAYrIaU5
```

Nota il `arn` per il `sourceServiceRevisions` o il `targetServiceRevisions`.

```
{
```

```

    "serviceDeployments": [
      {
        "serviceDeploymentArn": "arn:aws:ecs:us-west-2:123456789012:service-
deployment/example/sd-example/NCWGC2ZR-taawPAYrIaU5",
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/example/
sd-example",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/example",
        "updatedAt": "2024-09-10T16:49:35.572000+00:00",
        "sourceServiceRevision": {
          "arn": "arn:aws:ecs:us-west-2:123456789012:service-revision/
example/sd-example/4980306466373578954",
          "requestedTaskCount": 0,
          "runningTaskCount": 0,
          "pendingTaskCount": 0
        },
        "targetServiceRevision": {
          "arn": "arn:aws:ecs:us-west-2:123456789012:service-revision/
example/sd-example/4980306466373577095",
          "requestedTaskCount": 0,
          "runningTaskCount": 0,
          "pendingTaskCount": 0
        },
        "status": "IN_PROGRESS",
        "deploymentConfiguration": {
          "deploymentCircuitBreaker": {
            "enable": false,
            "rollback": false
          },
          "maximumPercent": 200,
          "minimumHealthyPercent": 100
        }
      }
    ],
    "failures": []
  }

```

2. Esegui `describe-service-revisions`. Usa quello arn che è stato restituito da `describe-service-deployments`.

Sostituisci le variabili con i tuoi valori.


```
aws ecs describe-service-revisions --service-revision-arns
arn:aws:ecs:region:123456789012:service-revision/cluster-name/service-
name/4980306466373577095
```

Convalida lo stato di un servizio Amazon ECS prima della distribuzione

Il tipo di distribuzione blu/verde utilizza il modello di distribuzione blu/verde controllato da.

CodeDeploy Utilizza questo tipo di implementazione per verificare una nuova implementazione di un servizio prima di inviarti traffico di produzione. Per ulteriori informazioni, consulta [Cosa c'è CodeDeploy](#) nella guida per l'AWS CodeDeploy utente. Convalida lo stato di un servizio Amazon ECS prima della distribuzione

Esistono tre modi in cui il traffico può variare durante una distribuzione blu/verde:

- **Canarie:** il traffico viene spostato in due incrementi. Puoi scegliere tra opzioni canary predefinite che specificano la percentuale del traffico reinstradato al set di attività aggiornato nel primo incremento e l'intervallo, in minuti, prima che il traffico rimanente venga reinstradato nel secondo incremento.
- **Lineare:** il traffico viene spostato in incrementi uguali con un numero uguale di minuti tra ogni incremento. Puoi scegliere tra opzioni lineari predefinite che specificano la percentuale del traffico reinstradato in ogni incremento e l'intervallo di tempo, in minuti, tra ciascun incremento.
- **R II-at-once** — Tutto il traffico viene spostato contemporaneamente dal set di attività originale al set di attività aggiornato.

Di seguito sono riportati i componenti utilizzati da Amazon ECS quando un servizio utilizza il tipo di distribuzione blu/verde: CodeDeploy

CodeDeploy applicazione

Una raccolta di risorse. CodeDeploy È costituita da uno o più gruppi di distribuzione.

CodeDeploy gruppo di distribuzione

Le impostazioni di distribuzione. Comprendono:

- Cluster e servizio Amazon ECS
- Informazioni listener e gruppo target del load balancer
- Strategia di rollback dell'implementazione

- Impostazioni di reinstradamento del traffico
- Impostazioni di terminazione della revisione originale
- Configurazione dell'implementazione
- CloudWatch configurazione degli allarmi che può essere impostata per interrompere le distribuzioni
- Impostazioni SNS o CloudWatch Events per le notifiche

Per ulteriori informazioni, consulta [Utilizzo di gruppi di implementazione](#) nella Guida per l'utente di AWS CodeDeploy .

CodeDeploy configurazione della distribuzione

Specifica in che modo CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione. Sono disponibili le seguenti configurazioni predefinite di distribuzione lineare e canary. È inoltre possibile creare distribuzioni lineari e canary personalizzate. Per ulteriori informazioni, consulta [Utilizzo di configurazioni di implementazione](#) nella Guida per l'utente di AWS CodeDeploy .

- CodeDeployDefault. ECSAllAtOnce: sposta tutto il traffico verso il contenitore Amazon ECS aggiornato contemporaneamente
- CodeDeployDefault. ECSLinear10 PercentEvery 1Minuti: Sposta il 10% del traffico ogni minuto fino a quando tutto il traffico non viene spostato.
- CodeDeployDefault. ECSLinear10 PercentEvery 3 minuti: sposta il 10 per cento del traffico ogni 3 minuti fino a quando tutto il traffico non viene spostato.
- CodeDeployDefault. ECSCanary10Percent5Minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato cinque minuti più tardi.
- CodeDeployDefault. ECSCanary10Percent15Minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato 15 minuti più tardi.

Revisione

Una revisione è il file delle specifiche dell' CodeDeploy applicazione (file). AppSpec Nel AppSpec file, si specifica l'ARN completo della definizione dell'attività e il contenitore e la porta del set di attività sostitutivo in cui il traffico deve essere instradato quando viene creata una nuova distribuzione. Il nome del container deve essere uno dei nomi di container cui si fa riferimento nella definizione delle attività. Se la configurazione di rete o la versione della piattaforma è stata aggiornata nella definizione del servizio, è necessario specificare anche tali dettagli nel AppSpec file. Puoi inoltre specificare le funzioni Lambda da eseguire durante gli eventi del ciclo di vita dell'implementazione. Le funzioni Lambda consentono di eseguire i test e restituire i parametri

durante l'implementazione. Per ulteriori informazioni, consulta [AppSpec File Reference](#) nella Guida AWS CodeDeploy per l'utente.

Considerazioni

Quando utilizzi il tipo di distribuzione blu/verde, tieni conto di quanto segue:

- Quando crei per la prima volta un servizio Amazon ECS che utilizza il tipo di implementazione blu/verde, viene creato anche un set di processi Amazon ECS.
- Il servizio deve essere configurato per utilizzare un sistema Application Load Balancer o un Network Load Balancer. Di seguito sono elencati i requisiti del load balancer:
 - È necessario aggiungere al load balancer un listener di produzione che viene utilizzato per instradare il traffico di produzione.
 - Al load balancer può essere aggiunto un test opzionale che viene utilizzato per instradare il traffico di test. Se specifichi un listener di test, CodeDeploy indirizza il traffico di test all'attività sostitutiva impostata durante una distribuzione.
 - I listener di produzione e di test devono appartenere entrambi allo stesso load balancer.
 - È necessario definire un gruppo target per il load balancer. Il gruppo target instrada il traffico verso l'attività originale impostata in un servizio attraverso il listener di produzione.
 - Quando viene utilizzato un Network Load Balancer, è supportata solo la configurazione dell'implementazione `CodeDeployDefault.ECSAllAtOnce`.
- Per i servizi configurati per utilizzare la scalabilità automatica del servizio e il tipo di implementazione blu/verde, la scalabilità automatica non viene bloccata durante un'implementazione, ma l'implementazione potrebbe non riuscire in alcune circostanze. Di seguito viene descritto questo comportamento in modo più dettagliato.
 - Se un servizio è scalabile e viene avviata una distribuzione, viene creato il set di attività verde e CodeDeploy aspetterà fino a un'ora prima che il set di attività verde raggiunga lo stato stazionario e non sposterà il traffico finché non lo farà.
 - Se un servizio è in fase di implementazione blu/verde e si verifica un evento di dimensionamento, il traffico continuerà a spostarsi per 5 minuti. Se il servizio non raggiunge lo stato stazionario entro 5 minuti, CodeDeploy interromperà la distribuzione e la contrassegnerà come fallita.
 - Se un servizio è in fase di implementazione blu/verde e si verifica un evento di dimensionamento, il numero dei processi desiderato potrebbe essere impostato su un valore imprevisto. Ciò è causato dal dimensionamento automatico che considera il numero dei processi

in esecuzione come capacità corrente, ovvero il doppio del numero appropriato di processi utilizzati nel calcolo del numero dei processi desiderato.

- I processi che utilizzano il tipo di avvio Fargate oppure i tipi di controller di implementazione CODE_DEPLOY non supportano la strategia di pianificazione DAEMON.
- Quando si crea inizialmente un' CodeDeploy applicazione e un gruppo di distribuzione, è necessario specificare quanto segue:
 - È necessario definire due gruppi target per il load balancer. Un gruppo di destinazione deve essere il gruppo di destinazione iniziale definito per il load balancer quando il servizio Amazon ECS è stato creato. L'unico requisito del secondo gruppo target è che non può essere associato a un load balancer diverso rispetto a quello utilizzato dal servizio.
- Quando crei una CodeDeploy distribuzione per un servizio Amazon ECS, CodeDeploy crea un set di attività sostitutivo (o set di attività verde) nella distribuzione. Se hai aggiunto un listener di test al load balancer, CodeDeploy indirizza il traffico di test verso il set di attività sostitutivo. Questo è il momento in cui è possibile eseguire i test di convalida. Quindi CodeDeploy reindirizza il traffico di produzione dal set di attività originale al set di attività di sostituzione in base alle impostazioni di reinstradamento del traffico per il gruppo di distribuzione.

Autorizzazioni IAM richieste

Blue/green deployments are made possible by a combination of the Amazon ECS and CodeDeploy APIs. Users must have the appropriate permissions for these services before they can use Amazon ECS blue/greendistribuzioni in o con AWS Management Console o. AWS CLI SDKs

Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, Amazon ECS richiede le seguenti autorizzazioni. Queste autorizzazioni sono state aggiunte alla policy IAM AmazonECS_FullAccess. Per ulteriori informazioni, consulta [Amazon ECS_FullAccess](#).

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "codedeploy:CreateApplication",
        "codedeploy:CreateDeployment",
        "codedeploy:CreateDeploymentGroup",
        "codedeploy:GetApplication",
        "codedeploy:GetDeployment",

```

```

        "codedeploy:GetDeploymentGroup",
        "codedeploy:ListApplications",
        "codedeploy:ListDeploymentGroups",
        "codedeploy:ListDeployments",
        "codedeploy:StopDeployment",
        "codedeploy:GetDeploymentTarget",
        "codedeploy:ListDeploymentTargets",
        "codedeploy:GetDeploymentConfig",
        "codedeploy:GetApplicationRevision",
        "codedeploy:RegisterApplicationRevision",
        "codedeploy:BatchGetApplicationRevisions",
        "codedeploy:BatchGetDeploymentGroups",
        "codedeploy:BatchGetDeployments",
        "codedeploy:BatchGetApplications",
        "codedeploy:ListApplicationRevisions",
        "codedeploy:ListDeploymentConfigs",
        "codedeploy:ContinueDeployment",
        "sns:ListTopics",
        "cloudwatch:DescribeAlarms",
        "lambda:ListFunctions"
    ],
    "Resource": ["*"]
}
]
}

```

Note

Oltre alle autorizzazioni Amazon ECS standard necessarie per eseguire processi e servizi, gli utenti hanno bisogno anche delle autorizzazioni `iam:PassRole` per utilizzare i ruoli IAM per le attività.

CodeDeploy necessita delle autorizzazioni per chiamare Amazon ECS APIs, modificare Elastic Load Balancing, richiamare le funzioni Lambda e CloudWatch descrivere gli allarmi, oltre alle autorizzazioni per modificare il conteggio desiderato del servizio per tuo conto. Prima di creare un servizio Amazon ECS che usa il tipo di implementazione blu/verde, occorre creare un ruolo IAM (`ecsCodeDeployRole`). Per ulteriori informazioni, consulta [Ruolo CodeDeploy IAM di Amazon ECS](#).

Gli esempi di policy IAM [Crea un esempio di servizio Amazon ECS](#) e [Esempio di aggiornamento del servizio Amazon ECS](#) mostrano le autorizzazioni necessarie affinché gli utenti possano utilizzare le implementazioni blu/verde di Amazon ECS nella AWS Management Console.

Implementazione di un servizio Amazon ECS utilizzando una distribuzione blu/verde

Scopri come creare un servizio Amazon ECS contenente un'attività Fargate che utilizza il tipo di distribuzione blu/verde con. AWS CLI

Note

È stato aggiunto il supporto per l'esecuzione di un'implementazione blu/verde per AWS CloudFormation. Per ulteriori informazioni, consulta [Eseguire le distribuzioni blu/green di Amazon ECS tramite l' CodeDeploy utilizzo AWS CloudFormation nella Guida per l'utente.AWS CloudFormation](#)

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere [Installazione o aggiornamento alla versione più recente di. AWS CLI](#)

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy IAM. [Amazon ECS_FullAccess](#)
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).
- Viene creato il ruolo CodeDeploy IAM di Amazon ECS. Per ulteriori informazioni, consulta [Ruolo CodeDeploy IAM di Amazon ECS](#).

Fase 1: Creazione di un Application Load Balancer

I servizi di Amazon ECS che utilizzano il tipo di implementazione blu/verde richiedono l'utilizzo di un servizio di Application Load Balancer o di un Network Load Balancer. In questa esercitazione viene utilizzato un Application Load Balancer.

Per creare un Application Load Balancer

1. Utilizzate il [create-load-balancer](#) comando per creare un Application Load Balancer. Specificare due sottoreti che non appartengono alla stessa zona di disponibilità, nonché un gruppo di sicurezza.

```
aws elbv2 create-load-balancer \  
  --name bluegreen-alb \  
  --subnets subnet-abcd1234 subnet-abcd5678 \  
  --security-groups sg-abcd1234 \  
  --region us-east-1
```

L'output include l'Amazon Resource Name (ARN) del load balancer, con il formato seguente:

```
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/  
e5ba62739c16e642
```

2. Utilizzate il [create-target-group](#) comando per creare un gruppo target. Questo gruppo target instraderà il traffico verso il set di attività originale nel servizio.

```
aws elbv2 create-target-group \  
  --name bluegreentarget1 \  
  --protocol HTTP \  
  --port 80 \  
  --target-type ip \  
  --vpc-id vpc-abcd1234 \  
  --region us-east-1
```

L'output include l'ARN del gruppo target, con il seguente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/  
bluegreentarget1/209a844cd01825a4
```

3. Utilizzare il comando [create-listener](#) per creare un listener del load balancer con una regola predefinita che inoltra le richieste al gruppo target.

```
aws elbv2 create-listener \  
  --load-balancer-arn  
  arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/  
e5ba62739c16e642 \  
  --protocol HTTP \  
  --port 80 \  
  --default-actions  
  Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/  
bluegreentarget1/209a844cd01825a4 \  
  --region us-east-1
```

L'output include l'ARN del listener, con il formato seguente:

```
arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/  
e5ba62739c16e642/665750bec1b03bd4
```

Fase 2: Creazione di un cluster Amazon ECS

Utilizzare il comando [create-cluster](#) per creare un cluster denominato `tutorial-bluegreen-cluster` da utilizzare.

```
aws ecs create-cluster \  
  --cluster-name tutorial-bluegreen-cluster \  
  --region us-east-1
```

L'output include l'ARN del cluster, con il formato seguente:

```
arn:aws:ecs:region:aws_account_id:cluster/tutorial-bluegreen-cluster
```

Fase 3: Registra una definizione di attività

Utilizzate il [register-task-definition](#) comando per registrare una definizione di attività compatibile con Fargate. Devi usare la modalità di rete `awsvpc`. Di seguito è riportata la definizione di attività di esempio usata per questo tutorial.

Crea innanzitutto un file denominato `fargate-task.json` con i seguenti contenuti. Assicurati di utilizzare l'ARN del ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).


```
{
  "family": "sample-fargate",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "sample-app",
      "image": "public.ecr.aws/docker/library/httpd:latest",
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
          "protocol": "tcp"
        }
      ],
      "essential": true,
      "entryPoint": [
        "sh",
        "-c"
      ],
      "command": [
        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-foreground\""]
      ]
    }
  ],
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "cpu": "256",
  "memory": "512"
}
```

Quindi, registra la definizione di attività utilizzando il file `fargate-task.json` creato.

```
aws ecs register-task-definition \
  --cli-input-json file:///fargate-task.json \
  --region us-east-1
```

Fase 4: Creazione di un servizio Amazon ECS

Utilizza il comando [create-service](#) per creare un servizio.

Crea innanzitutto un file denominato `service-bluegreen.json` con i seguenti contenuti.

```
{
  "cluster": "tutorial-bluegreen-cluster",
  "serviceName": "service-bluegreen",
  "taskDefinition": "tutorial-task-def",
  "loadBalancers": [
    {
      "targetGroupArn":
"arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4",
      "containerName": "sample-app",
      "containerPort": 80
    }
  ],
  "launchType": "FARGATE",
  "schedulingStrategy": "REPLICA",
  "deploymentController": {
    "type": "CODE_DEPLOY"
  },
  "platformVersion": "LATEST",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "assignPublicIp": "ENABLED",
      "securityGroups": [ "sg-abcd1234" ],
      "subnets": [ "subnet-abcd1234", "subnet-abcd5678" ]
    }
  },
  "desiredCount": 1
}
```

Quindi, crea il servizio utilizzando il file `service-bluegreen.json` creato.

```
aws ecs create-service \
  --cli-input-json file://service-bluegreen.json \
  --region us-east-1
```

L'output include l'ARN del servizio, con il formato seguente:

```
arn:aws:ecs:region:aws_account_id:service/service-bluegreen
```

Ottieni il nome DNS del sistema di bilanciamento del carico utilizzando il comando seguente.

```
aws elbv2 describe-load-balancers --name bluegreen-alb --query  
'LoadBalancers[*].DNSName'
```

Immetti il nome DNS nel browser Web e dovresti visualizzare una pagina Web che mostra l'applicazione di esempio con uno sfondo blu.

Fase 5: Creazione delle risorse AWS CodeDeploy

Utilizza i seguenti passaggi per creare CodeDeploy l'applicazione, il gruppo target Application Load Balancer per il gruppo di CodeDeploy distribuzione e il gruppo di CodeDeploy distribuzione.

Per creare risorse CodeDeploy

1. Utilizzate il comando [create-application](#) per creare un' CodeDeploy applicazione. Specificare la piattaforma di calcolo ECS.

```
aws deploy create-application \  
  --application-name tutorial-bluegreen-app \  
  --compute-platform ECS \  
  --region us-east-1
```

L'output include l'ID applicazione, con il formato seguente:

```
{  
  "applicationId": "b8e9c1ef-3048-424e-9174-885d7dc9dc11"  
}
```

2. Utilizzate il [create-target-group](#) comando per creare un secondo gruppo target di Application Load Balancer, che verrà utilizzato durante la creazione del gruppo di CodeDeploy distribuzione.

```
aws elbv2 create-target-group \  
  --name bluegreentarget2 \  
  --protocol HTTP \  
  --port 80 \  
  --target-type ip \  
  --vpc-id "vpc-0b6dd82c67d8012a1" \  
  --health-check-path /
```

```
--region us-east-1
```

L'output include l'ARN per il gruppo target, con il seguente formato:

```
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/  
bluegreentarget2/708d384187a3cfdc
```

3. Utilizzate il [create-deployment-group](#) comando per creare un gruppo CodeDeploy di distribuzione.

Crea innanzitutto un file denominato `tutorial-deployment-group.json` con i seguenti contenuti. Questo esempio utilizza la risorsa che è stata creata. Per `laServiceRoleArn`, specifica l'ARN del tuo ruolo Amazon ECS IAM. CodeDeploy Per ulteriori informazioni, consulta [Ruolo CodeDeploy IAM di Amazon ECS](#).

```
{  
  "applicationName": "tutorial-bluegreen-app",  
  "autoRollbackConfiguration": {  
    "enabled": true,  
    "events": [ "DEPLOYMENT_FAILURE" ]  
  },  
  "blueGreenDeploymentConfiguration": {  
    "deploymentReadyOption": {  
      "actionOnTimeout": "CONTINUE_DEPLOYMENT",  
      "waitTimeInMinutes": 0  
    },  
    "terminateBlueInstancesOnDeploymentSuccess": {  
      "action": "TERMINATE",  
      "terminationWaitTimeInMinutes": 5  
    }  
  },  
  "deploymentGroupName": "tutorial-bluegreen-dg",  
  "deploymentStyle": {  
    "deploymentOption": "WITH_TRAFFIC_CONTROL",  
    "deploymentType": "BLUE_GREEN"  
  },  
  "loadBalancerInfo": {  
    "targetGroupPairInfoList": [  
      {  
        "targetGroups": [  
          {  
            "name": "bluegreentarget1"  
          }  
        ]  
      }  
    ]  
  }  
}
```

```

        {
            "name": "bluegreentarget2"
        },
        "prodTrafficRoute": {
            "listenerArns": [
                "arn:aws:elasticloadbalancing:region:aws_account_id:listener/app/bluegreen-alb/e5ba62739c16e642/665750bec1b03bd4"
            ]
        }
    ],
    "serviceRoleArn": "arn:aws:iam::aws_account_id:role/ecsCodeDeployRole",
    "ecsServices": [
        {
            "serviceName": "service-bluegreen",
            "clusterName": "tutorial-bluegreen-cluster"
        }
    ]
}

```

Quindi crea il gruppo di CodeDeploy distribuzione.

```

aws deploy create-deployment-group \
  --cli-input-json file://tutorial-deployment-group.json \
  --region us-east-1

```

L'output include l'ID del gruppo di distribuzione, con il formato seguente:

```

{
  "deploymentGroupId": "6fd9bdc6-dc51-4af5-ba5a-0a4a72431c88"
}

```

Fase 6: Creare e monitorare una CodeDeploy distribuzione

Prima di creare una CodeDeploy distribuzione, aggiorna la definizione command dell'attività `fargate-task.json` come segue per modificare il colore di sfondo dell'app di esempio in verde.

```

{
  ...

```

```

    "containerDefinitions": [
      {
        ...
        "command": [
          "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color: #097969;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
<h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-
foreground\""]
        ]
      },
      ...
    ]
  }
}

```

Registra la definizione di attività aggiornata utilizzando il comando seguente.

```

aws ecs register-task-definition \
  --cli-input-json file://fargate-task.json \
  --region us-east-1

```

Ora, utilizzate i seguenti passaggi per creare e caricare un file di specifiche dell'applicazione (AppSpec file) e una CodeDeploy distribuzione.

Per creare e monitorare una CodeDeploy distribuzione

1. Crea e carica un AppSpec file utilizzando i seguenti passaggi.
 - a. Creare un file denominato `appspec.yaml` con il contenuto del gruppo di distribuzione CodeDeploy . Questo esempio utilizza la definizione di attività aggiornata.

```

version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:region:aws_account_id:task-
definition/tutorial-task-def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80

```

```
PlatformVersion: "LATEST"
```

- b. Usa il comando [s3 mb](#) per creare un bucket Amazon S3 per il file. AppSpec

```
aws s3 mb s3://tutorial-bluegreen-bucket
```

- c. Usa il comando [s3 cp](#) per caricare il AppSpec file nel bucket Amazon S3.

```
aws s3 cp ./appspec.yaml s3://tutorial-bluegreen-bucket/appspec.yaml
```

2. Crea la CodeDeploy distribuzione utilizzando i seguenti passaggi.

- a. Crea un file denominato `create-deployment.json` con il contenuto della CodeDeploy distribuzione. Questo esempio utilizza le risorse create in precedenza nel tutorial.

```
{
  "applicationName": "tutorial-bluegreen-app",
  "deploymentGroupName": "tutorial-bluegreen-dg",
  "revision": {
    "revisionType": "S3",
    "s3Location": {
      "bucket": "tutorial-bluegreen-bucket",
      "key": "appspec.yaml",
      "bundleType": "YAML"
    }
  }
}
```

- b. Utilizzare il comando [create-deployment](#) per creare l'implementazione.

```
aws deploy create-deployment \
  --cli-input-json file://create-deployment.json \
  --region us-east-1
```

L'output include l'ID di distribuzione, con il formato seguente:

```
{
  "deploymentId": "d-RPCR1U3TW"
}
```

3. Utilizzate il [get-deployment-target](#) comando per ottenere i dettagli della distribuzione, specificando l'`deploymentId` output precedente.

```
aws deploy get-deployment-target \
--deployment-id "d-IMJU3A8TW" \
--target-id tutorial-bluegreen-cluster:service-bluegreen \
--region us-east-1
```

Inizialmente, lo stato dell'implementazione è InProgress. Il traffico viene indirizzato al set di attività originale, che presenta taskSetLabel con valore BLUE, uno stato PRIMARY e un trafficWeight di 100.0. Il set di attività di sostituzione ha taskSetLabel con valore GREEN, uno stato ACTIVE e un trafficWeight di 0.0. Il browser Web in cui hai inserito il nome DNS mostra ancora l'app di esempio con uno sfondo blu.

```
{
  "deploymentTarget": {
    "deploymentTargetType": "ECSTarget",
    "ecsTarget": {
      "deploymentId": "d-RPCR1U3TW",
      "targetId": "tutorial-bluegreen-cluster:service-bluegreen",
      "targetArn": "arn:aws:ecs:region:aws_account_id:service/service-bluegreen",
      "lastUpdatedAt": "2023-08-10T12:07:24.797000-05:00",
      "lifecycleEvents": [
        {
          "lifecycleEventName": "BeforeInstall",
          "startTime": "2023-08-10T12:06:22.493000-05:00",
          "endTime": "2023-08-10T12:06:22.790000-05:00",
          "status": "Succeeded"
        },
        {
          "lifecycleEventName": "Install",
          "startTime": "2023-08-10T12:06:22.936000-05:00",
          "status": "InProgress"
        },
        {
          "lifecycleEventName": "AfterInstall",
          "status": "Pending"
        },
        {
          "lifecycleEventName": "BeforeAllowTraffic",
          "status": "Pending"
        },
        {
          "lifecycleEventName": "AllowTraffic",
```



```

        "status": "Pending"
    },
    {
        "lifecycleEventName": "AfterAllowTraffic",
        "status": "Pending"
    }
],
"status": "InProgress",
"taskSetsInfo": [
    {
        "identifer": "ecs-svc/9223370493423413672",
        "desiredCount": 1,
        "pendingCount": 0,
        "runningCount": 1,
        "status": "ACTIVE",
        "trafficWeight": 0.0,
        "targetGroup": {
            "name": "bluegreentarget2"
        },
        "taskSetLabel": "Green"
    },
    {
        "identifer": "ecs-svc/9223370493425779968",
        "desiredCount": 1,
        "pendingCount": 0,
        "runningCount": 1,
        "status": "PRIMARY",
        "trafficWeight": 100.0,
        "targetGroup": {
            "name": "bluegreentarget1"
        },
        "taskSetLabel": "Blue"
    }
]
}
}
}

```

Continua a recuperare i dettagli di implementazione utilizzando il comando fino a quando lo stato è Succeeded, come mostrato nel seguente output. Il traffico viene ora instradato al set di attività di sostituzione, che presenta uno stato PRIMARY e un trafficWeight di 100.0.

Aggiorna il browser Web in cui hai inserito il nome DNS del sistema di bilanciamento del carico e ora dovresti visualizzare l'app di esempio con uno sfondo verde.

```
{
  "deploymentTarget": {
    "deploymentTargetType": "ECSTarget",
    "ecsTarget": {
      "deploymentId": "d-RPCR1U3TW",
      "targetId": "tutorial-bluegreen-cluster:service-bluegreen",
      "targetArn": "arn:aws:ecs:region:aws_account_id:service/service-bluegreen",
      "lastUpdatedAt": "2023-08-10T12:07:24.797000-05:00",
      "lifecycleEvents": [
        {
          "lifecycleEventName": "BeforeInstall",
          "startTime": "2023-08-10T12:06:22.493000-05:00",
          "endTime": "2023-08-10T12:06:22.790000-05:00",
          "status": "Succeeded"
        },
        {
          "lifecycleEventName": "Install",
          "startTime": "2023-08-10T12:06:22.936000-05:00",
          "endTime": "2023-08-10T12:08:25.939000-05:00",
          "status": "Succeeded"
        },
        {
          "lifecycleEventName": "AfterInstall",
          "startTime": "2023-08-10T12:08:26.089000-05:00",
          "endTime": "2023-08-10T12:08:26.403000-05:00",
          "status": "Succeeded"
        },
        {
          "lifecycleEventName": "BeforeAllowTraffic",
          "startTime": "2023-08-10T12:08:26.926000-05:00",
          "endTime": "2023-08-10T12:08:27.256000-05:00",
          "status": "Succeeded"
        },
        {
          "lifecycleEventName": "AllowTraffic",
          "startTime": "2023-08-10T12:08:27.416000-05:00",
          "endTime": "2023-08-10T12:08:28.195000-05:00",
          "status": "Succeeded"
        }
      ]
    }
  }
}
```

```

        "lifecycleEventName": "AfterAllowTraffic",
        "startTime": "2023-08-10T12:08:28.715000-05:00",
        "endTime": "2023-08-10T12:08:28.994000-05:00",
        "status": "Succeeded"
    }
],
"status": "Succeeded",
"taskSetsInfo": [
    {
        "identifer": "ecs-svc/9223370493425779968",
        "desiredCount": 1,
        "pendingCount": 0,
        "runningCount": 1,
        "status": "ACTIVE",
        "trafficWeight": 0.0,
        "targetGroup": {
            "name": "bluegreentarget1"
        },
        "taskSetLabel": "Blue"
    },
    {
        "identifer": "ecs-svc/9223370493423413672",
        "desiredCount": 1,
        "pendingCount": 0,
        "runningCount": 1,
        "status": "PRIMARY",
        "trafficWeight": 100.0,
        "targetGroup": {
            "name": "bluegreentarget2"
        },
        "taskSetLabel": "Green"
    }
]
}
}
}

```

Fase 7: pulire

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi.

Rimozione delle risorse del tutorial

1. Usa il [delete-deployment-group](#) comando per eliminare il gruppo CodeDeploy di distribuzione.

```
aws deploy delete-deployment-group \  
  --application-name tutorial-bluegreen-app \  
  --deployment-group-name tutorial-bluegreen-dg \  
  --region us-east-1
```

2. Utilizzate il comando [delete-application](#) per eliminare l' CodeDeploy applicazione.

```
aws deploy delete-application \  
  --application-name tutorial-bluegreen-app \  
  --region us-east-1
```

3. Utilizza il comando [delete-service](#) per eliminare il servizio Amazon ECS. L'uso del flag `--force` consente di eliminare un servizio anche se il numero di attività non è stato ridotto a zero.

```
aws ecs delete-service \  
  --service arn:aws:ecs:region:aws_account_id:service/service-bluegreen \  
  --force \  
  --region us-east-1
```

4. Utilizza il comando [delete-cluster](#) per eliminare il cluster Amazon ECS.

```
aws ecs delete-cluster \  
  --cluster tutorial-bluegreen-cluster \  
  --region us-east-1
```

5. Usa il comando [s3 rm](#) per eliminare il AppSpec file dal bucket Amazon S3.

```
aws s3 rm s3://tutorial-bluegreen-bucket/appspec.yaml
```

6. Utilizza il comando [s3 rb](#) per eliminare il bucket Amazon S3.

```
aws s3 rb s3://tutorial-bluegreen-bucket
```

7. Utilizzate il [delete-load-balancer](#) comando per eliminare l'Application Load Balancer.

```
aws elbv2 delete-load-balancer \  
  --load-balancer-arn arn:aws:elasticloadbalancing:us-east-1:123456789012:loadbalancer/app/tutorial-bluegreen-lb
```

```
--load-balancer-arn  
arn:aws:elasticloadbalancing:region:aws_account_id:loadbalancer/app/bluegreen-alb/  
e5ba62739c16e642 \  
--region us-east-1
```

8. Utilizzate il [delete-target-group](#) comando per eliminare i due gruppi target di Application Load Balancer.

```
aws elbv2 delete-target-group \  
--target-group-arn  
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/  
bluegreentarget1/209a844cd01825a4 \  
--region us-east-1
```

```
aws elbv2 delete-target-group \  
--target-group-arn  
arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/  
bluegreentarget2/708d384187a3cfdc \  
--region us-east-1
```

Implementa i servizi Amazon ECS utilizzando un controller di terze parti

Il tipo di implementazione esterna consente di usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implementazione per un servizio Amazon ECS. I dettagli del servizio vengono gestiti tramite le operazioni API del servizio di gestione (CreateService, UpdateService e DeleteService) o della gestione dei set di attività (CreateTaskSet, UpdateTaskSet, UpdateServicePrimaryTaskSet e DeleteTaskSet). Ciascuna operazione API gestisce un sottoinsieme di parametri di definizione del servizio.

L'operazione API UpdateService aggiorna i parametri del periodo di tolleranza per il conteggio desiderato e il controllo dello stato di un servizio. Se è necessario aggiornare il tipo di avvio, la versione della piattaforma, i dettagli del load balancer, la configurazione di rete o la definizione di attività, occorre creare un nuovo set di attività.

L'operazione API UpdateTaskSet aggiorna solo il parametro di dimensionamento per un set di attività.

L'operazione API UpdateServicePrimaryTaskSet modifica il set di attività impostato come principale in un servizio. Chiamando l'operazione API DescribeServices, vengono restituiti tutti

i campi specificati per un set di attività principale. Se viene aggiornato il set di attività principale per un servizio, quando viene definito il nuovo set principale eventuali valori del parametro esistenti sul nuovo ma diversi da quelli impostati sul precedente set di attività in un servizio verranno aggiornati ai nuovi valori. Se non viene definito un set di attività principale per un servizio, i relativi campi nella descrizione del servizio sono nulli.

Considerazioni sull'implementazione esterna

Quando utilizzi il tipo di distribuzione esterna, tieni conto di quanto segue:

- I tipi di load balancer supportati sono un Application Load Balancer o un Network Load Balancer.
- Il tipo di avvio Fargate oppure i tipi di controller di implementazione EXTERNAL non supportano la strategia di pianificazione DAEMON.

Flusso di lavoro dell'implementazione esterna

Di seguito è riportato il flusso di lavoro di base per la gestione di una distribuzione esterna su Amazon ECS.

Come gestire un servizio Amazon ECS tramite un controller di implementazione esterno

1. Crea un servizio Amazon ECS. L'unico parametro obbligatorio è il nome del servizio. Durante la creazione di un servizio tramite un controller di distribuzione esterno, è possibile specificare i parametri indicati di seguito. Tutti gli altri, invece, vengono specificati al momento della creazione di un set di attività nel servizio.

`serviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del servizio. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. I nomi dei servizi devono essere univoci all'interno di un cluster, ma puoi avere servizi dai nomi simili in più cluster all'interno di una Regione o in più Regioni.

desiredCount

Il numero di istanze della definizione di attività per il set specificato, da posizionare e mantenere in esecuzione nel servizio.

deploymentConfiguration

I parametri di distribuzione opzionali che determinano quante attività vengono eseguite durante un'implementazione e l'ordine di arresto e di avvio delle attività.

tags

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati applicati al servizio per aiutarti a catalogarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Quando un servizio viene eliminato, vengono eliminati anche i tag. Al servizio è possibile applicare un massimo di 50 tag. Per ulteriori informazioni, consulta [Taggare le risorse Amazon ECS](#).

key

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

value

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

enableECSTags

Specifica se usare i tag gestiti di Amazon ECS per i processi all'interno del servizio. Per ulteriori informazioni, consulta [Utilizza i tag per la fatturazione](#).

propagateTags

Tipo: stringa

Valori validi: TASK_DEFINITION | SERVICE

Campo obbligatorio: no

Specifica se copiare i tag dalla definizione di attività o dal servizio nelle attività del servizio. Se non viene specificato alcun valore, i tag non vengono copiati. I tag possono essere copiati solo nelle attività all'interno del servizio durante la creazione del servizio. Per aggiungere tag a un processo dopo la creazione di un servizio o di un processo, utilizza l'operazione API `TagResource`.

schedulingStrategy

La strategia di pianificazione da utilizzare. I servizi che utilizzano un controller di distribuzione esterno supportano solo la strategia di pianificazione `REPLICA`.

placementConstraints

Serie di oggetti vincolo di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di 10 vincoli per attività (questo limite include i vincoli nella definizione di attività e quelli specificati in fase di runtime). Se utilizzi il tipo di avvio Fargate, i vincoli di posizionamento dei processi non sono supportati.

placementStrategy

Gli oggetti strategia di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di quattro regole di strategia per ogni servizio.

Di seguito è illustrato un esempio di definizione per la creazione di un servizio che utilizza un controller di distribuzione esterno.

```
{
  "cluster": "",
  "serviceName": "",
```



```
"desiredCount": 0,
"role": "",
"deploymentConfiguration": {
  "maximumPercent": 0,
  "minimumHealthyPercent": 0
},
"placementConstraints": [
  {
    "type": "distinctInstance",
    "expression": ""
  }
],
"placementStrategy": [
  {
    "type": "binpack",
    "field": ""
  }
],
"schedulingStrategy": "REPLICA",
"deploymentController": {
  "type": "EXTERNAL"
},
"tags": [
  {
    "key": "",
    "value": ""
  }
],
"enableECSTags": true,
"propagateTags": "TASK_DEFINITION"
}
```

2. Creare un set di attività iniziali. Il set di attività contiene i seguenti i dettagli sul servizio:

taskDefinition

La definizione delle attività nel set da utilizzare.

launchType

Tipo: stringa

Valori validi: EC2 | FARGATE | EXTERNAL

Campo obbligatorio: no

Il tipo di avvio con cui eseguire il servizio. Se non viene specificato un tipo di avvio, per impostazione predefinita viene utilizzato il `capacityProviderStrategy` di default. Per ulteriori informazioni, consulta [Tipi di avvio di Amazon ECS](#).

Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

`platformVersion`

Tipo: string

Campo obbligatorio: no

La versione della piattaforma su cui sono in esecuzione le attività nel servizio. Viene specificata una versione della piattaforma solo per le attività con tipo di avvio Fargate. Se non è specificata, la versione più recente (LATEST) viene utilizzata di default.

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Quando specifichi la versione della piattaforma LATEST durante l'esecuzione di un'attività o la creazione di un servizio, ottieni la versione di piattaforma più aggiornata disponibile per le tue attività. Quando incrementi il servizio, tali attività riceveranno la versione della piattaforma specificata nell'implementazione corrente del servizio. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Note

Le versioni della piattaforma non sono specificate per le attività che utilizzano il tipo di avvio EC2.

`loadBalancers`

Un oggetto che rappresenta il load balancer da utilizzare con il tuo servizio. Quando si utilizza un controller di implementazione esterno, sono supportati solo Application Load Balancer e il Network Load Balancer. Se si utilizza un Application Load Balancer, è consentito un solo gruppo di destinazione di Application Load Balancer per ogni set di processi.

Il frammento di codice seguente mostra un oggetto `loadBalancer` di esempio da utilizzare.

```
"loadBalancers": [  
  {  
    "targetGroupArn": "",  
    "containerName": "",  
    "containerPort": 0  
  }  
]
```

 Note

Quando si specifica un oggetto `loadBalancer`, è necessario specificare `targetGroupArn` e omettere i parametri `loadBalancerName`.

`networkConfiguration`

La configurazione di rete per il servizio. Questo parametro è obbligatorio per le definizioni di attività che utilizzano la modalità di rete `awsvpc` per ricevere la propria interfaccia di rete elastica e non è supportato per altre modalità. Per ulteriori informazioni sulla rete per il tipo di lancio Fargate, vedere. [Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#)

`serviceRegistries`

I dettagli dei registri del servizio di individuazione da assegnare a questo servizio. Per ulteriori informazioni, consulta [Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS](#).

`scale`

Percentuale a virgola mobile del numero desiderato di attività da posizionare e mantenere in esecuzione nel set di attività. Il valore è specificato come percentuale totale del conteggio desiderato (`desiredCount`) di un servizio. Come valori sono accettati i numeri da 0 a 100.

Di seguito è riportato un esempio JSON per la creazione di un set di attività per un controller di distribuzione esterno.

```
{
```

```
"service": "",
"cluster": "",
"externalId": "",
"taskDefinition": "",
"networkConfiguration": {
  "awsvpcConfiguration": {
    "subnets": [
      ""
    ],
    "securityGroups": [
      ""
    ],
    "assignPublicIp": "DISABLED"
  }
},
"loadBalancers": [
  {
    "targetGroupArn": "",
    "containerName": "",
    "containerPort": 0
  }
],
"serviceRegistries": [
  {
    "registryArn": "",
    "port": 0,
    "containerName": "",
    "containerPort": 0
  }
],
"launchType": "EC2",
"capacityProviderStrategy": [
  {
    "capacityProvider": "",
    "weight": 0,
    "base": 0
  }
],
"platformVersion": "",
"scale": {
  "value": null,
  "unit": "PERCENT"
},
"clientToken": ""
```

```
}
```

- Quando è necessario modificare il servizio, utilizzare l'operazione API `CreateTaskSet`, `UpdateService` o `UpdateTaskSet`, a seconda dei parametri da aggiornare. Se è stato creato un set di attività, utilizzare il parametro `scale` per ogni set di attività in un servizio, per determinare il numero di attività da mantenere in esecuzione nel servizio. Ad esempio, se si dispone di un servizio che contiene un `tasksetA` e si crea un `tasksetB`, è consigliabile testare la validità del `tasksetB` prima di trasferire a esso il traffico di produzione. È possibile impostare il parametro `scale` per entrambi i set di attività su 100e, una volta pronti a spostare tutto il traffico di produzione nel `tasksetB`, è possibile aggiornare `scale` per `tasksetA` su 0 per ridurlo.

Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS

Il servizio può essere configurato opzionalmente per utilizzare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività del servizio.

Note

Quando si utilizzano set di processi, tutti i processi del set devono essere configurati per utilizzare Elastic Load Balancing o non utilizzare Elastic Load Balancing.

I servizi Amazon ECS ospitati su AWS Fargate supportano Application Load Balancer, Network Load Balancer e Gateway Load Balancer. Utilizza la tabella seguente per scoprire quale tipo di bilanciamento del carico utilizzare.

Tipo di Load Balancer	Utilizzare in questi casi	
Application Load Balancer	Indirizza il traffico HTTP/HTTPS (o livello 7). Gli Application Load Balancer offrono diverse funzionalità che li rendono particolarmente	

Tipo di Load Balancer	Utilizzare in questi casi	
	appropriati per l'uso con i servizi Amazon ECS: • Ogni servizio è in grado di servire traffico proveniente da più load balancer ed esporre più porte con carico bilanciato specificando più gruppi di destinazione. • Sono supportati da attività ospitate sia su Fargate che su istanze EC2 . • Gli Application Load Balancer consentono di usare la mappatura dinamica delle porte dell'host (in modo che per ogni istanza di container siano consentiti più processi provenienti dallo stesso servizio). • Gli Application Load Balancer supportano il routing basato su percorsi e le regole di priorità (in modo che più servizi possano utilizzare la stessa porta listener su un singolo Application Load Balancer).	
Network Load Balancer	Indirizza il traffico TCP o UDP (o livello 4).	

Tipo di Load Balancer	Utilizzare in questi casi	
Gateway Load Balancer	Indirizza il traffico TCP o UDP (o livello 4). Utilizza dispositivi virtuali, come firewall, sistemi di rilevamento e prevenzione delle intrusioni e sistemi di ispezione approfondita dei pacchetti.	

Ti consigliamo di utilizzare Application Load Balancers per i tuoi servizi Amazon ECS in modo da poter sfruttare queste ultime funzionalità, a meno che il tuo servizio non richieda una funzionalità disponibile solo con Network Load Balancer o Gateway Load Balancer. Per ulteriori informazioni sull'Elastic Load Balancing e le differenze tra questi tipi di load balancer, consulta la [Guida per l'utente di Elastic Load Balancing](#).

Con il load balancer paghi solo in base all'uso effettivo. Per ulteriori informazioni, consulta [Prezzi di Elastic Load Balancing](#).

Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico per Amazon ECS

I sistemi di bilanciamento del carico indirizzano le richieste solo verso gli obiettivi integri nelle zone di disponibilità del sistema di bilanciamento del carico. Ogni destinazione è registrata in un gruppo target. Il load balancer verifica lo stato di salute di ciascun target, utilizzando le impostazioni di controllo dello stato del gruppo target. Dopo aver registrato il bersaglio, deve superare un controllo dello stato di salute per essere considerato integro. Amazon ECS monitora il sistema di bilanciamento del carico. Il load balancer invia periodicamente controlli di integrità al container Amazon ECS. L'agente Amazon ECS monitora e attende che il sistema di bilanciamento del carico riferisca sullo stato del contenitore. Lo fa prima di considerare il container in buono stato.

Due parametri di controllo dello stato di Elastic Load Balancing influiscono sulla velocità di implementazione:

- Intervallo di controllo dello stato di salute: determina il periodo di tempo approssimativo, in secondi, tra i controlli sanitari di un singolo contenitore. Per impostazione predefinita, il sistema di bilanciamento del carico effettua i controlli ogni 30 secondi.

Questo parametro è denominato:

- `HealthCheckIntervalSeconds` nell'API Elastic Load Balancing
- Interval sulla console Amazon EC2
- Numero di soglie di integrità: determina il numero di controlli sanitari consecutivi necessari prima di considerare integro un contenitore non integro. Per impostazione predefinita, il load balancer richiede il superamento di cinque controlli di integrità prima di segnalare che il contenitore di destinazione è integro.

Questo parametro è denominato:

- `HealthyThresholdCount` nell'API Elastic Load Balancing
- Soglia integra sulla EC2 console Amazon

Con le impostazioni predefinite, il tempo totale per determinare lo stato di salute di un contenitore è di due minuti e 30 secondi ($30 \text{ seconds} * 5 = 150 \text{ seconds}$).

Puoi accelerare il processo di controllo dello stato se il servizio si avvia e si stabilizza in meno di 10 secondi. Per accelerare il processo, riduci il numero di controlli sanitari e l'intervallo tra i controlli.

- `HealthCheckIntervalSeconds`(nome API Elastic Load Balancing) o `Interval` (nome EC2 console Amazon): 5
- `HealthyThresholdCount`(nome API Elastic Load Balancing) o `Healthy threshold` (nome EC2 console Amazon): 2

Con questa impostazione, il processo di controllo dello stato richiede 10 secondi rispetto all'impostazione predefinita di due minuti e 30 secondi.

Per ulteriori informazioni sui parametri del controllo dello stato di Elastic Load Balancing, consulta [Controlli dello stato per i gruppi target](#) nella Elastic Load Balancing User Guide.

Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS

Per consentire l'ottimizzazione, i client mantengono una connessione keep alive al servizio container. Ciò consente alle richieste successive di quel client di riutilizzare la connessione esistente. Quando si desidera interrompere il traffico verso un container, si invia una notifica al sistema di bilanciamento del carico. Il load balancer controlla periodicamente se il client ha chiuso la connessione keep alive. L'agente Amazon ECS monitora il sistema di bilanciamento del carico e attende che il sistema di bilanciamento del carico segnali che la connessione keep alive è chiusa (la destinazione è in uno stato). UNUSED

La quantità di tempo che il load balancer attende per spostare la destinazione verso lo stato corrisponde al ritardo di annullamento della registrazione. UNUSED È possibile configurare il seguente parametro di bilanciamento del carico per velocizzare le distribuzioni.

- `deregistration_delay.timeout_seconds`: 300 (impostazione predefinita)

Se disponi di un servizio con un tempo di risposta inferiore a 1 secondo, imposta il parametro sul valore seguente per fare in modo che il load balancer attenda solo 5 secondi prima di interrompere la connessione tra il client e il servizio di back-end:

- `deregistration_delay.timeout_seconds`: 5

Note

Non impostate il valore su 5 secondi quando disponete di un servizio con richieste di lunga durata, come caricamenti lenti di file o connessioni di streaming.

Reattività SIGTERM

Amazon ECS invia innanzitutto un segnale SIGTERM all'attività per notificare che l'applicazione deve essere completata e chiusa. Quindi, Amazon ECS invia un messaggio SIGKILL. Quando le applicazioni ignorano il SIGTERM, il servizio Amazon ECS deve attendere l'invio del segnale SIGKILL per terminare il processo.

La quantità di tempo che Amazon ECS attende per inviare il messaggio SIGKILL è determinata dalla seguente opzione dell'agente Amazon ECS:

- ECS_CONTAINER_STOP_TIMEOUT: 30 (impostazione predefinita)

Per ulteriori informazioni sul parametro Container Agent, consulta [Amazon ECS Container Agent on GitHub](#).

Per accelerare il periodo di attesa, imposta il parametro dell'agente Amazon ECS sul seguente valore:

- ECS_CONTAINER_STOP_TIMEOUT 2:

Se l'applicazione impiega più di 1 secondo, moltiplica il valore per 2 e usa quel numero come valore.

In questo caso, Amazon ECS attende 2 secondi che il contenitore si spenga, quindi Amazon ECS invia un messaggio SIGKILL quando l'applicazione non si ferma.

Puoi anche modificare il codice dell'applicazione per intercettare il segnale SIGTERM e reagire ad esso. Di seguito è riportato un esempio in JavaScript:

```
process.on('SIGTERM', function() {  
  server.close();  
})
```

Questo codice fa sì che il server HTTP smetta di ascoltare tutte le nuove richieste, finisca di rispondere a tutte le richieste in corso, quindi il processo Node.js termina perché il ciclo di eventi non ha nulla a che fare. Pertanto, se il processo impiega solo 500 ms per completare le richieste in corso, termina in anticipo senza dover attendere il timeout di arresto e ricevere un SIGKILL.

Usa un Application Load Balancer per Amazon ECS

Un Application Load Balancer seleziona il routing a livello di applicazione (HTTP/HTTPS), supporta il routing in base al percorso e può instradare le richieste a una o più porte su ogni istanza di container nel cluster. Gli Application Load Balancer supportano la mappatura dinamica delle porte dell'host. Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). All'avvio dell'attività, il contenitore NGINX viene registrato con Application Load Balancer come combinazione di ID di istanza e porta e il traffico viene distribuito all'ID dell'istanza e alla porta corrispondenti a quel contenitore. Con la mappatura dinamica

possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta la [Guida per l'utente per Application Load Balancer](#).

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le distribuzioni, consulta:

- [Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico per Amazon ECS](#)
- [Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS](#)

Considera quanto segue quando utilizzi Application Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Il gruppo target deve avere il tipo di indirizzo IP impostato su. IPv4

Sebbene le attività siano supportate IPv6, l'integrazione dei servizi con IPv6 Elastic Load Balancing non è supportata.

- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'interfaccia di rete elastica, non a un'EC2 istanza Amazon.
- Se il tuo servizio richiede l'accesso a più porte con carico bilanciato, come la porta 80 e la porta 443 per un servizio HTTP/HTTPS, puoi configurare due listener. Un listener è responsabile di HTTPS che inoltra la richiesta al servizio e un altro listener responsabile del reindirizzamento delle richieste HTTP alla porta HTTPS appropriata. Per ulteriori informazioni, consulta [Creazione di un listener per Application Load Balancer](#) nella Guida per l'utente per Application Load Balancer.
- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. È possibile utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del bilanciamento del carico solo per il controller di distribuzione ECS mobile, non blu/verde o esterno. AWS CloudFormation AWS CodeDeploy Quando aggiungi, aggiorni o

rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento [UpdateService](#) all'API di Amazon Elastic Container Service.

- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta [Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS](#).
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- Utilizza un gruppo target unico per ogni servizio.

L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Application Load Balancer, consulta [Creare un Application Load Balancer in Application Load Balancers](#)

Usa un Network Load Balancer per Amazon ECS

Un Network Load Balancer seleziona il routing a livello di trasporto (TCP/SSL). È in grado di gestire milioni di richieste al secondo. Dopo aver ricevuto una connessione, il load balancer seleziona una destinazione dal gruppo di destinazione per la regola predefinita, utilizzando un algoritmo di routing per l'hash del flusso. Tenta quindi di aprire una connessione TCP per la destinazione selezionata sulla porta specificata nella configurazione del listener. Inoltre la richiesta senza modificare le intestazioni. I Network Load Balancer supportano la mappatura dinamica delle porte dell'host. Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). Una volta avviata l'attività, il container NGINX viene registrato con il Network Load Balancer come combinazione di ID istanza e porta, mentre il traffico viene instradato all'ID istanza e alla porta corrispondenti a tale container. Con la mappatura dinamica possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta la [Guida per l'utente dei Network Load Balancer](#).

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le implementazioni, consulta:

- [Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico per Amazon ECS](#)
- [Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS](#)

Considera quanto segue quando utilizzi Network Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Non è possibile associare più di cinque gruppi target a un servizio.
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'interfaccia di rete elastica, non a un'EC2 istanza Amazon.
- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. Puoi utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del load balancer solo per il controller di distribuzione ECS mobile, non AWS CodeDeploy blu/verde o esterno. AWS CloudFormation Quando aggiungi, aggiorni o rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento [UpdateService](#) all'API di Amazon Elastic Container Service.
- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.

- Quando si utilizza un Gateway Load Balancer configurato con indirizzi IP come destinazioni e Client IP Preservation disattivata, le richieste vengono viste come provenienti dall'indirizzo IP privato del Gateway Load Balancer. Ciò significa che i servizi alla base di un Gateway Load Balancer sono effettivamente aperti al mondo non appena si autorizzano le richieste in entrata e i controlli di integrità nel gruppo di sicurezza target.
- Per le attività di Fargate, è necessario utilizzare la versione della piattaforma 1.4.0 (Linux) o 1.0.0 (Windows).
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta [Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS](#).
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- La conservazione dell'indirizzo IP del client Network Load Balancer è compatibile con i target Fargate.
- Utilizzate un gruppo target unico per ogni servizio.

L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Network Load Balancer, consulta [Creare un Network Load Balancer in Network Load](#) Balancer.

Important

Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, necessaria per il tipo di avvio Fargate, è necessario scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'interfaccia di rete elastica, non a un' EC2 istanza Amazon.

Non è possibile registrare istanze per ID di istanza se hanno i seguenti tipi di istanza: C1,, CC1,, CC2, CG1 CG2, G1 CR1, G2,, M1 HI1, M2 HS1, M3 e T1. È possibile registrare le istanze di questi tipi in base all'indirizzo IP.

Usa un Gateway Load Balancer per Amazon ECS

Un Gateway Load Balancer funziona al terzo livello del modello Open Systems Interconnection (OSI), il livello di rete. È in ascolto per tutti i pacchetti IP attraverso tutte le porte e inoltra il traffico al gruppo di destinazione specificato nella regola del listener. Mantiene la persistenza dei flussi verso uno

specifico dispositivo di destinazione utilizzando 5 tuple (per i flussi). TCP/UDP flows) or 3-tuple (for non-TCP/UDP. Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). All'avvio dell'attività, il contenitore NGINX viene registrato con Gateway Load Balancer come combinazione di ID di istanza e porta e il traffico viene distribuito all'ID dell'istanza e alla porta corrispondenti a quel contenitore. Con la mappatura dinamica possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta [Cos'è un Gateway Load Balancer in Gateway Load Balancers](#).

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le implementazioni, consulta:

- [Ottimizza i parametri di controllo dello stato del sistema di bilanciamento del carico per Amazon ECS](#)
- [Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS](#)

Considera quanto segue quando utilizzi Gateway Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di rete `awsvpc` sono associate a un'interfaccia di rete elastica, non a un'istanza Amazon EC2.
- Per i servizi con attività che utilizzano una modalità di rete diversa da `awsvpc`, i Gateway Load Balancer non sono supportati.
- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. È possibile utilizzare AWS Copilot, AWS CLI o SDK per modificare la configurazione del load balancer solo per il controller di distribuzione ECS mobile, non blu/verde o esterno. AWS CloudFormation, AWS CodeDeploy. Quando aggiungi, aggiorni o

rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento [UpdateService](#) all'API di Amazon Elastic Container Service.

- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.
- Quando si utilizza un Gateway Load Balancer configurato con indirizzi IP come destinazioni, le richieste vengono viste come provenienti dall'indirizzo IP privato del Gateway Load Balancer. Ciò significa che i servizi alla base di un Gateway Load Balancer sono effettivamente aperti al mondo non appena si autorizzano le richieste in entrata e i controlli di integrità nel gruppo di sicurezza target.
- Per le attività di Fargate, è necessario utilizzare la versione della piattaforma 1.4.0 (Linux) o 1.0.0 (Windows).
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta [Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS](#).
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- Utilizza un gruppo target unico per ogni servizio.

L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Gateway Load Balancer, consulta [Guida introduttiva a Gateway Load Balancer in Gateway Load Balancers](#)

Important

Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, necessaria per il tipo di avvio Fargate, è necessario scegliere `ip` come tipo di destinazione e non `instance`. Questo perché le attività che utilizzano la modalità di `awsvpc` rete sono associate a un'interfaccia di rete elastica, non a un' EC2 istanza Amazon.

Non è possibile registrare istanze per ID di istanza se hanno i seguenti tipi di istanza: C1,, CC1,, CC2, CG1 CG2, G1 CR1, G2,, M1 HI1, M2 HS1, M3 e T1. È possibile registrare le istanze di questi tipi in base all'indirizzo IP.

Registrazione di più gruppi target con un servizio Amazon ECS

Il servizio Amazon ECS è in grado di servire traffico proveniente da più load balancer ed esporre più porte con carico bilanciato quando si specificano più gruppi di destinazione in una definizione del servizio.

Per creare un servizio che specifichi più gruppi target, devi creare il servizio utilizzando l'API Amazon ECS, l'SDK o un AWS CLI modello. AWS CloudFormation Dopo aver creato il servizio, è possibile visualizzare il servizio e i gruppi di destinazione registrati in esso con la AWS Management Console. Devi utilizzare [UpdateService](#) per modificare la configurazione del load balancer di un servizio esistente.

Più gruppi di destinazione possono essere specificati in una definizione del servizio utilizzando il formato seguente. Per la sintassi completa di una definizione di servizio, vedere [Modello di definizione del servizio](#).

```
"loadBalancers":[
  {
    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
    target_group_name_1/1234567890123456",
    "containerName":"container_name",
    "containerPort":container_port
  },
  {
    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
    target_group_name_2/6543210987654321",
    "containerName":"container_name",
    "containerPort":container_port
  }
]
```

Considerazioni

Considera quanto segue durante la specifica di più gruppi di destinazione in una definizione del servizio:

- Per i servizi che utilizzano Application Load Balancer o un Network Load Balancer (load balancer di rete), non è possibile collegare più di cinque gruppi di destinazione a un servizio.
- La specifica di più gruppi di destinazione in una definizione di servizio è supportata solo nelle seguenti condizioni:
 - Il servizio deve utilizzare un sistema Application Load Balancer o un Network Load Balancer.
 - Il servizio deve utilizzare il tipo di controller di implementazione (ECS) con aggiornamento in sequenza.
- La specificazione di più gruppi target è supportata per i servizi che contengono attività che utilizzano sia il tipo Fargate EC2 che quello di avvio.
- Durante la creazione di un servizio che specifica più gruppi di destinazione, è necessario creare il ruolo collegato ai servizi Amazon ECS. Il ruolo viene creato omettendo il parametro `role` nelle richieste API o la proprietà `Role` in AWS CloudFormation. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).

Definizioni del servizio di esempio

Di seguito sono riportati alcuni casi d'uso per la specifica di più gruppi di destinazione in una definizione del servizio. Per la sintassi completa di una definizione di servizio, vedere [Modello di definizione del servizio](#).

Disporre di bilanciatori di carico separati per il traffico interno ed esterno

Nel seguente caso d'uso, un servizio utilizza due load balancer separati, uno per il traffico interno e un secondo per il traffico della connessione Internet, per lo stesso container e porta.

```
"loadBalancers":[
  //Internal ELB
  {

    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
    "containerName":"nginx",
    "containerPort":8080
  },
```

```
//Internet-facing ELB
{

  "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
  target_group_name_2/6543210987654321",
    "containerName":"nginx",
    "containerPort":8080
  }
}
```

Esposizione di più porte dallo stesso contenitore

Nel seguente caso d'uso, un servizio utilizza un load balancer, ma espone più porte dallo stesso container. Ad esempio, un container Jenkins potrebbe esporre la porta 8080 per l'interfaccia Web Jenkins e la porta 50000 per l'API.

```
"loadBalancers":[
  {

    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
    target_group_name_1/1234567890123456",
      "containerName":"jenkins",
      "containerPort":8080
    },
    {

    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
    target_group_name_2/6543210987654321",
      "containerName":"jenkins",
      "containerPort":50000
    }
  ]
```

Esposizione delle porte di più container

Nel seguente caso d'uso, un servizio utilizza un load balancer e due gruppi di destinazione per esporre porte da container separati.

```
"loadBalancers":[
  {

    "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
    target_group_name_1/1234567890123456",
```

```
    "containerName": "webserver",
    "containerPort": 80
  },
  {
    "targetGroupArn": "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
    "containerName": "database",
    "containerPort": 3306
  }
]
```

Ridimensiona automaticamente il tuo servizio Amazon ECS

Il ridimensionamento automatico è la capacità di aumentare o diminuire automaticamente il numero desiderato di attività nel servizio Amazon ECS. Amazon ECS utilizza il servizio di Application Auto Scaling per fornire questa funzionalità. Per ulteriori informazioni, consulta la [Guida per l'utente di Application Auto Scaling](#).

Amazon ECS pubblica CloudWatch parametri con l'utilizzo medio di CPU e memoria del tuo servizio. Per ulteriori informazioni, consulta [Metriche di utilizzo del servizio Amazon ECS](#). Puoi utilizzare queste e altre CloudWatch metriche per scalare il tuo servizio (aggiungere altre attività) per far fronte a una domanda elevata nelle ore di punta e scalare il servizio (eseguire meno attività) per ridurre i costi nei periodi di basso utilizzo.

Amazon ECS Service Auto Scaling supporta i seguenti tipi di scalabilità automatica:

- [Usa una metrica target per scalare i servizi Amazon ECS](#): aumenta o diminuisce il numero di attività eseguite dal servizio in base al valore di destinazione per un parametro specifico. Questa operazione può essere paragonata al modo in cui il termostato regola la temperatura di una casa. Tu selezioni la temperatura, il termostato si occupa del resto.
- [Usa incrementi predefiniti basati sugli CloudWatch allarmi per scalare i servizi Amazon ECS](#): aumenta o diminuisce il numero di attività eseguite dal servizio in base a una serie di regolazioni di dimensionamento, chiamate regolazioni per fasi, che variano in base alla dimensione dell'utilizzo fuori limite segnalato dall'allarme.
- [Usa azioni pianificate per scalare i servizi Amazon ECS](#)—Aumentare o diminuire il numero di attività eseguite dal servizio in base alla data e all'ora.

- [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#)—Aumenta o riduci il numero di attività eseguite dal servizio in base all'analisi storica dei dati di carico per rilevare i modelli giornalieri o settimanali nei flussi di traffico.

Considerazioni

Quando usi le policy di dimensionamento, tieni in considerazione quanto segue:

- Amazon ECS invia i parametri a intervalli di 1 minuto a CloudWatch. Le metriche non sono disponibili finché i cluster e i servizi non le inviano a e non è possibile creare CloudWatch CloudWatch allarmi per metriche che non esistono.
- Le policy di dimensionamento supportano un periodo di attesa. Questo è il numero di secondi da attendere per rendere effettiva un'attività di dimensionamento precedente.
 - Per gli eventi di dimensionamento orizzontale, l'intenzione è di aumentare continuamente (ma non eccessivamente). Dopo che la scalabilità automatica dei servizi ha correttamente eseguito il dimensionamento orizzontale tramite una policy di dimensionamento, inizia a calcolare il periodo di attesa. La policy di dimensionamento non aumenterà di nuovo la capacità desiderata, a meno che non venga attivato un aumento orizzontale o che venga raggiunto il tempo di raffreddamento. Mentre è attivo il periodo di attesa di incremento, la capacità aggiunta dall'attività di incremento iniziale viene calcolata come parte della capacità desiderata per il successivo evento di incremento.
 - Per gli eventi di riduzione orizzontale, l'intenzione è di ridurre orizzontalmente in modo conservativo per proteggere la disponibilità dell'applicazione, in modo che le attività di riduzione siano bloccate fino alla scadenza del tempo di raffreddamento. Tuttavia, se un altro allarme attiva un'attività di aumento orizzontale durante il tempo di raffreddamento di riduzione orizzontale, Service Auto Scaling esegue immediatamente la riduzione orizzontale della destinazione. In questo caso, il periodo di attesa di riduzione si interrompe e non viene completato.
- Lo scheduler di servizi rispetta il numero desiderato in qualsiasi momento, ma fino a quando avrai policy di dimensionamento e allarmi su un servizio attivi, Service Auto Scaling potrebbe modificare un numero desiderato da te impostato manualmente.
- Se il numero desiderato di un servizio è impostato al di sotto del valore minimo di capacità e un allarme avvia un'attività di scalabilità orizzontale, Service Auto Scaling ridimensiona il conteggio desiderato fino al valore di capacità minimo e quindi continua a scalare in base alle esigenze, in base alla politica di scalabilità associata all'allarme. Tuttavia, un'attività di dimensionamento

in riduzione non modifica il numero desiderato, poiché esso è già inferiore al valore di capacità minimo.

- Se il numero desiderato di un servizio è impostato al di sopra del valore di capacità massima e un allarme avvia una scalabilità dell'attività, Service Auto Scaling ridimensiona il conteggio desiderato fino al valore di capacità massima e quindi continua a scalare secondo necessità, in base alla politica di scalabilità associata all'allarme. Tuttavia, un'attività di aumento non modifica il numero desiderato, poiché esso è già superiore al valore di capacità massimo.
- Durante le attività di dimensionamento, il numero reale delle attività in esecuzione in un servizio è il valore che Service Auto Scaling utilizza come punto di partenza, in contrapposizione al numero desiderato. Questo è ciò che dovrebbe essere la capacità di elaborazione. In questo modo si impedisce un dimensionamento eccessivo (fuori controllo) che non può essere soddisfatto, ad esempio, se non ci sono sufficienti risorse di istanze di container per inserire le attività aggiuntive. Se la capacità dell'istanza di container è disponibile in un secondo momento, l'attività di dimensionamento in attesa potrebbe andare a buon fine e, quindi, le attività di dimensionamento potrebbero continuare dopo il periodo di attesa.
- Se desideri che il numero di attività venga dimensionato a zero quando non è necessario eseguire alcuna operazione, imposta una capacità minima pari a 0. Con le policy di dimensionamento con monitoraggio degli obiettivi, quando la capacità effettiva è 0 e il parametro indica che esiste una domanda di carico di lavoro, Service Auto Scaling attende l'invio di un punto dati prima del dimensionamento orizzontale. In questo caso, viene dimensionato in base alla quantità minima possibile come punto di partenza e quindi riprende il dimensionamento in base al numero effettivo di processi in esecuzione.
- Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta [Scalabilità automatica e implementazioni dei servizi](#).
- Sono disponibili diverse opzioni di Application Auto Scaling per le attività di Amazon ECS. Il monitoraggio degli obiettivi è la modalità più semplice da utilizzare, in quanto è necessario soltanto impostare un valore target per un parametro, ad esempio l'utilizzo medio della CPU. L'autoscaler gestisce automaticamente il numero di attività necessarie per raggiungere tale valore. Il dimensionamento per fasi consente di reagire più rapidamente alle variazioni della domanda, poiché si definiscono le soglie specifiche per i parametri di dimensionamento e il numero di attività da aggiungere o rimuovere quando le soglie vengono superate. In particolare, consente di reagire molto rapidamente alle variazioni della domanda riducendo al minimo il tempo di superamento di una soglia di allarme.

Ottimizza la scalabilità automatica del servizio Amazon ECS

Un servizio Amazon ECS è una raccolta gestita di attività. Ogni servizio ha una definizione di attività associata, un numero di attività desiderato e una strategia di posizionamento opzionale. La scalabilità automatica del servizio Amazon ECS viene implementata tramite il servizio Application Auto Scaling. Application Auto Scaling utilizza le CloudWatch metriche come fonte per le metriche di scalabilità. Utilizza inoltre gli CloudWatch allarmi per impostare soglie su quando ampliare o disattivare il servizio. Puoi fornire le soglie per la scalabilità impostando un obiettivo metrico, denominato ridimensionamento del tracciamento degli obiettivi, o specificando delle soglie, denominate scalabilità dei passaggi. Una volta configurato, Application Auto Scaling calcola continuamente il numero di attività desiderato appropriato per il servizio. Inoltre, notifica ad Amazon ECS quando il numero di attività desiderato deve cambiare, ridimensionandolo orizzontalmente o ridimensionandolo.

Per utilizzare il service auto scaling in modo efficace, è necessario scegliere una metrica di scalabilità appropriata.

Un'applicazione deve essere scalata orizzontalmente se si prevede che la domanda sia superiore alla capacità attuale. Al contrario, un'applicazione può essere scalata per ridurre i costi quando le risorse superano la domanda.

Identifica una metrica

Per una scalabilità efficace, è fondamentale identificare una metrica che indichi l'utilizzo o la saturazione. Questa metrica deve presentare le seguenti proprietà per essere utile per la scalabilità.

- La metrica deve essere correlata alla domanda. Quando le risorse vengono mantenute stabili, ma la domanda cambia, anche il valore della metrica deve cambiare. La metrica dovrebbe aumentare o diminuire quando la domanda aumenta o diminuisce.
- Il valore della metrica deve essere scalato in proporzione alla capacità. Quando la domanda rimane costante, l'aggiunta di più risorse deve comportare una modifica proporzionale del valore della metrica. Pertanto, il raddoppio del numero di attività dovrebbe far diminuire la metrica del 50%.

Il modo migliore per identificare una metrica di utilizzo è eseguire test di carico in un ambiente di preproduzione come un ambiente di staging. Le soluzioni commerciali e open source per i test di carico sono ampiamente disponibili. Queste soluzioni in genere possono generare carichi sintetici o simulare il traffico utente reale.

Per avviare il processo di test di carico, crea dashboard per le metriche di utilizzo dell'applicazione. Queste metriche includono l'utilizzo della CPU, l'utilizzo della memoria, le operazioni di I/O, la

profondità della coda di I/O e il throughput di rete. Puoi raccogliere queste metriche con un servizio come Container Insights. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#). Durante questo processo, assicurati di raccogliere e tracciare le metriche relative ai tempi di risposta dell'applicazione o ai tassi di completamento del lavoro.

Inizia con una piccola richiesta o una percentuale di inserimento lavorativo ridotta. Mantieni questa frequenza costante per diversi minuti per consentire all'applicazione di riscaldarsi. Quindi, aumenta lentamente la velocità e mantienila costante per alcuni minuti. Ripeti questo ciclo, aumentando ogni volta la frequenza fino a quando i tempi di risposta o di completamento dell'applicazione non saranno troppo lenti per soddisfare gli obiettivi in termini di livello di servizio (). SLOs

Durante il test di carico, esaminate ciascuna metrica di utilizzo. Le metriche che aumentano insieme al carico sono le migliori candidate a fungere da migliori metriche di utilizzo.

Successivamente, identifica la risorsa che raggiunge la saturazione. Allo stesso tempo, esamina anche le metriche di utilizzo per vedere quale si appiattisce prima ad un livello elevato o raggiunge un picco e poi blocca prima l'applicazione. Ad esempio, se l'utilizzo della CPU aumenta dallo 0% al 70-80% man mano che si aggiunge carico e poi rimane a quel livello dopo aver aggiunto ancora più carico, si può affermare con certezza che la CPU è satura. A seconda dell'architettura della CPU, potrebbe non raggiungere mai il 100%. Ad esempio, supponiamo che l'utilizzo della memoria aumenti man mano che aggiungi carico e che l'applicazione si blocchi improvvisamente quando raggiunge il limite di memoria dell'attività o dell' EC2 istanza Amazon. In questa situazione, è probabile che la memoria sia stata completamente consumata. L'applicazione potrebbe consumare più risorse. Pertanto, scegli la metrica che rappresenta la risorsa che si esaurisce per prima.

Infine, riprova a testare il carico dopo aver raddoppiato il numero di attività o EC2 istanze Amazon. Supponiamo che la metrica chiave aumenti o diminuisca della metà rispetto a prima. In tal caso, la metrica è proporzionale alla capacità. Questa è una buona metrica di utilizzo per la scalabilità automatica.

Consideriamo ora questo scenario ipotetico. Supponiamo di testare un'applicazione e di scoprire che l'utilizzo della CPU alla fine raggiunge l'80% a 100 richieste al secondo. Quando viene aggiunto più carico, non aumenta più l'utilizzo della CPU. Tuttavia, fa sì che l'applicazione risponda più lentamente. Quindi, si esegue nuovamente il test di carico, raddoppiando il numero di attività ma mantenendo la frequenza al valore di picco precedente. Se ritieni che l'utilizzo medio della CPU scenda a circa il 40%, l'utilizzo medio della CPU è un buon candidato per una metrica di scalabilità. D'altra parte, se l'utilizzo della CPU rimane all'80% dopo l'aumento del numero di attività, l'utilizzo

medio della CPU non è una buona metrica di scalabilità. In tal caso, sono necessarie ulteriori ricerche per trovare una metrica adeguata.

Modelli applicativi e proprietà di scalabilità comuni

Vengono eseguiti software di tutti i tipi. AWS Molti carichi di lavoro sono creati internamente, mentre altri si basano su popolari software open source. Indipendentemente dalla loro origine, abbiamo osservato alcuni modelli di progettazione comuni per i servizi. La scalabilità efficace dipende in gran parte dal modello.

L'efficiente server legato alla CPU

L'efficiente server legato alla CPU non utilizza quasi nessuna risorsa oltre alla CPU e al throughput di rete. Ogni richiesta può essere gestita dalla sola applicazione. Le richieste non dipendono da altri servizi come i database. L'applicazione è in grado di gestire centinaia di migliaia di richieste simultanee e a tale scopo può utilizzarne più CPUs di una in modo efficiente. Ogni richiesta è gestita da un thread dedicato con un basso sovraccarico di memoria oppure esiste un ciclo di eventi asincrono che viene eseguito su ogni CPU che soddisfa le richieste. Ogni replica dell'applicazione è ugualmente in grado di gestire una richiesta. L'unica risorsa che potrebbe esaurirsi prima della CPU è la larghezza di banda della rete. Nei servizi legati alla CPU, l'utilizzo della memoria, anche al picco della velocità effettiva, è una frazione delle risorse disponibili.

Questo tipo di applicazione è adatto per il ridimensionamento automatico basato su CPU.

L'applicazione gode della massima flessibilità in termini di scalabilità. Può essere scalato verticalmente fornendo EC2 istanze Amazon più grandi o Fargate v. CPUs Inoltre, può essere ridimensionato orizzontalmente aggiungendo altre repliche. L'aggiunta di più repliche o il raddoppio delle dimensioni dell'istanza dimezzano l'utilizzo medio della CPU rispetto alla capacità.

Se utilizzi la EC2 capacità di Amazon per questa applicazione, valuta la possibilità di collocarla su istanze ottimizzate per il calcolo come la c5 famiglia o. c6g

L'efficiente server legato alla memoria

L'efficiente server legato alla memoria alloca una quantità significativa di memoria per richiesta. In caso di massima concorrenza, ma non necessariamente della velocità effettiva, la memoria si esaurisce prima dell'esaurimento delle risorse della CPU. La memoria associata a una richiesta viene liberata al termine della richiesta. Le richieste aggiuntive possono essere accettate purché sia disponibile memoria.

Questo tipo di applicazione è adatto per il ridimensionamento automatico basato sulla memoria.

L'applicazione gode della massima flessibilità in termini di scalabilità. Può essere scalato sia

verticalmente fornendogli risorse di memoria Amazon o EC2 Fargate più grandi. Inoltre, può essere ridimensionato orizzontalmente aggiungendo altre repliche. L'aggiunta di più repliche o il raddoppio delle dimensioni dell'istanza possono dimezzare l'utilizzo medio della memoria rispetto alla capacità.

Se utilizzi la EC2 capacità di Amazon per questa applicazione, valuta la possibilità di collocarla su istanze ottimizzate per la memoria come la `r5` famiglia o `r6g`.

Alcune applicazioni legate alla memoria non liberano la memoria associata a una richiesta al termine, quindi una riduzione della concorrenza non si traduce in una riduzione della memoria utilizzata. Per questo, non è consigliabile utilizzare il ridimensionamento basato sulla memoria.

Il server basato sul lavoratore

Il server basato sul lavoratore elabora una richiesta per ogni singolo thread di lavoro una dopo l'altra. I thread di lavoro possono essere thread leggeri, come thread POSIX. Possono anche essere thread più pesanti, come i processi UNIX. Indipendentemente dal thread, c'è sempre una concorrenza massima che l'applicazione è in grado di supportare. Di solito il limite di concorrenza è impostato proporzionalmente alle risorse di memoria disponibili. Se viene raggiunto il limite di concorrenza, le richieste aggiuntive vengono inserite in una coda di backlog. Se la coda del backlog supera i limiti, le richieste aggiuntive in arrivo vengono immediatamente respinte. Le applicazioni più comuni che si adattano a questo modello includono il server web Apache e Unicorn.

La concorrenza delle richieste è in genere la metrica migliore per scalare questa applicazione. Poiché esiste un limite di concorrenza per ogni replica, è importante eseguire la scalabilità orizzontale prima che venga raggiunto il limite medio.

Il modo migliore per ottenere le metriche relative alla concorrenza delle richieste consiste nel farle riportare dall'applicazione. CloudWatch Ogni replica dell'applicazione può pubblicare il numero di richieste simultanee come metrica personalizzata ad alta frequenza. Si consiglia di impostare la frequenza su almeno una volta al minuto. Dopo aver raccolto diversi report, puoi utilizzare la concorrenza media come metrica di scala. Questa metrica viene calcolata prendendo la concorrenza totale e dividendola per il numero di repliche. Ad esempio, se la concorrenza totale è 1000 e il numero di repliche è 10, la concorrenza media è 100.

Se la tua applicazione è alla base di un Application Load Balancer, puoi anche utilizzare la `ActiveConnectionCount` metrica per il load balancer come fattore nella metrica di scalabilità. La `ActiveConnectionCount` metrica deve essere divisa per il numero di repliche per ottenere un valore medio. Per la scalatura deve essere utilizzato il valore medio, anziché il valore di conteggio non elaborato.

Affinché questa progettazione funzioni al meglio, la deviazione standard della latenza di risposta deve essere piccola a basse frequenze di richiesta. Consigliamo che, durante i periodi di scarsa domanda, la maggior parte delle richieste riceva risposta in breve tempo e che non molte richieste impieghino molto più tempo della media per rispondere. Il tempo di risposta medio dovrebbe essere vicino al tempo di risposta del 95° percentile. In caso contrario, potrebbero verificarsi sovraccarichi dalla coda. Ciò porta a errori. Si consiglia di fornire repliche aggiuntive laddove necessario per mitigare il rischio di overflow.

Il server di attesa

Il server di attesa esegue alcune elaborazioni per ogni richiesta, ma il funzionamento dipende in larga misura da uno o più servizi downstream. Le applicazioni container spesso fanno un uso intensivo di servizi downstream come database e altri servizi API. La risposta di questi servizi può richiedere del tempo, in particolare in scenari ad alta capacità o ad alta concorrenza. Questo perché queste applicazioni tendono a utilizzare poche risorse della CPU e a sfruttare la massima concorrenza in termini di memoria disponibile.

Il servizio di attesa è adatto sia nel modello di server legato alla memoria che nel modello di server basato sui lavoratori, a seconda di come è progettata l'applicazione. Se la concorrenza dell'applicazione è limitata solo dalla memoria, l'utilizzo medio della memoria deve essere utilizzato come metrica di scalabilità. Se la concorrenza dell'applicazione si basa su un limite di lavoratori, è necessario utilizzare la concorrenza media come metrica di scalabilità.

Il server basato su Java

Se il server basato su Java è legato alla CPU e si adatta proporzionalmente alle risorse della CPU, potrebbe essere adatto all'efficiente modello di server associato alla CPU. In tal caso, l'utilizzo medio della CPU potrebbe essere appropriato come metrica di scalabilità. Tuttavia, molte applicazioni Java non sono legate alla CPU, il che le rende difficili da scalare.

Per prestazioni ottimali, si consiglia di allocare quanta più memoria possibile all'heap Java Virtual Machine (JVM). Le versioni recenti di JVM, tra cui Java 8 update 191 o versioni successive, impostano automaticamente la dimensione dell'heap il più grande possibile per adattarla al contenitore. Ciò significa che, in Java, l'utilizzo della memoria è raramente proporzionale all'utilizzo delle applicazioni. Con l'aumento della frequenza delle richieste e della concorrenza, l'utilizzo della memoria rimane costante. Per questo motivo, non è consigliabile scalare i server basati su Java in base all'utilizzo della memoria. Al contrario, in genere consigliamo di ridimensionare l'utilizzo della CPU.

In alcuni casi, i server basati su Java incontrano l'esaurimento dell'heap prima di esaurire la CPU. Se l'applicazione è soggetta all'esaurimento dell'heap in caso di elevata simultaneità, la metrica di scalabilità migliore è rappresentata dalle connessioni medie. Se la tua applicazione è soggetta all'esaurimento dell'heap a un throughput elevato, la frequenza media delle richieste è la metrica di scalabilità migliore.

Server che utilizzano altri runtime raccolti tramite garbage

Molte applicazioni server si basano su runtime che eseguono la raccolta dei dati indesiderati, ad esempio .NET e Ruby. Queste applicazioni server potrebbero rientrare in uno dei modelli descritti in precedenza. Tuttavia, come nel caso di Java, non è consigliabile scalare queste applicazioni in base alla memoria, poiché il loro utilizzo medio della memoria osservato spesso non è correlato alla velocità effettiva o alla concorrenza.

Per queste applicazioni, si consiglia di ridimensionare l'utilizzo della CPU se l'applicazione è vincolata alla CPU. Altrimenti, consigliamo di scalare in base al throughput medio o alla concorrenza media, in base ai risultati dei test di carico.

Job processor

Molti carichi di lavoro prevedono l'elaborazione asincrona dei processi. Includono applicazioni che non ricevono richieste in tempo reale, ma si iscrivono a una coda di lavoro per ricevere offerte di lavoro. Per questi tipi di applicazioni, la metrica di scalabilità corretta è quasi sempre la profondità della coda. L'aumento della coda indica che il lavoro in sospeso supera la capacità di elaborazione, mentre una coda vuota indica che c'è più capacità del lavoro da fare.

AWS i servizi di messaggistica, come Amazon SQS e Amazon Kinesis Data Streams, CloudWatch forniscono metriche che possono essere utilizzate per la scalabilità. Per Amazon SQS, `ApproximateNumberOfMessagesVisible` è la metrica migliore. Per Kinesis Data Streams, prendi in considerazione `MillisBehindLatest` utilizzo della metrica, pubblicata dalla Kinesis Client Library (KCL). È necessario calcolare la media di questa metrica tra tutti i consumatori prima di utilizzarla per la scalabilità.

Scalabilità automatica e implementazioni dei servizi

Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante un'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non vengano sospesi. Se desideri sospendere i processi di scalabilità orizzontale durante le implementazioni in corso, attieniti alla seguente procedura.

1. Chiama il [describe-scalable-targets](#) comando, specificando l'ID della risorsa del servizio associato alla destinazione scalabile in Application Auto Scaling (Esempio:). `service/default/sample-webapp` Registra l'output. Ne avrai bisogno quando chiamerai il comando successivo.
2. Richiamate il [register-scalable-target](#) comando, specificando l'ID della risorsa, lo spazio dei nomi e la dimensione scalabile. Specifica `true` sia per `DynamicScalingInSuspended` che per `DynamicScalingOutSuspended`.
3. Una volta completata la distribuzione, puoi chiamare il [register-scalable-target](#) comando per riprendere il ridimensionamento.

Per ulteriori informazioni, consulta [Sospensione e ripresa del dimensionamento per Application Auto Scaling](#).

Usa una metrica target per scalare i servizi Amazon ECS

Con le policy di dimensionamento con monitoraggio degli obiettivi, puoi scegliere un parametro e impostare un valore obiettivo. Amazon ECS Service Auto Scaling crea e gestisce gli allarmi che controllano CloudWatch la politica di scalabilità e calcola la regolazione della scalabilità in base alla metrica e al valore target. La policy di dimensionamento aggiunge o rimuove le attività di servizio in base alle necessità, per mantenere il parametro al valore di destinazione specificato o vicino a esso. Oltre a mantenere il parametro vicino al valore di destinazione, una policy di dimensionamento di monitoraggio obiettivi si adatta alle fluttuazioni del parametro dovute a un modello di carico fluttuante e riduce al minimo le fluttuazioni rapide del numero di attività in esecuzione all'interno del servizio.

Le politiche di tracciamento di Target eliminano la necessità di definire manualmente gli allarmi e le regolazioni di scalabilità. CloudWatch Amazon ECS lo gestisce automaticamente in base all'obiettivo impostato.

Tieni presenti le seguenti informazioni quando usi le policy di tracciamento delle destinazioni:

- Una policy di dimensionamento di monitoraggio obiettivi presuppone che essa debba eseguire un dimensionamento orizzontale quando il parametro specificato supera il valore di destinazione. Non puoi utilizzare una policy di dimensionamento di monitoraggio obiettivi per il dimensionamento orizzontale quando il parametro specificato è inferiore al valore di destinazione.
- Una policy di dimensionamento di monitoraggio obiettivi non esegue il dimensionamento quando il parametro specificato non dispone di dati sufficienti. Non esegue la scalabilità in quanto la carenza di dati non viene interpretata come basso utilizzo.

- Potrebbero esserci delle differenze tra il valore di destinazione e i punti di dati dei parametri reali. Ciò avviene perché Service Auto Scaling agisce sempre con prudenza, arrotondando per eccesso o per difetto quando determina la capacità da aggiungere o rimuovere. In questo modo si impedisce l'aggiunta di capacità insufficiente o la rimozione di capacità eccessiva.
- Per garantire la disponibilità delle applicazioni, il servizio aumenta in proporzione al parametro il più veloce possibile, ma si riduce in modo più graduale.
- Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta [Scalabilità automatica e implementazioni dei servizi](#).
- Puoi avere più policy di dimensionamento con monitoraggio degli obiettivi per un servizio Amazon ECS, purché ciascuna di esse utilizzi parametri differenti. Lo scopo di Service Auto Scaling è sempre quello di assegnare la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di monitoraggio degli obiettivi siano pronte o meno per l'aumento o la riduzione orizzontale. Il servizio viene aumentato se una qualsiasi delle policy di monitoraggio delle destinazioni è pronta per l'aumento, ma viene ridotto solo se tutte le policy di monitoraggio delle destinazioni (con la parte di riduzione abilitata) sono pronte per la riduzione.
- Non modificare o eliminare gli CloudWatch allarmi gestiti da Service Auto Scaling per una politica di scalabilità di tracciamento degli obiettivi. Service Auto Scaling elimina gli allarmi automaticamente quando elimini la policy di dimensionamento.
- Il parametro `ALBRequestCountPerTarget` per le policy di scalabilità del monitoraggio delle destinazioni non è supportato per il tipo di implementazione blu/verde.

Per ulteriori informazioni sulle policy di dimensionamento di tracciamento target, consulta la sezione relativa alle [policy di dimensionamento di tracciamento target](#) nella Guida per l'utente di Application Auto Scaling.

Crea una politica di scalabilità di tracciamento mirata per la scalabilità automatica del servizio Amazon ECS

Crea una politica di scalabilità di tracciamento mirata per consentire ad Amazon ECS di aumentare o diminuire automaticamente il numero di attività desiderato nel tuo servizio. Il monitoraggio del target funziona sulla base di un valore metrico target.

Console

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS](#).
2. Determina le metriche da utilizzare per la policy. Sono disponibili i seguenti parametri:
 - ECSServiceMedia CPUUtilization: l'utilizzo medio della CPU che il servizio dovrebbe utilizzare.
 - ECSServiceAverageMemoryUtilization— Utilizzo medio della memoria che il servizio dovrebbe utilizzare.
 - ALBRequestCountPerTarget— Il numero medio di richieste al minuto che l'attività dovrebbe idealmente ricevere.
3. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
4. Nella pagina Clusters (Cluster), scegli il cluster.
5. Nella pagina dei dettagli del cluster, nella sezione Servizi, quindi scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

6. Scegli Imposta il numero di attività.
7. Nella sezione Conteggio delle attività del servizio Amazon ECS, scegli Usa scalabilità automatica.

Viene visualizzata la sezione Conteggio delle attività.

- a. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. In Massimo, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Viene visualizzata la pagina delle politiche.

8. Scegli Crea politica di scalabilità.

Viene visualizzata la pagina Crea policy.

9. In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi.
10. In Policy name (Nome policy), immetti il nome della policy.

11. Per Tipo di metrica, scegli le metriche dall'elenco di opzioni.
12. Per l'utilizzo di Target, inserisci il valore target per la percentuale di attività che Amazon ECS deve mantenere. La scalabilità automatica del servizio consente di scalare la capacità fino a raggiungere l'utilizzo medio previsto o fino a raggiungere il numero massimo di attività specificato.
13. In Impostazioni aggiuntive, procedi come segue
 - a. Per il periodo di recupero scalabile, inserite la quantità di tempo in secondi dopo il completamento di un'attività scalabile prima che possa iniziare un'altra attività scalabile.
 - b. Per Periodo di recupero con scalabilità orizzontale, inserite il periodo di attesa, in secondi, per l'entrata in vigore di un'attività di scale-out precedente.
 - c. Per creare solo una politica di scalabilità orizzontale, seleziona Disabilita scale-in.
14. Scegli Crea politica di scalabilità.

AWS CLI

1. Registra il tuo servizio Amazon ECS come destinazione scalabile utilizzando il [register-scalable-target](#) comando.
2. Crea una politica di scalabilità utilizzando il comando. [put-scaling-policy](#)

Usa incrementi predefiniti basati sugli CloudWatch allarmi per scalare i servizi Amazon ECS

Con le politiche di scalabilità graduale, puoi creare e gestire gli CloudWatch allarmi che richiamano il processo di scalabilità. Quando viene violato un allarme, Amazon ECS avvia la politica di scalabilità associata a tale allarme. La politica di scalabilità graduale ridimensiona le attività utilizzando una serie di aggiustamenti, noti come aggiustamenti in fasi. La portata della regolazione varia in base alla dimensione dell'utilizzo fuori limite segnalato dall'allarme.

- Se la violazione supera la prima soglia, Amazon ECS applica la prima fase di adeguamento.
- Se la violazione supera la seconda soglia, Amazon ECS applica la seconda fase di adeguamento e così via.

Ti consigliamo vivamente di utilizzare le politiche di scalabilità di Target Tracking per adattare parametri come l'utilizzo medio della CPU o il numero medio di richieste per target. Le metriche che

diminuiscono all'aumentare della capacità e aumentano quando la capacità diminuisce possono essere utilizzate per ridimensionare proporzionalmente o in base al numero di attività utilizzando il monitoraggio del target. Questo aiuta a garantire che Amazon ECS segua da vicino la curva di domanda per le tue applicazioni.

Crea una politica di scalabilità graduale per la scalabilità automatica del servizio Amazon ECS

Crea una politica di scalabilità graduale per consentire ad Amazon ECS di aumentare o diminuire automaticamente il numero desiderato di attività nel tuo servizio. La scalabilità delle fasi viene eseguita sulla base di una serie di regolazioni di scalabilità, note come regolazioni delle fasi, che variano in base alla dimensione della violazione dell'allarme.

Console

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS](#).
2. Determina le metriche da utilizzare per la policy. Sono disponibili i seguenti parametri:
 - ECSServiceMedia CPUUtilization: l'utilizzo medio della CPU che il servizio dovrebbe utilizzare.
 - ECSServiceAverageMemoryUtilization— Utilizzo medio della memoria che il servizio dovrebbe utilizzare.
 - ALBRequestCountPerTarget— Il numero medio di richieste al minuto che l'attività dovrebbe idealmente ricevere.
3. Crea gli CloudWatch allarmi per le metriche. Per ulteriori informazioni, consulta [Creare un CloudWatch allarme basato su una soglia statica](#) nella Amazon CloudWatch User Guide.
4. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
5. Nella pagina Clusters (Cluster), scegli il cluster.
6. Nella pagina dei dettagli del cluster, nella sezione Servizi, quindi scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

7. Scegli Imposta il numero di attività.
8. Nella sezione Conteggio delle attività del servizio Amazon ECS, scegli Usa scalabilità automatica.

Viene visualizzata la sezione Conteggio delle attività.

- a. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. In Massimo, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Viene visualizzata la pagina delle politiche.

9. Scegli Crea politica di scalabilità.

Viene visualizzata la pagina Crea policy.

10. Per il tipo di policy Scaling, scegli Step Scaling.

11. Configura le proprietà di scalabilità orizzontale. In Procedure per aggiungere attività, procedi come segue:

- a. In Policy name (Nome policy), immetti il nome della policy.
- b. Per il nome della CloudWatch sveglia, scegli la CloudWatch sveglia.
- c. Per il tipo di aggregazione metrica, scegli come confrontare la metrica selezionata con la soglia definita.
- d. Per i tipi di regolazione A, scegli se l'adeguamento si basa su una modifica del numero di attività o su una modifica della percentuale di attività.
- e. In Azioni da intraprendere, inserisci i valori relativi all'azione da intraprendere.

Scegli Aggiungi passaggio per aggiungere azioni aggiuntive.

12. Configura le proprietà di scalabilità. In Procedure per rimuovere le attività, procedi come segue:

- a. In Policy name (Nome policy), immetti il nome della policy.
- b. Per il nome della CloudWatch sveglia, scegli la CloudWatch sveglia.
- c. Per il tipo di aggregazione metrica, scegli come confrontare la metrica selezionata con la soglia definita.
- d. Per i tipi di aggiustamento, scegli se l'adeguamento si basa su una modifica del numero di attività o su una modifica della percentuale di attività.
- e. In Azioni da intraprendere, inserisci i valori relativi all'azione da intraprendere.

Scegli Aggiungi passaggio per aggiungere azioni aggiuntive.

13. Per Periodo di recupero, inserisci la quantità di tempo, in secondi, per attendere che una precedente attività di ridimensionamento abbia effetto. Per una politica di aggiunta, si tratta del periodo successivo a un'attività di scalabilità orizzontale in cui la politica di scalabilità blocca le attività di scalabilità orizzontale e limita il numero di attività che possono essere scalate orizzontalmente alla volta. Per quanto riguarda una politica di rimozione, si tratta del periodo successivo a un'attività di scalabilità che deve trascorrere prima che possa iniziare un'altra attività scalabile.
14. Scegli Crea politica di scalabilità.

AWS CLI

1. Registra il tuo servizio Amazon ECS come destinazione scalabile utilizzando il [register-scalable-target](#) comando.
2. Crea una politica di scalabilità utilizzando il comando. [put-scaling-policy](#)

Usa azioni pianificate per scalare i servizi Amazon ECS

Con la scalabilità pianificata, puoi configurare la scalabilità automatica per la tua applicazione in base a variazioni di carico prevedibili creando azioni pianificate che aumentano o diminuiscono il numero di attività in momenti specifici. Ciò consente di dimensionare l'applicazione in modo proattivo per far fronte alle variazioni di carico prevedibili.

Queste azioni di dimensionamento pianificate consentono di ottimizzare costi e prestazioni. L'applicazione dispone di un numero sufficiente di attività per gestire il picco di traffico infrasettimanale, ma non supera il numero di attività in altri orari.

È possibile utilizzare simultaneamente il dimensionamento pianificato e le policy di dimensionamento per ottenere i vantaggi degli approcci proattivi e reattivi al dimensionamento. Dopo l'esecuzione di un'azione di scalabilità pianificata, la politica di scalabilità può continuare a decidere se aumentare ulteriormente il numero di attività. Ciò consente di garantire un numero sufficiente di attività per gestire il carico dell'applicazione. Sebbene l'applicazione sia scalabile in base alla domanda, la capacità attuale deve rientrare nel numero minimo e massimo di attività impostato dall'azione pianificata.

È possibile configurare il ridimensionamento della pianificazione utilizzando. AWS CLI Per ulteriori informazioni sulla scalabilità pianificata, vedere Scheduled [Scaling](#) nella Application Auto Scaling User Guide.

Crea un'azione pianificata per la scalabilità automatica del servizio Amazon ECS

Crea un'azione pianificata per far sì che Amazon ECS aumenti o riduca il numero di attività eseguite dal tuo servizio in base alla data e all'ora.

Console

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Scegli Service auto scaling.

Viene visualizzata la pagina Service Auto Scaling.

5. Se non hai configurato la scalabilità automatica del servizio, scegli Imposta il numero di attività.

Viene visualizzata la sezione relativa al conteggio delle attività del servizio Amazon ECS.

In Conteggio delle attività del servizio Amazon ECS, scegli Usa la scalabilità automatica del servizio per modificare il numero di attività desiderato per il servizio.

Viene visualizzata la sezione Conteggio delle attività.

- a. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. In Massimo, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Scegli Salva.

Viene visualizzata la pagina delle politiche.

6. Scegli Azioni pianificate, quindi scegli Crea.

Viene visualizzata la pagina Crea azione pianificata.

7. Per Nome azione, inserisci un nome univoco.
8. In Time zone (Fuso orario), scegli un fuso orario.

Tutti i fusi orari elencati provengono dal database del fuso orario IANA. Per ulteriori informazioni, vedere [Elenco dei fusi orari del database tz](#).

9. Per Ora di inizio, inserisci la data e l'ora di inizio dell'azione.

Se scegli una pianificazione periodica, l'ora di avvio definisce quando verrà eseguita la prima operazione pianificata della serie ricorrente.

10. In Compute (Calcolo), seleziona una delle opzioni disponibili.

- Per scalare in base a una pianificazione ricorrente, scegli la frequenza con cui Amazon ECS esegue l'azione pianificata.
- Se scegli un'opzione che inizia con Rate, l'espressione cron viene creata automaticamente.
- Se scegli Cron, inserisci un'espressione cron che specifichi quando eseguire l'operazione, in UTC.
- Per ridimensionare solo una volta, scegli Once.

11. In Regolazioni delle attività, procedi come segue:

- In Minimo, inserisci il numero minimo di attività che il servizio deve eseguire.
- In Massimo, inserisci il numero massimo di attività che il servizio deve eseguire.

12. Scegli Crea operazione pianificata.

CLI

Utilizza AWS CLI quanto segue per configurare le politiche di scalabilità pianificate per il tuo servizio. Sostituisci ogni *user input placeholder* con le tue informazioni.

Esempio: dimensionamento unico

Utilizzate il [put-scheduled-action](#) comando seguente con le --MaxCapacity opzioni --start-time "YYYY-MM-DDThh:mm:ssZ" and e --MinCapacity e o entrambe.

```
aws application-autoscaling put-scheduled-action --service-namespace ecs \  
  --resource-id service/my-cluster/my-service \  
  --scheduled-action-name my-one-time-schedule \  
  --start-time 2021-01-30T12:00:00 \  
  --scalable-target-action MinCapacity=3,MaxCapacity=10
```

Esempio: per pianificare il ridimensionamento in base a una pianificazione ricorrente

Utilizza il seguente comando [put-scheduled-action](#). Sostituiscili *user input* con i tuoi valori.

```
aws application-autoscaling put-scheduled-action --service-namespace ecs \  
  --resource-id service/my-cluster/my-service \  
  --scheduled-action-name my-recurring-action \  
  --schedule "rate(5 hours)" \  
  --start-time 2021-01-30T12:00:00 \  
  --end-time 2021-01-31T22:00:00 \  
  --scalable-target-action MinCapacity=3,MaxCapacity=10
```

Il programma di ricorrenza specificato viene eseguito in base al fuso orario UTC. Per specificare un fuso orario diverso, includi l'opzione `--time-zone` e il nome del fuso orario IANA, come nell'esempio seguente.

```
--time-zone "America/New_York"
```

Per ulteriori informazioni, vedere [Elenco dei fusi orari del database tz](#).

Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva

La scalabilità predittiva analizza i dati di caricamento passati provenienti dai flussi di traffico per analizzare i modelli giornalieri o settimanali. Utilizza quindi questa analisi per anticipare le esigenze future e aumentare in modo proattivo le attività del servizio in base alle esigenze.

La scalabilità automatica predittiva è particolarmente utile nelle seguenti situazioni.

- Traffico ciclico - Maggiore utilizzo delle risorse durante il normale orario lavorativo e riduzione dell'utilizzo delle risorse durante la sera e nei fine settimana.
- Schemi di on-and-off carico di lavoro ricorrenti - Gli esempi includono l'elaborazione in batch, i test o l'analisi periodica dei dati.
- Applicazioni con tempi di inizializzazione lunghi - Ciò può influire sulle prestazioni delle applicazioni durante gli eventi di scalabilità orizzontale causando una latenza notevole.

Se l'inizializzazione delle applicazioni impiega molto tempo e il traffico aumenta regolarmente, è consigliabile prendere in considerazione l'utilizzo del ridimensionamento predittivo. Ti aiuta a scalare più velocemente aumentando in modo proattivo il numero di attività per i carichi previsti, anziché utilizzare solo politiche di scalabilità dinamiche, come Target Tracking o Step Scaling. Aiutandovi a evitare la possibilità di sovraccaricare il numero di attività, la scalabilità predittiva può anche potenzialmente farvi risparmiare denaro.

Ad esempio, consideriamo un'applicazione che ha un utilizzo elevato durante l'orario di lavoro e uno ridotto durante la notte. All'inizio di ogni giornata lavorativa, la scalabilità predittiva consente di ridimensionare le attività prima del primo afflusso di traffico. Ciò permette all'applicazione di mantenere elevata disponibilità e prestazioni quando si passa da un periodo di utilizzo inferiore a un periodo di utilizzo più elevato. Non è necessario attendere che il dimensionamento dinamico reagisca alla variazione di traffico. Inoltre, non è necessario perdere tempo a esaminare i modelli di caricamento dell'applicazione e a cercare di pianificare la giusta quantità di attività utilizzando la scalabilità pianificata.

La scalabilità predittiva è una funzionalità a livello di servizio che ridimensiona l'attività del servizio indipendentemente dalla scalabilità della capacità di elaborazione sottostante (ad esempio, o Fargate). Per Fargate, AWS gestisce e ridimensiona automaticamente la capacità sottostante in base ai requisiti delle attività. Per quanto riguarda la EC2 capacità, puoi utilizzare i provider di capacità di gruppo Auto Scaling per ridimensionare automaticamente EC2 le istanze sottostanti in base ai requisiti di scalabilità delle tue attività.

Indice

- [Come funziona la scalabilità predittiva in Amazon ECS](#)
- [Crea una politica di scalabilità predittiva per la scalabilità automatica del servizio Amazon ECS](#)
- [Valuta le tue politiche di scalabilità predittiva per Amazon ECS](#)
- [Utilizza azioni pianificate per sovrascrivere i valori di previsione per Amazon ECS](#)
- [Policy di scalabilità predittiva avanzata che utilizza metriche personalizzate per Amazon ECS](#)

Come funziona la scalabilità predittiva in Amazon ECS

Qui puoi scoprire le considerazioni sull'utilizzo della scalabilità predittiva, come funziona e quali sono i limiti.

Considerazioni sull'utilizzo della scalabilità predittiva

- Vuoi assicurarti che la scalabilità predittiva sia adatta al tuo carico di lavoro. Puoi verificarlo configurando le politiche di scalabilità in modalità solo previsione e vedere cosa consiglia la console. È necessario valutare la previsione e i consigli prima di iniziare a utilizzare la scalabilità predittiva.
- Prima che la scalabilità predittiva possa iniziare a fare previsioni, sono necessarie almeno 24 ore di dati storici. Maggiore è il numero di dati storici disponibili, più efficace è la previsione, ideale per due settimane. Dovrai inoltre attendere 24 ore prima che la scalabilità predittiva possa generare

nuove previsioni quando elimini un servizio Amazon ECS e ne crei uno nuovo. Un modo per velocizzare questa operazione consiste nell'utilizzare parametri personalizzati per aggregare i parametri tra il vecchio e il nuovo servizio Amazon ECS.

- Scegli una metrica di carico che rappresenti accuratamente il carico completo della tua applicazione e sia l'aspetto dell'applicazione su cui è più importante scalare.
- La scalabilità dinamica con scalabilità predittiva ti aiuta a seguire da vicino la domanda della tua applicazione, in modo da poterla scalare in avanti durante le pause e in orizzontale durante gli aumenti imprevisti del traffico. Quando sono attive più politiche di scalabilità, ciascuna politica determina il numero desiderato di attività in modo indipendente e il numero di attività desiderato è impostato sul massimo di tali attività.
- È possibile utilizzare la scalabilità predittiva insieme alle politiche di scalabilità dinamica, come il monitoraggio degli obiettivi o la scalabilità dei passaggi, in modo che le applicazioni si scalino in base a modelli storici e in tempo reale. Di per sé, la scalabilità predittiva non consente la scalabilità delle attività.
- Se utilizzi un ruolo personalizzato quando chiami l'`register-scalable-target` API, potresti ricevere un errore che indica che la politica di scalabilità predittiva può funzionare solo con SLR abilitata. In questo caso dovresti chiamare di `register-scalable-target` nuovo ma senza `role-arn`. Usa SLR per registrare il target scalabile e chiama l'API `put-scaling-policy`.

Funzionamento del dimensionamento predittivo

Utilizzate la scalabilità predittiva creando una politica di scalabilità predittiva che specifica la metrica da monitorare e analizzare. CloudWatch La scalabilità predittiva deve disporre di almeno 24 ore di dati per iniziare a prevedere i valori futuri.

Dopo aver creato la policy, la scalabilità predittiva inizia ad analizzare i dati metrici relativi agli ultimi 14 giorni per identificare i modelli. Questa analisi viene utilizzata per generare previsioni orarie del fabbisogno per le prossime 48 ore. I CloudWatch dati più recenti vengono utilizzati per aggiornare la previsione ogni sei ore. Con l'arrivo di nuovi dati, la scalabilità predittiva migliora continuamente l'accuratezza delle previsioni future.

La prima volta che abiliti la scalabilità predittiva, questa viene eseguita in modalità di sola previsione. Genera previsioni in questa modalità, ma non ridimensiona il servizio Amazon ECS in base a tali previsioni. Ciò significa che puoi valutare l'accuratezza e l'idoneità della previsione. È possibile visualizzare i dati di previsione utilizzando l'operazione `GetPredictiveScalingForecast` API o il AWS Management Console.

Quando decidi di iniziare a utilizzare la scalabilità predittiva, passa la politica di scalabilità alla modalità previsione e scala. In questa modalità si verifica quanto segue.

Per impostazione predefinita, il tuo servizio Amazon ECS viene ridimensionato all'inizio di ogni ora in base alla previsione per quell'ora. Puoi scegliere di iniziare prima utilizzando la `SchedulingBufferTime` proprietà nell'operazione `PutScalingPolicy` API. Ciò consente l'avvio di nuove attività prima della domanda prevista e dà loro il tempo di avviarsi e prepararsi a gestire il traffico.

Limite massimo di attività

Quando registri i servizi Amazon ECS per la scalabilità, definisci un numero massimo di attività che possono essere avviate per servizio. Per impostazione predefinita, quando vengono impostate politiche di scalabilità, non è possibile aumentare il numero di attività oltre il limite massimo.

In alternativa, puoi consentire l'aumento automatico del numero massimo di attività del servizio se la previsione si avvicina o supera il numero massimo di attività del servizio Amazon ECS.

Warning

Fai attenzione quando consenti l'aumento automatico del numero massimo di attività. Ciò può comportare l'avvio di più attività del previsto, se l'aumento del numero massimo di attività non viene monitorato e gestito. L'aumento del numero massimo di attività diventa quindi il nuovo normale numero massimo di attività per il servizio Amazon ECS fino a quando non lo aggiorni manualmente. Il numero massimo di attività non torna automaticamente al massimo originale.

Regioni supportate

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Osaka)

- Asia Pacific (Seul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- China (Ningxia)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Stoccolma)
- Medio Oriente (Bahrein)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Crea una politica di scalabilità predittiva per la scalabilità automatica del servizio Amazon ECS

Crea una politica di scalabilità predittiva per far sì che Amazon ECS aumenti o riduca il numero di attività eseguite dal tuo servizio in base ai dati storici.

Note

Un nuovo servizio deve fornire almeno 24 ore di dati prima di poter generare una previsione.

Console

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta [Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS](#).
2. Determina le metriche da utilizzare per la policy. Sono disponibili i seguenti parametri:

- `ECSServiceMedia CPUUtilization`: l'utilizzo medio della CPU che il servizio dovrebbe utilizzare.
- `ECSServiceAverageMemoryUtilization`— Utilizzo medio della memoria che il servizio dovrebbe utilizzare.
- `ALBRequestCountPerTarget`— Il numero medio di richieste al minuto che l'attività dovrebbe idealmente ricevere.

In alternativa, puoi utilizzare una metrica personalizzata. È necessario definire i seguenti valori:

- Caricamento: una metrica che rappresenta con precisione il carico completo dell'applicazione ed è l'aspetto dell'applicazione su cui è più importante basare la scalabilità.
- Metrica di scalabilità: il miglior indicatore del grado di utilizzo è ideale per la tua applicazione.

3. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)

4. Nella pagina Clusters (Cluster), scegli il cluster.

5. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

6. Scegli Service auto scaling, quindi scegli Imposta il numero di attività.

7. Nella sezione Conteggio delle attività del servizio Amazon ECS, scegli Usa scalabilità automatica.

Viene visualizzata la sezione Conteggio delle attività.

- a. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. In Massimo, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Viene visualizzata la pagina delle politiche.

8. Scegli Crea politica di scalabilità.

Viene visualizzata la pagina Crea policy.

9. Per il tipo di policy Scaling, scegli Predictive Scaling.

10. In Policy name (Nome policy), immetti il nome della policy.
11. Per Metric pair, scegli le tue metriche dall'elenco di opzioni.

Se hai scelto Application Load Balancer request count per target (Conteggio delle richieste Application Load Balancer per destinazione), scegli un gruppo di destinazione in Target group (Gruppo di destinazione). Il conteggio delle richieste di Application Load Balancer per destinazione è supportato solo se è stato collegato un gruppo target di Application Load Balancer per il servizio.

Se hai scelto Coppia di metriche personalizzata, scegli le singole metriche dagli elenchi per Load metric e Scaling metric.

12. Per l'utilizzo di Target, inserisci il valore target per la percentuale di attività che Amazon ECS deve mantenere. La scalabilità automatica del servizio consente di scalare la capacità fino a raggiungere l'utilizzo medio previsto o fino a raggiungere il numero massimo di attività specificato.
13. Scegli Crea politica di scalabilità.

AWS CLI

Utilizza AWS CLI quanto segue per configurare le politiche di scalabilità predittiva per il tuo servizio Amazon ECS. Sostituisci ogni *user input placeholder* con le tue informazioni.

Per ulteriori informazioni sui CloudWatch parametri che puoi specificare, consulta [PredictiveScalingMetricSpecification](#) l'Amazon EC2 Auto Scaling API Reference.

Esempio 1: una politica di scalabilità predittiva con memoria predefinita.

Di seguito è riportato un esempio di policy con una configurazione di memoria predefinita.

```
cat policy.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 40,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ECSServiceMemoryUtilization"
      }
    }
  ],
  "SchedulingBufferTime": 3600,
```

```
"MaxCapacityBreachBehavior": "HonorMaxCapacity",
"Mode": "ForecastOnly"
}
```

L'esempio seguente illustra la creazione della politica eseguendo il [put-scaling-policy](#) comando con il file di configurazione specificato.

```
aws application-autoscaling put-scaling-policy \
--service-namespace ecs \
--region us-east-1 \
--policy-name predictive-scaling-policy-example \
--resource-id service/MyCluster/test \
--policy-type PredictiveScaling \
--scalable-dimension ecs:service:DesiredCount \
--predictive-scaling-policy-configuration file://policy.json
```

In caso di successo, questo comando restituisce l'ARN della politica.

```
{
  "PolicyARN": "arn:aws:autoscaling:us-
east-1:012345678912:scalingPolicy:d1d72dfe-5fd3-464f-83cf-824f16cb88b7:resource/ecs/
service/MyCluster/test:policyName/predictive-scaling-policy-example",
  "Alarms": []
}
```

Esempio 2: una politica di scalabilità predittiva con CPU predefinita.

Di seguito è riportato un esempio di policy con una configurazione CPU predefinita.

```
cat policy.json
{
  "MetricSpecifications": [
    {
      "TargetValue": 0.00000004,
      "PredefinedMetricPairSpecification": {
        "PredefinedMetricType": "ECSServiceCPUUtilization"
      }
    }
  ],
  "SchedulingBufferTime": 3600,
  "MaxCapacityBreachBehavior": "HonorMaxCapacity",
  "Mode": "ForecastOnly"
```

```
}
```

L'esempio seguente illustra la creazione della politica eseguendo il [put-scaling-policy](#) comando con il file di configurazione specificato.

```
aws aas put-scaling-policy \  
--service-namespace ecs \  
--region us-east-1 \  
--policy-name predictive-scaling-policy-example \  
--resource-id service/MyCluster/test \  
--policy-type PredictiveScaling \  
--scalable-dimension ecs:service:DesiredCount \  
--predictive-scaling-policy-configuration file://policy.json
```

In caso di successo, questo comando restituisce l'ARN della politica.

```
{  
  "PolicyARN": "arn:aws:autoscaling:us-  
east-1:012345678912:scalingPolicy:d1d72dfe-5fd3-464f-83cf-824f16cb88b7:resource/ecs/  
service/MyCluster/test:policyName/predictive-scaling-policy-example",  
  "Alarms": []  
}
```

Valuta le tue politiche di scalabilità predittiva per Amazon ECS

Prima di utilizzare una politica di scalabilità predittiva per scalare i tuoi servizi, esamina i consigli e gli altri dati per la tua policy nella console Amazon ECS. È un'opzione importante per assicurarsi che le previsioni siano accurate prima di applicare una policy di dimensionamento predittivo che dimensiona la capacità effettiva.

Se il servizio è nuovo, attendi 24 ore per creare la prima previsione.

Quando AWS crea una previsione, utilizza dati storici. Se il servizio non dispone ancora di molti dati storici recenti, il ridimensionamento predittivo potrebbe temporaneamente completare la previsione con aggregati creati dagli aggregati storici attualmente disponibili. Le previsioni vengono popolate per un massimo di due settimane prima della data di creazione di una policy.

Visualizzazione dei suggerimenti per il dimensionamento predittivo

Per un'analisi efficace, il service auto scaling dovrebbe avere almeno due policy di scalabilità predittiva da confrontare. Tuttavia, è ancora possibile esaminare i risultati per una singola policy. Quando crei più policy, puoi valutare una policy che utilizza un parametro rispetto a un parametro

che ne utilizza uno diverso. Puoi anche valutare l'impatto di diverse combinazioni di valori di destinazione e parametri. Dopo aver creato le politiche di scalabilità predittiva, Amazon ECS inizia immediatamente a valutare quale politica potrebbe fare un lavoro migliore per scalare il tuo gruppo.

Per visualizzare i tuoi consigli nella console Amazon ECS

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Scegli Service auto scaling.
5. Scegli la politica di scalabilità predittiva, quindi scegli Azioni, Scalabilità predittiva, Visualizza raccomandazione.

Puoi visualizzare i dettagli su una politica insieme alla nostra raccomandazione. Il suggerimento indica se l'utilizzo della policy di dimensionamento predittivo garantisce risultati migliori rispetto al non utilizzo.

Se non sei sicuro che una policy di dimensionamento predittivo sia appropriata per il tuo gruppo, consulta le colonne Impatto sulla disponibilità e Impatto sui costi per scegliere quella giusta. Le informazioni di ogni colonna indicano l'impatto della policy.

- Impatto sulla disponibilità: descrive se la politica eviterebbe un impatto negativo sulla disponibilità prevedendo attività sufficienti per gestire il carico di lavoro, rispetto al mancato utilizzo della policy.
- Impatto sui costi: descrive se la politica eviterebbe un impatto negativo sui costi evitando di sovradimensionare le attività rispetto al mancato utilizzo della politica. A causa di un eccesso di approvvigionamento, i servizi vengono sottoutilizzati o inattivi, il che non fa che aumentare l'impatto sui costi.

Se disponi di più policy, accanto al nome della policy che offre i maggiori vantaggi in termini di disponibilità a un costo inferiore viene visualizzato il tag Previsione migliore. Viene attribuito un peso maggiore all'impatto sulla disponibilità.

6. (Facoltativo) Per selezionare il periodo di tempo desiderato per i risultati delle raccomandazioni, scegliete il valore preferito dal menu a discesa Periodo di valutazione: 2 giorni, 1 settimana o 2 settimane. Per impostazione predefinita, il periodo di valutazione è rappresentato dalle

ultime due settimane. Un periodo di valutazione maggiore fornisce più punti dati per i risultati del suggerimento. Tuttavia, l'aggiunta di più punti dati potrebbe non migliorare i risultati se i modelli di carico sono cambiati, ad esempio dopo un periodo di domanda eccezionalmente elevata. In questo caso, puoi ottenere un suggerimento più mirato esaminando i dati più recenti.

Note

I suggerimenti vengono generati solo per le policy in modalità solo previsione. Questa funzione restituisce i risultati migliori quando una policy è in modalità solo previsione per tutto il periodo di valutazione. Se avvii una policy in modalità Previsione e dimensionamento e in un secondo momento la modifichi in una modalità solo previsione, è probabile che i risultati siano falsati. Questo accade perché la policy ha già contribuito alla capacità effettiva.

Analisi dei grafici di monitoraggio di dimensionamento predittivo

Nella console, puoi rivedere le previsioni dei giorni, delle settimane o dei mesi precedenti per visualizzare le prestazioni della politica nel tempo. È inoltre possibile utilizzare queste informazioni per valutare l'accuratezza delle previsioni al momento di decidere se consentire a una politica di ridimensionare il numero effettivo di attività.

Per esaminare i grafici di monitoraggio della scalabilità predittiva nella console Amazon ECS

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Scegli Service auto scaling.
5. Scegli la politica di scalabilità predittiva, quindi scegli Azioni, Scalabilità predittiva, Visualizza grafico.
6. Nella sezione Monitoraggio, puoi visualizzare le previsioni passate e future della policy in termini di carico e capacità rispetto ai valori effettivi. Il grafico Carico mostra la previsione di carico e i valori effettivi per il parametro di carico scelto. Il grafico della capacità mostra il numero di attività previste dalla policy. Include anche il numero effettivo di attività avviate. La linea verticale separa i valori storici dalle previsioni future. Questi grafici diventano disponibili poco dopo la creazione della policy.

7. (Facoltativo) Per modificare la quantità di dati cronologici mostrati nel grafico, scegli il valore desiderato dal menu a discesa Periodo di valutazione, nella parte superiore della pagina. Il periodo di valutazione non trasforma in alcun modo i dati di questa pagina, ma modifica soltanto la quantità di dati cronologici mostrati.

Confronto dei dati nel grafico Carico

Ogni riga orizzontale rappresenta un diverso insieme di punti dati riportati a intervalli di un'ora:

1. Carico effettivo osservato utilizza la statistica SUM per il parametro di carico scelto per mostrare il carico orario totale in passato.
2. Carico previsto dalla policy mostra la previsione del carico orario. Questa previsione si basa sulle due settimane precedenti di osservazioni del carico effettivo.

Confronto dei dati nel grafico Capacità

Ogni riga orizzontale rappresenta un diverso insieme di punti dati riportati a intervalli di un'ora:

1. Il numero effettivo di attività osservate mostra la capacità effettiva del servizio Amazon ECS in passato, che dipende dalle altre politiche di scalabilità e dalla dimensione minima del gruppo in vigore per il periodo di tempo selezionato.
2. Capacità prevista dalla policy mostra la capacità di base che si prevede di avere all'inizio di ogni ora, quando la policy è in modalità Previsione e dimensionamento.
3. Il numero di attività richiesto dedotto mostra il numero ideale di attività del tuo servizio per mantenere la metrica di scalabilità al valore target che hai scelto.
4. Il numero minimo di attività indica il numero minimo di attività del servizio.
5. La capacità massima indica il numero massimo di attività del servizio.

Ai fini del calcolo della capacità richiesta dedotta, partiamo dal presupposto che ogni attività venga utilizzata allo stesso modo in base a un valore obiettivo specificato. In pratica, il numero di attività non viene utilizzato allo stesso modo. Partendo dal presupposto che l'utilizzo sia distribuito uniformemente tra le attività, tuttavia, è possibile effettuare una stima verosimile della quantità di capacità necessaria. Il numero richiesto di attività viene quindi calcolato in modo inversamente proporzionale alla metrica di scalabilità utilizzata per la politica di scalabilità predittiva. In altre parole, all'aumentare del numero di attività, la metrica di scalabilità diminuisce alla stessa velocità. Ad esempio, se il numero di attività raddoppia, la metrica di scalabilità deve diminuire della metà.

La formula per la capacità richiesta differita è:

$$\text{sum of } (\text{actualServiceUnits} * \text{scalingMetricValue}) / (\text{targetUtilization})$$

Ad esempio, prendiamo `actualServiceUnits` (10) e `scalingMetricValue` (30) per una determinata ora. Prendiamo quindi il valore `targetUtilization` specificato nella policy di dimensionamento predittivo (60) e calcoliamo la capacità richiesta differita per la stessa ora. Viene restituito il valore 5. Ciò significa che cinque è la quantità di capacità richiesta necessaria per mantenere tale capacità in modo direttamente inversamente proporzionale al valore target del parametro di dimensionamento.

Note

Sono disponibili diversi fattori per regolare e migliorare i risparmi sui costi e la disponibilità dell'applicazione.

- Utilizza il dimensionamento predittivo per la capacità di base e il dimensionamento dinamico per gestire la capacità aggiuntiva. Il dimensionamento dinamico funziona indipendentemente dal dimensionamento predittivo, riducendo e aumentando orizzontalmente in base all'utilizzo corrente. Innanzitutto, Amazon ECS calcola il numero consigliato di attività per ogni politica di scalabilità non pianificata. Quindi, si ridimensiona in base alla politica che fornisce il maggior numero di attività.
- Per consentire la scalabilità integrata quando il carico diminuisce, il servizio deve sempre avere almeno una politica di scalabilità dinamica con la parte scalabile abilitata.
- Puoi migliorare le prestazioni di dimensionamento assicurandoti che la capacità minima e massima non siano troppo restrittive. A una policy con un numero consigliato di attività che non rientra nell'intervallo di capacità minima e massima verrà impedita la scalabilità verso l'alto e verso il basso.

Monitora i parametri di scalabilità predittiva per Amazon ECS con CloudWatch

Puoi usare Amazon CloudWatch per monitorare i tuoi dati per la scalabilità predittiva. Una politica di scalabilità predittiva raccoglie i dati che vengono utilizzati per prevedere il carico futuro. I dati raccolti vengono archiviati automaticamente CloudWatch a intervalli regolari e possono essere utilizzati per visualizzare le prestazioni della politica nel tempo. Puoi anche creare CloudWatch allarmi per avvisarti quando gli indicatori di performance cambiano oltre i limiti che hai definito.

Visualizzazione dei dati di previsione storici

I dati di previsione del carico per una politica di scalabilità predittiva possono essere visualizzati CloudWatch e possono essere utili quando si visualizzano le previsioni rispetto ad altre CloudWatch metriche in un unico grafico. Puoi anche vedere le tendenze nel tempo visualizzando un intervallo di tempo più ampio. Puoi accedere ai parametri cronologici fino a 15 mesi per avere una prospettiva migliore sulle performance di una policy.

Per visualizzare i dati di previsione storici utilizzando la console CloudWatch

1. Apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nel pannello di navigazione, scegli Metrics (Parametri), quindi scegli All metrics (Tutti i parametri).
3. Scegliete lo spazio dei nomi metrici Application Auto Scaling.
4. Scegliete Predictive Scaling Load Forecasts.
5. Nel campo di ricerca, inserisci il nome della politica di scalabilità predittiva o il nome del gruppo di servizi Amazon ECS, quindi premi Invio per filtrare i risultati.
6. Per creare il grafico di un parametro, seleziona la casella di controllo accanto al parametro. Per modificare il nome del grafico, seleziona l'icona a forma di matita. Per modificare l'intervallo di tempo, seleziona uno dei valori predefiniti o scegli custom (personalizzato). Per ulteriori informazioni, consulta la sezione [Rappresentazione grafica di una metrica](#) nella Amazon CloudWatch User Guide.
7. Per modificare la statistica, seleziona la scheda Graphed metrics (Parametri nel grafico). Scegli l'intestazione di colonna o un valore singolo, quindi scegli un'altra statistica. Sebbene sia possibile scegliere qualsiasi statistica per ogni metrica, non tutte le statistiche sono utili per le metriche. PredictiveScalingLoadForecast Ad esempio, le statistiche media, minimo e massimo sono utili, ma la statistica somma non lo è.
8. Per aggiungere un altro parametro al grafico, in Browse (Sfoglia), scegli All (Tutti), trova il parametro specifico, quindi seleziona la casella di controllo accanto a esso. Puoi aggiungere fino a 10 parametri.
9. (Facoltativo) Per aggiungere il grafico a una CloudWatch dashboard, scegli Azioni, Aggiungi alla dashboard.

Creazione di parametri di precisione utilizzando la matematica dei parametri

Con la matematica metrica, puoi interrogare più CloudWatch metriche e utilizzare espressioni matematiche per creare nuove serie temporali basate su queste metriche. Puoi visualizzare le serie temporali risultanti sulla CloudWatch console e aggiungerle ai dashboard. Per ulteriori informazioni sulla matematica dei parametri, consulta [Using metric Math nella](#) Amazon User Guide. CloudWatch

Utilizzando la matematica metrica, puoi rappresentare graficamente i dati generati da Service Auto Scaling per la scalabilità predittiva in diversi modi. Questo ti aiuta a monitorare le prestazioni delle policy nel tempo e ti aiuta a capire se la tua combinazione di parametri può essere migliorata.

Ad esempio, è possibile utilizzare un'espressione matematica dei parametri per monitorare il [errore percentuale assoluto medio](#) (MAPE). Il parametro MAPE aiuta a monitorare la differenza tra i valori previsti e i valori effettivi osservati durante una determinata finestra di previsione. Le modifiche nel valore di MAPE possono indicare se le prestazioni della policy stanno peggiorando nel tempo in quanto la natura della tua applicazione cambia. Un aumento del MAPE segnala un divario più ampio tra i valori previsti e i valori effettivi.

Esempio: espressione matematica dei parametri

Per iniziare a utilizzare questo tipo di grafico, puoi creare un'espressione matematica dei parametri come quella mostrata nell'esempio seguente.

Invece di un singolo parametro, è disponibile una serie di strutture di query di dati dei parametri per `MetricDataQueries`. Ogni articolo in `MetricDataQueries` ottiene un parametro o esegue un'espressione matematica. Il primo articolo, `e1`, è l'espressione matematica. L'espressione designata imposta il parametro `ReturnData` a `true`, che alla fine produce una singola serie temporale. Per tutte le altre parametri, il valore `ReturnData` è `false`.

Nell'esempio, l'espressione designata utilizza i valori effettivi e previsti come input e restituisce la nuova metrica (MAPE). `m1` è la CloudWatch metrica che contiene i valori di carico effettivi (supponendo che l'utilizzo della CPU sia la metrica di carico originariamente specificata per la policy denominata). `my-predictive-scaling-policy` `m2` è la CloudWatch metrica che contiene i valori di carico previsti. La sintassi matematica per il parametro MAPE è la seguente:

Media di (abs ((Effettivo - Previsto)/(Effettivo)))

Visualizza i parametri di precisione e imposta allarmi

Per visualizzare i dati delle metriche di precisione, seleziona la scheda Metriche nella console. CloudWatch È possibile rappresentare graficamente i dati da lì. Per ulteriori informazioni, consulta

[Aggiungere un'espressione matematica a un CloudWatch grafico](#) nella Amazon CloudWatch User Guide.

Puoi anche impostare un allarme su un parametro che stai monitorando dalla sezione Metrics (Parametri). Nella scheda Graphed metrics (Parametri nel grafico), seleziona l'icona Create alarm (Crea allarme) nella colonna Actions (Operazioni). L'icona Create alarm (Crea allarme) è rappresentata come una piccola campana. Per ulteriori informazioni e opzioni di notifica, consulta [Creazione di un CloudWatch allarme basato su un'espressione matematica metrica](#) e [Notifica agli utenti delle modifiche agli allarmi](#) nella Amazon CloudWatch User Guide.

In alternativa, puoi utilizzare [GetMetricData](#) ed [PutMetricAlarm](#) eseguire calcoli utilizzando la matematica metrica e creare allarmi in base all'output.

Utilizza azioni pianificate per sovrascrivere i valori di previsione per Amazon ECS

Talvolta, potrebbero essere disponibili ulteriori informazioni sui requisiti futuri dell'applicazione che il calcolo del forecast non è in grado di prendere in considerazione. Ad esempio, i calcoli previsionali potrebbero sottostimare le attività necessarie per un evento di marketing imminente. È possibile utilizzare le operazioni pianificate per sostituire temporaneamente il forecast nei periodi di tempo futuri. Le operazioni pianificate possono essere eseguite su base periodica o in una data e un'ora specifiche in cui si manifestino variazioni della domanda una tantum.

Ad esempio, è possibile creare un'azione pianificata con un numero di attività superiore a quello previsto. In fase di esecuzione, Amazon ECS aggiorna il numero minimo di attività del servizio. Poiché la scalabilità predittiva ottimizza il numero di attività, viene rispettata un'azione pianificata con un numero minimo di attività superiore ai valori previsti. In questo modo si evita che il numero di attività sia inferiore al previsto. Per non sovrascrivere più la previsione, utilizza una seconda azione pianificata per riportare il numero minimo di attività all'impostazione originale.

La procedura seguente descrive le fasi per sostituire il forecast nei periodi di tempo futuri.

Argomenti

- [Fase 1: analizza i dati di serie temporali \(opzionale\)](#)
- [Fase 2: creazione di due operazioni pianificate](#)

Important

Questo argomento presuppone che si stia cercando di sostituire la previsione per passare a una capacità superiore a quella prevista. Se è necessario ridurre temporaneamente il numero

di attività senza interferenze dovute a una politica di scalabilità predittiva, utilizza invece la modalità di sola previsione. In modalità di sola previsione, la scalabilità predittiva continuerà a generare previsioni, ma non aumenterà automaticamente il numero di attività. È quindi possibile monitorare l'utilizzo delle risorse e ridurre manualmente il numero di attività in base alle esigenze.

Fase 1: analizza i dati di serie temporali (opzionale)

Inizia analizzando i dati delle serie temporali dei forecast. Si tratta di un passaggio facoltativo, ma è utile se desideri comprendere i dettagli del forecast.

1. Recupero del forecast

Dopo aver creato il forecast, puoi avviare una query per un periodo di tempo specifico nel forecast. L'obiettivo della query è ottenere lo scenario completo dei dati delle serie temporali per un periodo di tempo specifico.

La query può includere fino a due giorni di dati di forecast futuro. Se utilizzi il dimensionamento predittivo per un certo periodo di tempo, puoi anche accedere ai dati dei forecast precedenti. Tuttavia, la durata massima tra l'ora di inizio e di fine è 30 giorni.

Per ottenere la previsione utilizzando il [get-predictive-scaling-forecast](#) AWS CLI comando, fornite i seguenti parametri nel comando:

- Immettete il nome del cluster nel `resource-id` parametro.
- Inserisci il nome della policy nel parametro `--policy-name`.
- Inserisci l'ora di inizio nel parametro `--start-time` affinché restituisca solo i dati di forecast per il periodo di tempo o dopo l'intervallo di tempo specificato.
- Inserisci l'ora di fine nel parametro `--end-time` affinché restituisca solo i dati di forecast per il periodo di tempo precedente all'intervallo di tempo specificato.

```
aws application-autoscaling get-predictive-scaling-forecast \
  --service-namespace ecs \
  --resource-id service/MyCluster/test \
  --policy-name cpu40-predictive-scaling-policy \
  --scalable-dimension ecs:service:DesiredCount \
  --start-time "2021-05-19T17:00:00Z" \
```

```
--end-time "2021-05-19T23:00:00Z"
```

Se riuscito, il comando restituirà dati simili a quelli dell'esempio seguente:

```
{
  "LoadForecast": [
    {
      "Timestamps": [
        "2021-05-19T17:00:00+00:00",
        "2021-05-19T18:00:00+00:00",
        "2021-05-19T19:00:00+00:00",
        "2021-05-19T20:00:00+00:00",
        "2021-05-19T21:00:00+00:00",
        "2021-05-19T22:00:00+00:00",
        "2021-05-19T23:00:00+00:00"
      ],
      "Values": [
        153.0655799339254,
        128.8288551285919,
        107.1179447150675,
        197.3601844551528,
        626.4039934516954,
        596.9441277518481,
        677.9675713779869
      ],
      "MetricSpecification": {
        "TargetValue": 40.0,
        "PredefinedMetricPairSpecification": {
          "PredefinedMetricType": "ASGCPUUtilization"
        }
      }
    }
  ],
  "CapacityForecast": {
    "Timestamps": [
      "2021-05-19T17:00:00+00:00",
      "2021-05-19T18:00:00+00:00",
      "2021-05-19T19:00:00+00:00",
      "2021-05-19T20:00:00+00:00",
      "2021-05-19T21:00:00+00:00",
      "2021-05-19T22:00:00+00:00",
      "2021-05-19T23:00:00+00:00"
    ]
  }
},
```

```

    "Values": [
      2.0,
      2.0,
      2.0,
      2.0,
      4.0,
      4.0,
      4.0
    ],
    "UpdateTime": "2021-05-19T01:52:50.118000+00:00"
  }

```

La risposta include due forecast: LoadForecast e CapacityForecast. LoadForecast mostra il forecast del carico orario. CapacityForecast mostra i valori di forecast per la capacità necessaria su base oraria per gestire il carico previsto pur mantenendo un TargetValue di 40,0 (40% utilizzo medio della CPU).

2. Identificazione del periodo di tempo di destinazione

Identifica l'ora o le ore in cui deve avvenire la variazione della domanda una tantum. Ricorda che le date e le ore mostrate nel forecast sono in UTC.

Fase 2: creazione di due operazioni pianificate

Ora crea quindi due operazioni pianificate per un periodo di tempo specifico in cui l'applicazione avrà un carico superiore a quello previsto. Ad esempio, se è previsto un evento di marketing che genererà traffico nel tuo sito per un periodo di tempo limitato, puoi pianificare un'operazione singola per aggiornare la capacità minima all'ora di inizio prevista. Quindi, pianifica un'altra operazione per riportare la capacità minima all'impostazione originale al termine dell'evento.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, quindi scegli il servizio.

Viene visualizzata la pagina dei dettagli del servizio.

4. Scegli Service Auto Scaling.

Viene visualizzata la pagina delle politiche.

5. Scegli Azioni pianificate, quindi scegli Crea.

Viene visualizzata la pagina di azione Crea pianificazione.

6. Per Nome azione, inserisci un nome univoco.
7. In Time zone (Fuso orario), scegli un fuso orario.

Tutti i fusi orari elencati provengono dal database del fuso orario IANA. Per ulteriori informazioni, vedere [Elenco dei fusi orari del database tz](#).

8. Per Ora di inizio, inserisci la data e l'ora di inizio dell'azione.
9. Per Recurrence (Ricorrenza), scegliere Once (Una tantum).
10. In Adeguamenti dell'attività, Per minimo, inserisci un valore inferiore o uguale al numero massimo di attività.
11. Scegli Crea operazione pianificata.

Viene visualizzata la pagina delle politiche.

12. Configura una seconda azione pianificata per riportare il numero minimo di attività all'impostazione originale alla fine dell'evento. La scalabilità predittiva può scalare il numero di attività solo quando il valore impostato per Minimo è inferiore ai valori di previsione.

Come creare due operazioni pianificate per eventi singoli (AWS CLI)

Per utilizzare il AWS CLI per creare le azioni pianificate, usa il comando [put-scheduled-update-group-action](#).

Ad esempio, definiamo una pianificazione che mantenga una capacità minima di tre istanze il 19 maggio alle 17:00 per otto ore. I comandi seguenti mostrano come implementare questo scenario.

Il primo comando [put-scheduled-update-group-action](#) indica ad Amazon Auto EC2 Scaling di aggiornare la capacità minima del gruppo Auto Scaling specificato alle 17:00 UTC del 19 maggio 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-start \  
  --auto-scaling-group-name my-asg --start-time "2021-05-19T17:00:00Z" --minimum-  
  capacity 3
```

Il secondo comando indica ad Amazon EC2 Auto Scaling di impostare la capacità minima del gruppo su una all'1:00 UTC del 20 maggio 2021.

```
aws autoscaling put-scheduled-update-group-action --scheduled-action-name my-event-end \
  --auto-scaling-group-name my-asg --start-time "2021-05-20T01:00:00Z" --minimum-
capacity 1
```

Dopo aver aggiunto queste azioni pianificate al gruppo Auto Scaling, Amazon Auto EC2 Scaling esegue le seguenti operazioni:

- Alle 17:00 UTC del 19 maggio 2021, viene eseguita la prima operazione pianificata. Se il gruppo include meno di tre istanze, il gruppo si dimensiona su tre istanze. Durante questo periodo e per le prossime otto ore, Amazon EC2 Auto Scaling può continuare a scalare orizzontalmente se la capacità prevista è superiore alla capacità effettiva o se è in vigore una politica di scalabilità dinamica.
- All' 01:00 UTC del 20 maggio 2021, viene eseguita la seconda operazione pianificata. Questo restituisce la capacità minima all'impostazione originale alla fine dell'evento.

Dimensionamento in base a pianificazioni ricorrenti

Per sostituire il forecast per lo stesso periodo di tempo ogni settimana, crea due operazioni pianificate e fornisci la logica di data e ora utilizzando un'espressione cron.

Il formato dell'espressione cron è costituito da cinque campi separati da spazi: [Minute] [Hour] [Day_of_Month] [Month_of_Year] [Day_of_Week]. I campi possono contenere tutti i valori consentiti, inclusi i caratteri speciali.

Ad esempio, la seguente espressione cron campi esegue un'operazione ogni giorno alle 06:30. L'asterisco viene utilizzato come carattere jolly per abbinare tutti i valori di un campo.

```
30 6 * * 2
```

Consulta anche

Per ulteriori informazioni su come gestire le azioni pianificate, consulta. [Usa azioni pianificate per scalare i servizi Amazon ECS](#)

Policy di scalabilità predittiva avanzata che utilizza metriche personalizzate per Amazon ECS

È possibile utilizzare metriche predefinite o personalizzate in una politica di scalabilità predittiva. Le metriche personalizzate sono utili quando le metriche predefinite, come CPU, memoria, ecc.) non sono sufficienti per descrivere sufficientemente il carico dell'applicazione.

Quando si crea una politica di scalabilità predittiva con metriche personalizzate, è possibile specificare altre metriche fornite da CloudWatch AWS. In alternativa, puoi specificare metriche che definisci e pubblichi tu stesso. Puoi anche utilizzare la matematica metrica per aggregare e trasformare le metriche esistenti in una nuova serie temporale che AWS non viene tracciata automaticamente. Un esempio è la combinazione dei valori nei dati calcolando nuove somme o medie, denominate aggregazioni. I dati risultanti sono chiamati aggregato.

La sezione seguente contiene le best practice e alcuni esempi di come costruire la struttura JSON per la policy.

Prerequisiti

Per aggiungere parametri personalizzati alla tua policy di dimensionamento predittivo, devi disporre delle autorizzazioni `cloudwatch:GetMetricData`.

Per specificare le tue metriche anziché le metriche AWS fornite, devi prima pubblicarle su CloudWatch. Per ulteriori informazioni, consulta [Pubblicazione di metriche personalizzate](#) nella Amazon CloudWatch User Guide.

Quando pubblichi i tuoi parametri, assicurati di pubblicare i punti dati con una frequenza minima di cinque minuti. I punti dati vengono recuperati in CloudWatch base alla durata del periodo necessario. Ad esempio, la specifica della metrica di carico utilizza metriche orarie per misurare il carico sull'applicazione. CloudWatch utilizza i dati metrici pubblicati per fornire un unico valore di dati per ogni periodo di un'ora aggregando tutti i punti dati con timestamp che rientrano in ogni periodo di un'ora.

Best practice

Le seguenti best practice consentono di utilizzare i parametri personalizzati in modo più efficace:

- La metrica più utile per la specifica della metrica di carico è una metrica che rappresenta il carico su un gruppo di Auto Scaling nel suo complesso.
- La metrica più utile per la scalabilità delle specifiche della metrica è la metrica media del throughput o dell'utilizzo per attività.
- L'utilizzo di destinazione deve corrispondere al tipo di parametro di dimensionamento. Per una configurazione di policy che utilizza l'utilizzo della CPU, si tratta, ad esempio, di una percentuale target.
- Se questi suggerimenti non vengono seguiti, i valori futuri previsti della serie temporale probabilmente non saranno corretti. Per verificare che i dati siano corretti, è possibile

visualizzare i valori previsti nella console. In alternativa, dopo aver creato la politica di scalabilità predittiva, ispeziona LoadForecast gli oggetti restituiti da una chiamata all'API.

[GetPredictiveScalingForecast](#)

- Ti consigliamo vivamente di configurare la scalabilità predittiva in modalità solo previsione in modo da poter valutare la previsione prima che la scalabilità predittiva inizi a scalare attivamente.

Limitazioni

- Puoi eseguire query su punti dati con un massimo di 10 parametri in un'unica specifica del parametro.
- Ai fini di questo limite, un'espressione conta come un parametro.

Risoluzione dei problemi di una politica di scalabilità predittiva con metriche personalizzate

Se si verifica un problema durante l'utilizzo di parametri personalizzati, si consiglia di procedere come segue:

- Se riscontri un problema in una distribuzione blu/verde durante l'utilizzo di un'espressione di ricerca, assicurati di aver creato un'espressione di ricerca che cerchi una corrispondenza parziale e non una corrispondenza esatta. È inoltre necessario verificare che la query trovi solo i gruppi di Auto Scaling in esecuzione nell'applicazione specifica. Per ulteriori informazioni sulla sintassi delle espressioni di ricerca, consulta la sintassi delle [espressioni di CloudWatch ricerca](#) nella Amazon CloudWatch User Guide.
- Il [put-scaling-policy](#) comando convalida un'espressione quando crei la tua politica di scalabilità. Tuttavia, esiste la possibilità che questo comando non riesca a identificare la causa esatta degli errori rilevati. Per risolvere i problemi, risolvi gli errori che ricevi in risposta a una richiesta al [get-metric-data](#) comando. È inoltre possibile risolvere i problemi relativi all'espressione dalla console. CloudWatch
- Devi specificare `false` per `ReturnData` se `MetricDataQueries` specifica la funzione `SEARCH()` da sola senza una funzione matematica come `SUM()`. Questo perché le espressioni di ricerca possono restituire più serie temporali e una specifica del parametro basata su un'espressione può restituire solo una serie temporale.
- Tutti i parametri coinvolti in un'espressione di ricerca devono avere la stessa risoluzione.

Creazione di JSON per la scalabilità predittiva di metriche personalizzate con Amazon ECS

La sezione seguente contiene esempi su come configurare la scalabilità predittiva da cui interrogare i dati. CloudWatch Esistono due metodi diversi per configurare questa opzione e il metodo scelto influisce sul formato utilizzato per costruire il JSON per la policy di scalabilità predittiva. Quando si utilizza la formula per i parametri, il formato del JSON varia ulteriormente in base alla formula eseguita.

1. Per creare una policy che ottenga i dati direttamente da altre CloudWatch metriche fornite da AWS o da metriche su cui pubblici, consulta. CloudWatch [Esempio di politica di scalabilità predittiva con metriche di carico e scalabilità personalizzate utilizzando AWS CLI](#)

Esempio di politica di scalabilità predittiva con metriche di carico e scalabilità personalizzate utilizzando AWS CLI

Per creare una politica di scalabilità predittiva con metriche di carico e scalabilità personalizzate con AWS CLI, memorizza gli argomenti per in un file JSON denominato. `--predictive-scaling-configuration config.json`

Inizi ad aggiungere parametri personalizzati sostituendo i valori sostituibili nell'esempio seguente con quelli dei tuoi parametri e del tuo obiettivo di destinazione.

```
{
  "MetricSpecifications": [
    {
      "TargetValue": 50,
      "CustomizedScalingMetricSpecification": {
        "MetricDataQueries": [
          {
            "Id": "scaling_metric",
            "MetricStat": {
              "Metric": {
                "MetricName": "MyUtilizationMetric",
                "Namespace": "MyNameSpace",
                "Dimensions": [
                  {
                    "Name": "MyOptionalMetricDimensionName",
                    "Value": "MyOptionalMetricDimensionValue"
                  }
                ]
              }
            }
          ]
        }
      },
    },
  ],
}
```

```

        "Stat": "Average"
      }
    }
  ],
  },
  "CustomizedLoadMetricSpecification": {
    "MetricDataQueries": [
      {
        "Id": "load_metric",
        "MetricStat": {
          "Metric": {
            "MetricName": "MyLoadMetric",
            "Namespace": "MyNameSpace",
            "Dimensions": [
              {
                "Name": "MyOptionalMetricDimensionName",
                "Value": "MyOptionalMetricDimensionValue"
              }
            ]
          },
          "Stat": "Sum"
        }
      }
    ]
  }
}

```

Per ulteriori informazioni, consulta la [MetricDataQuery](#) pagina di riferimento dell'API Amazon EC2 Auto Scaling.

Note

Di seguito sono riportate alcune risorse aggiuntive che possono aiutarti a trovare nomi di metriche, namespace, dimensioni e statistiche per le metriche: CloudWatch

- Per informazioni sui parametri disponibili per AWS i servizi, consulta i [AWS servizi che pubblicano CloudWatch metriche](#) nella Amazon CloudWatch User Guide.
- [Per ottenere il nome esatto della metrica, lo spazio dei nomi e le dimensioni \(se applicabili\) di una CloudWatch metrica con, consulta list-metrics. AWS CLI](#)

Per creare questo criterio, esegui il [put-scaling-policy](#) comando utilizzando il file JSON come input, come illustrato nell'esempio seguente.

```
aws application-autoscaling put-scaling-policy --policy-name my-predictive-scaling-policy \  
  --auto-scaling-group-name my-asg --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json
```

In caso di esito positivo, questo comando restituisce l'Amazon Resource Name (ARN) della policy.

```
{  
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-predictive-scaling-policy",  
  "Alarms": []  
}
```

Utilizzare le espressioni matematiche del parametro

La sezione seguente fornisce informazioni sull'utilizzo della matematica metrica con le politiche di scalabilità predittiva nella politica.

Comprendere la matematica dei parametri

Se tutto ciò che vuoi fare è aggregare i dati metrici esistenti, la matematica metrica ti consente di risparmiare lo sforzo e il costo della pubblicazione di un'altra CloudWatch metrica su. CloudWatch Puoi utilizzare qualsiasi metrica AWS fornita e puoi anche utilizzare le metriche che definisci come parte delle tue applicazioni.

Per ulteriori informazioni, consulta [Using metric Math](#) nella Amazon CloudWatch User Guide.

Se scegli di utilizzare un'espressione matematica dei parametri nella policy di dimensionamento predittivo, considera i seguenti punti:

- Le operazioni di matematica dei parametri utilizzano i punti dati della combinazione univoca di nome del parametro, spazio dei nomi e coppie di parametri chiave/valore delle dimensioni.
- Puoi utilizzare qualsiasi operatore aritmetico (+ - */^), funzione statistica (come AVG o SUM) o altra funzione che supporti. CloudWatch
- È possibile utilizzare i parametri e i risultati di altre espressioni matematiche nelle formule dell'espressione matematica.

- Le espressioni matematiche dei parametri possono essere costituite da aggregazioni diverse. Tuttavia, per il risultato finale dell'aggregazione è una best practice utilizzare Average per il parametro di dimensionamento e Sum per il parametro del carico.
- Qualsiasi espressione utilizzata in una specifica dei parametri deve restituire una singola serie temporale.

Per utilizzare i parametri matematici, procedi come segue:

- Scegli una o più metriche. CloudWatch Quindi, crea l'espressione. Per ulteriori informazioni, consulta [Using metric Math](#) nella Amazon CloudWatch User Guide.
- Verifica che l'espressione matematica della metrica sia valida utilizzando la CloudWatch console o l'API. CloudWatch [GetMetricData](#)

Esempio di policy di dimensionamento predittivo che combina parametri utilizzando la formula dei parametri (AWS CLI)

A volte, invece di specificare direttamente il parametro, potrebbe essere necessario prima elaborarne i dati in qualche modo. Ad esempio, potresti avere un'applicazione che estrae il lavoro da una coda Amazon SQS e potresti utilizzare il numero di elementi nella coda come policy di dimensionamento predittivo. Il numero di messaggi nella coda non definisce esclusivamente il numero di istanze necessarie. Pertanto, è necessario più lavoro per creare un parametro che si può utilizzare per calcolare il backlog per istanza.

Di seguito viene illustrato un esempio di policy di dimensionamento predittivo per questo scenario. Specifica i parametri di dimensionamento e del carico basati sul parametro `ApproximateNumberOfMessagesVisible` di Amazon SQS, ovvero il numero di messaggi disponibili per il recupero dalla coda. Utilizza anche il parametro Amazon EC2 Auto Scaling e un'espressione matematica per calcolare il backlog per istanza per la `GroupInServiceInstances` metrica di scaling.

```
aws application-autoscaling put-scaling-policy --policy-name my-sqs-custom-metrics-policy \  
  --policy-type PredictiveScaling \  
  --predictive-scaling-configuration file://config.json \  
  --service-namespace ecs \  
  --resource-id service/MyCluster/test \  
  "MetricSpecifications": [  
    {
```



```

    "TargetValue": 100,
    "CustomizedScalingMetricSpecification": {
      "MetricDataQueries": [
        {
          "Label": "Get the queue size (the number of messages waiting to be
processed)",
          "Id": "queue_size",
          "MetricStat": {
            "Metric": {
              "MetricName": "ApproximateNumberOfMessagesVisible",
              "Namespace": "AWS/SQS",
              "Dimensions": [
                {
                  "Name": "QueueName",
                  "Value": "my-queue"
                }
              ]
            },
            "Stat": "Sum"
          },
          "ReturnData": false
        },
        {
          "Label": "Get the group size (the number of running instances)",
          "Id": "running_capacity",
          "MetricStat": {
            "Metric": {
              "MetricName": "GroupInServiceInstances",
              "Namespace": "AWS/AutoScaling",
              "Dimensions": [
                {
                  "Name": "AutoScalingGroupName",
                  "Value": "my-asg"
                }
              ]
            },
            "Stat": "Sum"
          },
          "ReturnData": false
        },
        {
          "Label": "Calculate the backlog per instance",
          "Id": "scaling_metric",
          "Expression": "queue_size / running_capacity",

```

```

        "ReturnData": true
      }
    ]
  },
  "CustomizedLoadMetricSpecification": {
    "MetricDataQueries": [
      {
        "Id": "load_metric",
        "MetricStat": {
          "Metric": {
            "MetricName": "ApproximateNumberOfMessagesVisible",
            "Namespace": "AWS/SQS",
            "Dimensions": [
              {
                "Name": "QueueName",
                "Value": "my-queue"
              }
            ],
          },
          "Stat": "Sum"
        },
        "ReturnData": true
      }
    ]
  }
}
]
}
}

```

L'esempio restituisce l'ARN della policy.

```

{
  "PolicyARN": "arn:aws:autoscaling:region:account-id:scalingPolicy:2f4f5048-d8a8-4d14-b13a-d1905620f345:autoScalingGroupName/my-asg:policyName/my-sqs-custom-metrics-policy",
  "Alarms": []
}

```

Interconnetti i servizi Amazon ECS

Le applicazioni eseguite nelle attività di Amazon ECS spesso devono ricevere connessioni da Internet o connettersi ad altre applicazioni eseguite nei servizi Amazon ECS. Se hai bisogno di connessioni esterne da Internet, consigliamo di utilizzare Elastic Load Balancing. Per ulteriori informazioni sul

bilanciamento del carico integrato, consulta [the section called “Usa il bilanciamento del carico per distribuire il traffico di servizio”](#).

Se hai bisogno di un'applicazione per connetterti ad altre applicazioni eseguite nei servizi Amazon ECS, Amazon ECS offre i modi seguenti per farlo senza un sistema di bilanciamento del carico:

- Amazon ECS Service Connect

Consigliamo Service Connect, che fornisce la configurazione di Amazon ECS per l'individuazione dei servizi, la connettività e il monitoraggio del traffico. Con Service Connect, le tue applicazioni possono utilizzare nomi brevi e porte standard per connettersi ai servizi Amazon ECS nello stesso cluster o VPCs in altri cluster, anche all'interno dello stesso. Regione AWS

Quando usi Service Connect, Amazon ECS gestisce tutte le parti del service discovery: crea i nomi che possono essere scoperti, gestisce dinamicamente le voci per ogni attività all'inizio e all'interruzione delle attività, esegue un agente in ogni attività configurato per scoprire i nomi. L'applicazione può cercare i nomi utilizzando la funzionalità standard per i nomi DNS e stabilendo connessioni. Se l'applicazione lo fa già, non è necessario modificarla per utilizzare Service Connect.

Fornisci la configurazione completa all'interno di ogni definizione di servizio e attività. Amazon ECS gestisce le modifiche a questa configurazione in ogni distribuzione del servizio, per garantire che tutte le attività di una distribuzione si comportino nello stesso modo. Ad esempio, un problema comune relativo al DNS as service discovery è il controllo di una migrazione. Se si modifica un nome DNS in modo che punti ai nuovi indirizzi IP sostitutivi, potrebbe essere necessario il tempo TTL massimo prima che tutti i client inizino a utilizzare il nuovo servizio. Con Service Connect, la distribuzione del client aggiorna la configurazione sostituendo le attività del client. È possibile configurare l'interruttore automatico di distribuzione e altre configurazioni di distribuzione per influire sulle modifiche di Service Connect allo stesso modo di qualsiasi altra distribuzione.

Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

- Individuazione dei servizi di Amazon ECS

Un altro approccio alla service-to-service comunicazione è la comunicazione diretta tramite Service Discovery. In questo approccio, puoi utilizzare l'integrazione del rilevamento dei AWS Cloud Map servizi con Amazon ECS. Utilizzando Service Discovery, Amazon ECS sincronizza l'elenco delle attività avviate con AWS Cloud Map, che mantiene un nome host DNS che si risolve negli indirizzi IP interni di una o più attività di quel particolare servizio. Altri servizi in Amazon VPC possono

utilizzare questo nome host DNS per inviare traffico direttamente a un altro contenitore utilizzando il suo indirizzo IP interno.

Questo approccio alla service-to-service comunicazione offre una bassa latenza. Non ci sono componenti aggiuntivi tra i contenitori. Il traffico viaggia direttamente da un container all'altro.

Questo approccio è adatto quando si utilizza la modalità `awsvpc` di rete, in cui ogni attività ha il proprio indirizzo IP univoco. La maggior parte dei software supporta solo l'uso di A record DNS, che si risolvono direttamente in indirizzi IP. Quando si utilizza la modalità di `awsvpc` rete, l'indirizzo IP per ogni operazione è un A record. Tuttavia, se utilizzi la modalità di `bridge` rete, è possibile che più contenitori condividano lo stesso indirizzo IP. Inoltre, le mappature dinamiche delle porte fanno sì che ai contenitori vengano assegnati in modo casuale i numeri di porta su quel singolo indirizzo IP. A questo punto, un A record non è più sufficiente per l'individuazione del servizio. È inoltre necessario utilizzare un SRV record. Questo tipo di record può tenere traccia sia degli indirizzi IP che dei numeri di porta, ma richiede la configurazione appropriata delle applicazioni. Alcune applicazioni predefinite utilizzate potrebbero non supportare SRV i record.

Un altro vantaggio della modalità di `awsvpc` rete è che si dispone di un gruppo di sicurezza unico per ogni servizio. È possibile configurare questo gruppo di sicurezza per consentire le connessioni in entrata solo dai servizi upstream specifici che devono comunicare con quel servizio.

Lo svantaggio principale della service-to-service comunicazione diretta tramite Service Discovery è che è necessario implementare una logica aggiuntiva per ripetere i tentativi e gestire gli errori di connessione. I record DNS hanno un periodo time-to-live (TTL) che controlla per quanto tempo vengono memorizzati nella cache. L'aggiornamento del record DNS e la scadenza della cache richiedono del tempo per consentire alle applicazioni di recuperare la versione più recente del record DNS. Pertanto, l'applicazione potrebbe finire per risolvere il record DNS in modo che punti a un altro contenitore che non è più presente. L'applicazione deve gestire i nuovi tentativi e disporre di una logica per ignorare i backend non validi.

Per ulteriori informazioni, consulta [Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS](#)

- Lattice Amazon VPC

Amazon VPC Lattice è un servizio di rete di applicazioni gestito che i clienti di Amazon ECS utilizzano per osservare, proteggere e monitorare le applicazioni create su servizi di AWS elaborazione e account senza dover modificare il codice. VPCs

VPC Lattice utilizza gruppi target, che sono una raccolta di risorse di calcolo. Questi obiettivi eseguono la tua applicazione o il tuo servizio e possono essere EC2 istanze Amazon, indirizzi IP, funzioni Lambda e Application Load Balancer. Associando i propri servizi Amazon ECS a un gruppo target VPC Lattice, i clienti possono ora abilitare le attività di Amazon ECS come destinazioni IP in VPC Lattice. Amazon ECS registra automaticamente le attività nel gruppo target VPC Lattice quando vengono lanciate le attività per il servizio registrato.

Per ulteriori informazioni, consulta [Usa Amazon VPC Lattice per connettere, osservare e proteggere i tuoi servizi Amazon ECS](#).

Tabella di compatibilità della modalità di rete

La tabella seguente illustra la compatibilità tra queste opzioni e le modalità di rete delle attività. Nella tabella, "client" si riferisce all'applicazione che effettua le connessioni dall'interno di un'attività Amazon ECS.

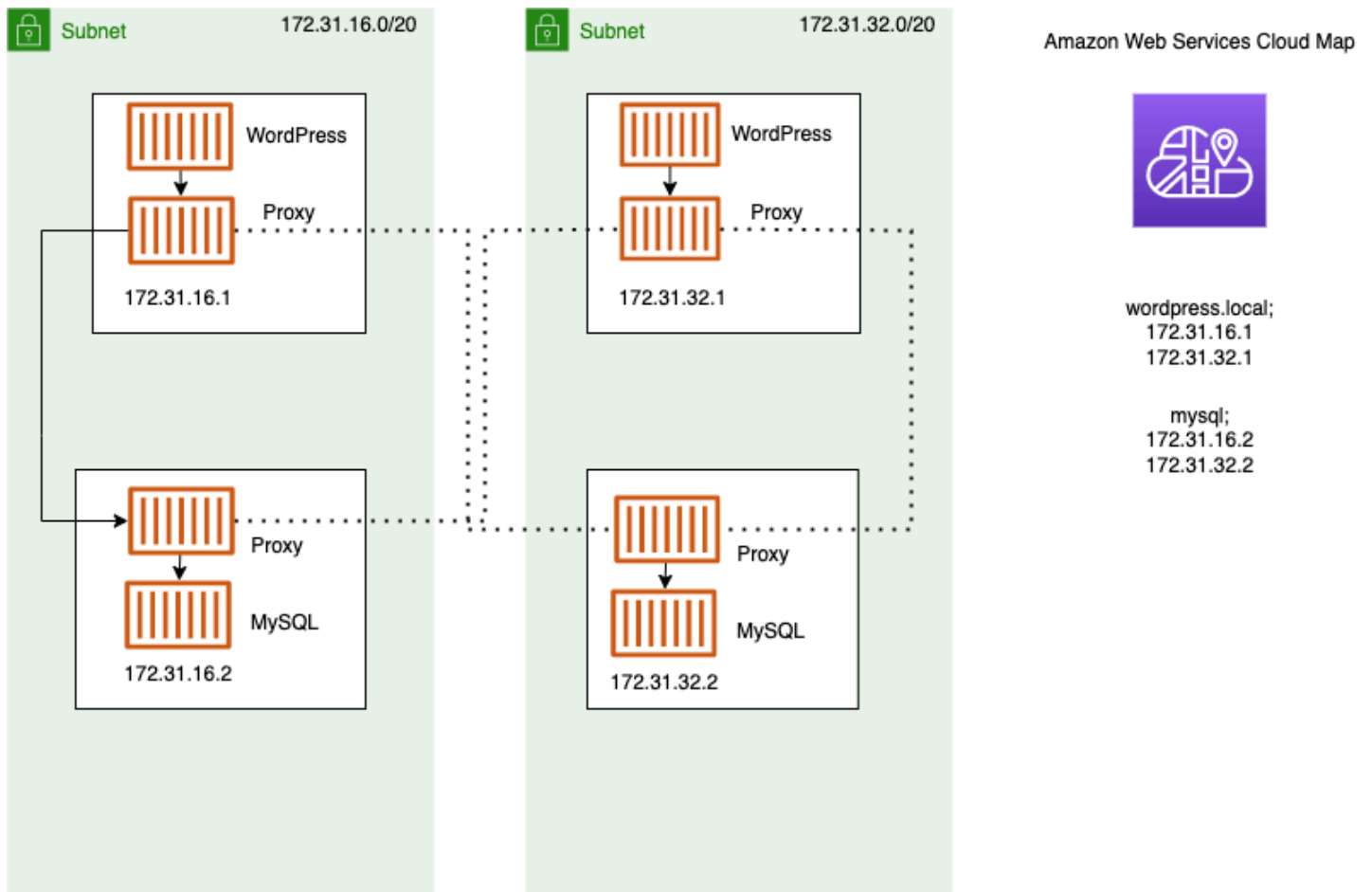
Opzioni di interconnessione	Collegato	awsvpc	Host
Individuazione dei servizi	sì, ma richiede che i clienti conoscano i record SRV nel DNS senza hostPort.	sì	sì, ma richiede che i clienti conoscano i record SRV nel DNS senza hostPort.
Service Connect	sì	sì	no
VPC Lattice	sì	sì	sì

Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi

Amazon ECS Service Connect fornisce la gestione della service-to-service comunicazione come configurazione Amazon ECS. Crea sia l'individuazione dei servizi che una rete di servizi in Amazon ECS. Ciò fornisce la configurazione completa all'interno di ogni servizio che gestisci tramite distribuzioni di servizi, un modo unificato di fare riferimento ai tuoi servizi all'interno di namespace che

non dipendono dalla configurazione DNS VPC e metriche e log standardizzati per monitorare tutte le tue applicazioni. Service Connect interconnette solo i servizi.

Il diagramma seguente mostra un esempio di rete Service Connect con 2 sottoreti nel VPC e 2 servizi. Un servizio client che viene eseguito WordPress con 1 attività in ciascuna sottorete. Un servizio server che esegue MySQL con 1 attività in ogni sottorete. Entrambi i servizi sono altamente disponibili e resistenti ai problemi relativi alle attività e alle zone di disponibilità, poiché ogni servizio esegue più attività distribuite su 2 sottoreti. Le frecce piene mostrano una connessione da WordPress a MySQL. Ad esempio, un comando `mysql --host=mysql CLI` che viene eseguito dall'interno del WordPress contenitore nell'operazione con l'indirizzo IP. `172.31.16.1` Il comando utilizza il nome breve `mysql` sulla porta predefinita per MySQL. Questo nome e questa porta si connettono al proxy Service Connect nella stessa attività. Il proxy utilizzato nell' WordPress operazione utilizza il bilanciamento del carico a tutto tondo e tutte le informazioni relative agli errori precedenti nel rilevamento dei valori anomali per scegliere a quale task MySQL connettersi. Come mostrato dalle frecce piene nel diagramma, il proxy si connette al secondo proxy nell'attività MySQL con l'indirizzo IP `172.31.16.2`. Il secondo proxy si connette al server MySQL locale nella stessa attività. Entrambi i proxy riportano le prestazioni di connessione visibili nei grafici nelle console Amazon ECS e CloudWatch Amazon, in modo da poter ottenere i parametri delle prestazioni da tutti i tipi di applicazioni allo stesso modo.



I seguenti termini sono utilizzati insieme a Service Connect.

nome porta

La configurazione della definizione di attività di Amazon ECS che assegna un nome a una particolare mappatura delle porte. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

alias del cliente

La configurazione del servizio Amazon ECS che assegna il numero di porta utilizzato nell'endpoint. Inoltre, l'alias del client può assegnare il nome DNS dell'endpoint, sovrascrivendo il nome del rilevamento. Se nel servizio Amazon ECS non viene fornito un nome di rilevamento, l'alias del client sostituisce il nome della porta come nome dell'endpoint. Per esempi di endpoint, consulta la definizione di endpoint. È possibile assegnare più alias client a un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

nome del rilevamento

Il nome intermedio opzionale che è possibile creare per una porta specificata dalla definizione di attività. Questo nome viene utilizzato per creare un servizio. AWS Cloud Map Se questo nome non viene fornito, viene utilizzato il nome della porta indicato nella definizione di attività. È possibile assegnare più nomi di rilevamento a una porta specifica di un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

AWS Cloud Map i nomi dei servizi devono essere univoci all'interno di un namespace. A causa di questa limitazione, è possibile avere una sola configurazione di Service Connect senza un nome di rilevamento per una particolare definizione di attività in ogni spazio dei nomi.

endpoint

L'URL per connettersi a un'API o a un sito Web. L'URL contiene il protocollo, un nome DNS e la porta. Per ulteriori informazioni sugli endpoint in generale, consulta la sezione [Endpoint](#) nel glossario AWS all'interno della documentazione Riferimenti generali di Amazon Web Services.

Service Connect crea endpoint che si connettono ai servizi Amazon ECS e configura le attività nei servizi Amazon ECS per connettersi agli endpoint. L'URL contiene il protocollo, un nome DNS e la porta. Il protocollo e il nome della porta vengono selezionati nella definizione dell'attività, poiché la porta deve corrispondere all'applicazione che si trova all'interno dell'immagine di container. Nel servizio, si seleziona ogni porta in base al nome e si può assegnare il nome DNS. Se non si specifica un nome DNS nella configurazione del servizio Amazon ECS, per impostazione predefinita viene utilizzato il nome della porta indicato nella definizione di attività. Ad esempio, un endpoint Service Connect potrebbe essere `http://blog:80`, `grpc://checkout:8080` o `http://_db.production.internal:99`.

Servizio Service Connect

La configurazione di un singolo endpoint in un servizio Amazon ECS. Questa è una parte della configurazione Service Connect, costituita da una singola riga nella configurazione di Service Connect e del nome del rilevamento nella console o da un oggetto nell'elenco `services` nella configurazione JSON di un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

Per ulteriori informazioni, consulta il riferimento [ServiceConnectService](#) all'API di Amazon Elastic Container Service.

spazio dei nomi

Il nome breve o il nome completo Amazon Resource Name (ARN) dello spazio dei AWS Cloud Map nomi da utilizzare con Service Connect. Lo spazio dei nomi deve trovarsi nello stesso spazio del servizio Regione AWS e del cluster Amazon ECS. Il tipo di namespace in AWS Cloud Map non influisce su Service Connect.

Service Connect utilizza lo spazio dei AWS Cloud Map nomi come raggruppamento logico di attività Amazon ECS che comunicano tra loro. Ogni servizio Amazon ECS può appartenere a un solo spazio dei nomi. I servizi all'interno di un namespace possono essere distribuiti su diversi cluster Amazon ECS all'interno dello stesso nello stesso Regione AWS . Account AWS Puoi organizzare liberamente i servizi in base a qualsiasi criterio.

servizi client

Un servizio che esegue un'applicazione client di rete. Questo servizio deve avere uno spazio dei nomi configurato. Ogni attività del servizio può rilevare e connettersi a tutti gli endpoint nello spazio dei nomi tramite un container proxy Service Connect.

Se uno dei container dell'attività deve connettersi a un endpoint da un servizio in uno spazio dei nomi, scegli un servizio client. Se un'applicazione front-end, un proxy inverso o un sistema di bilanciamento del carico riceve traffico esterno tramite altri metodi come Elastic Load Balancing, potrebbe utilizzare questo tipo di configurazione di Service Connect.

servizio client-server

Un servizio Amazon ECS che esegue un'applicazione di rete o del servizio Web. Questo servizio deve avere uno spazio dei nomi e almeno un endpoint configurati. Ogni attività del servizio è raggiungibile utilizzando gli endpoint. Il container proxy Service Connect ascolta il nome e la porta dell'endpoint per indirizzare il traffico verso i container delle app nell'attività.

Se uno dei container espone e ascolta il traffico di rete su una porta, scegli un servizio client-server. Queste applicazioni non devono connettersi ad altri servizi client-server nello stesso spazio dei nomi, ma è necessaria la configurazione del client. Un backend, un middleware, un livello aziendale o la maggior parte dei microservizi possono utilizzare questo tipo di configurazione Service Connect. Se desideri che un'applicazione front-end, un proxy inverso o un sistema di bilanciamento del carico riceva traffico da altri servizi configurati con Service Connect nello stesso spazio dei nomi, questi servizi devono utilizzare questo tipo di configurazione Service Connect.

La funzionalità Service Connect crea una rete virtuale di servizi correlati. La stessa configurazione del servizio può essere utilizzata su più spazi dei nomi diversi per eseguire set di applicazioni indipendenti ma identici. Service Connect definisce il container del proxy nel servizio Amazon ECS. In questo modo, la stessa definizione di attività può essere utilizzata per eseguire applicazioni identiche in spazi dei nomi diversi con configurazioni Service Connect diverse. Ogni attività eseguita dal servizio esegue un contenitore proxy all'interno dell'attività.

Service Connect è adatto per connessioni tra servizi Amazon ECS all'interno dello stesso spazio dei nomi. Per le seguenti applicazioni, è necessario utilizzare un metodo di interconnessione aggiuntivo per connettersi a un servizio Amazon ECS configurato con Service Connect:

- Attività configurate in altri namespace
- Attività non configurate per Service Connect
- Altre applicazioni esterne ad Amazon ECS

Queste applicazioni possono connettersi tramite il proxy Service Connect, ma non possono risolvere i nomi degli endpoint Service Connect.

Affinché queste applicazioni risolvano gli indirizzi IP delle attività di Amazon ECS, è necessario utilizzare un altro metodo di interconnessione.

Argomenti

- [Prezzi](#)
- [Componenti Amazon ECS Service Connect](#)
- [Panoramica della configurazione di Amazon ECS Service Connect](#)
- [Crittografa il traffico Amazon ECS Service Connect](#)
- [Configurazione di Amazon ECS Service Connect con AWS CLI](#)

Prezzi

- I prezzi di Amazon ECS Service Connect dipendono dall'utilizzo EC2 dell'infrastruttura Amazon AWS Fargate per ospitare i carichi di lavoro containerizzati. Quando si utilizza Amazon ECS su AWS Outposts, i prezzi seguono lo stesso modello utilizzato quando si utilizza EC2 direttamente Amazon. Per ulteriori informazioni, consulta [Prezzi di Amazon ECS](#).
- Non sono previsti costi aggiuntivi per l'utilizzo di Amazon ECS Service Connect.
- AWS Cloud Map l'utilizzo è completamente gratuito se utilizzato da Service Connect.

- I clienti pagano per le risorse di elaborazione utilizzate da Amazon ECS Service Connect, tra cui vCPU e memoria. Poiché l'agente Amazon ECS Service Connect viene eseguito all'interno di un'attività del cliente, non sono previsti costi aggiuntivi per eseguirla. Le risorse relative alle attività sono condivise tra il carico di lavoro del cliente e l'agente Amazon ECS Service Connect.
- Quando utilizzano la funzionalità di crittografia del traffico di Amazon ECS Service Connect con AWS Private CA, i clienti pagano per l'autorità di certificazione privata che creano e per ogni certificato TLS emesso. [Per ulteriori dettagli, consulta AWS Private Certificate Authority la pagina dei prezzi](#). Per stimare il costo mensile dei certificati TLS, i clienti devono conoscere il numero di servizi Amazon ECS con TLS abilitato, moltiplicarlo per il costo del certificato e quindi moltiplicarlo per sei. Poiché Amazon ECS Service Connect ruota automaticamente i certificati TLS ogni cinque giorni, vengono emessi in media sei certificati per servizio Amazon ECS, al mese.

Componenti Amazon ECS Service Connect

Quando usi Amazon ECS Service Connect, configuri ogni servizio Amazon ECS per eseguire un'applicazione server che riceve richieste di rete (servizio client-server) o per eseguire un'applicazione client che effettua le richieste (servizio client).

Quando ti prepari a iniziare a usare Service Connect, inizia con un servizio client-server. È possibile aggiungere una configurazione Service Connect a un nuovo servizio o a un servizio esistente. Amazon ECS crea un endpoint Service Connect nello spazio dei nomi. Inoltre, Amazon ECS crea una nuova implementazione nel servizio per sostituire le attività attualmente in esecuzione.

Le attività esistenti e altre applicazioni potranno continuare a connettersi agli endpoint esistenti e alle applicazioni esterne. Se un servizio client-server aggiunge attività scalando orizzontalmente, le nuove connessioni dei client verranno bilanciate tra tutte le attività. Se un servizio client-server viene aggiornato, le nuove connessioni dei client verranno bilanciate tra le attività della nuova versione.

Le attività esistenti non possono essere risolte e connettersi al nuovo endpoint. Solo le nuove attività con una configurazione Service Connect nello stesso namespace e che iniziano a essere eseguite dopo questa distribuzione possono risolversi e connettersi a questo endpoint.

Ciò significa che l'operatore dell'applicazione client determina quando la configurazione della propria app cambia, anche se l'operatore dell'applicazione server può modificare la configurazione in qualsiasi momento. L'elenco degli endpoint nel namespace può cambiare ogni volta che viene distribuito un servizio nel namespace. Le attività esistenti e le attività sostitutive continuano a comportarsi come dopo la distribuzione più recente.

Considera i seguenti esempi:

Innanzitutto, si supponga di creare un'applicazione disponibile sulla rete Internet pubblica in un unico AWS CloudFormation modello e in un unico AWS CloudFormation stack. La scoperta pubblica e la raggiungibilità dovrebbero essere create per ultimo AWS CloudFormation, incluso il servizio client frontend. I servizi devono essere creati in questo ordine per evitare un periodo di tempo in cui il servizio client di front-end è attivo e disponibile al pubblico, ma un back-end non lo è. Ciò elimina l'invio di messaggi di errore al pubblico durante quel periodo di tempo. Nel AWS CloudFormation, devi utilizzare il `dependsOn` per indicare AWS CloudFormation che non è possibile creare più servizi Amazon ECS in parallelo o contemporaneamente. È necessario aggiungere `dependsOn` al servizio client di front-end per ogni servizio client-server di back-end a cui si connettono le attività del client.

In secondo luogo, supponiamo che esista un servizio di front-end senza la configurazione di Service Connect. Le attività si connettono a un servizio di back-end esistente. Aggiungi prima una configurazione di Service Connect client-server al servizio di back-end utilizzando lo stesso nome nel DNS o `clientAlias` utilizzato dal front-end. Questo crea una nuova distribuzione, quindi tutti i metodi di rilevamento del rollback della distribuzione AWS SDKs e altri metodi per ripristinare e ripristinare il servizio di backend alla distribuzione e alla configurazione precedenti. AWS Management Console AWS CLI Se sei soddisfatto delle prestazioni e del comportamento del servizio di back-end, aggiungi una configurazione di Service Connect client o client-server al servizio di front-end. Solo le attività della nuova implementazione utilizzano il proxy Service Connect aggiunto a quelle nuove attività. In caso di problemi con questa configurazione, è possibile eseguire il rollback e ripristinare la configurazione precedente utilizzando il rilevamento del rollback di distribuzione o AWS Management Console altri metodi per ripristinare AWS SDKs e ripristinare il servizio di backend alla distribuzione e alla configurazione precedenti. AWS CLI Se si utilizza un altro sistema di individuazione dei servizi basato sul DNS anziché su Service Connect, qualsiasi applicazione frontend o client inizia a utilizzare nuovi endpoint e modifica la configurazione degli endpoint dopo la scadenza della cache DNS locale, in genere impiegando diverse ore.

Rete

Per impostazione predefinita, il proxy Service Connect ascolta la mappatura delle porte `containerPort` dalla definizione dell'attività. Le regole del gruppo di sicurezza devono consentire il traffico in entrata (in ingresso) verso questa porta dalle sottoreti in cui verranno eseguiti i client.

Anche se si imposta un numero di porta nella configurazione del servizio Service Connect, ciò non cambia la porta per il servizio client-server su cui è in ascolto il proxy Service Connect. Quando imposti questo numero di porta, Amazon ECS modifica la porta dell'endpoint a cui si connettono i servizi client, sul proxy Service Connect all'interno di tali attività. Il proxy nel servizio client si connette al proxy nel servizio client-server utilizzando la `containerPort`.

Se desideri modificare la porta su cui il proxy Service Connect è in ascolto, modifica `ingressPortOverride` nella configurazione di Service Connect del servizio client-server. Se si modifica questo numero di porta, è necessario consentire il traffico in entrata su questa porta utilizzata dal traffico diretto a questo servizio.

Il traffico inviato dalle applicazioni ai servizi Amazon ECS configurati per Service Connect richiede che Amazon VPC e le sottoreti dispongano di regole della tabella di instradamento e di regole dell'ACL di rete che consentano i numeri di porta `containerPort` e `ingressPortOverride` che si stanno utilizzando.

È possibile utilizzare Service Connect per inviare traffico tra VPCs. Gli stessi requisiti per le regole della tabella di routing, la rete ACLs e i gruppi di sicurezza si applicano a entrambi VPCs.

Ad esempio, due cluster creano attività diverse VPCs. Un servizio in ogni cluster è configurato per utilizzare lo stesso spazio dei nomi. Le applicazioni di questi due servizi possono risolvere ogni endpoint nello spazio dei nomi senza alcuna configurazione DNS del VPC. Tuttavia, i proxy non possono connettersi a meno che il peering VPC, le tabelle di routing VPC o subnet e la rete VPC non consentano il traffico sui numeri di porta e. ACLs `containerPort` `ingressPortOverride`

Per le attività che utilizzano la modalità `bridge` di rete, è necessario creare un gruppo di sicurezza con una regola in entrata che consenta il traffico sull'intervallo di porte dinamiche superiore. Quindi, assegna il gruppo di sicurezza a tutte le EC2 istanze del cluster Service Connect.

Proxy Service Connect

Se crei o aggiorni un servizio con la configurazione Service Connect, Amazon ECS aggiunge un nuovo contenitore a ogni nuova attività non appena viene avviata. Questo modello di utilizzo di un container separato è denominato `sidecar`. Questo container non è presente nella definizione dell'attività e non è possibile configurarlo. Amazon ECS gestisce la configurazione del contenitore nel servizio. Ciò consente di riutilizzare le stesse definizioni di attività tra più servizi, namespace e attività senza Service Connect.

Risorse proxy

- Per le definizioni delle attività, è necessario impostare i parametri di CPU e memoria.

Si consiglia di aggiungere altre 256 unità CPU e almeno 64 MiB di memoria alla CPU dell'attività e alla memoria per il contenitore proxy Service Connect. Su AWS Fargate, la quantità minima di memoria che è possibile impostare è 512 MiB. Su Amazon EC2, è richiesta la memoria per la definizione delle attività.

- Per il servizio, si imposta la configurazione del registro nella configurazione Service Connect.
- Se prevedi che le attività di questo servizio ricevano più di 500 richieste al secondo al massimo del carico, ti consigliamo di aggiungere 512 unità CPU alla CPU dell'attività in questa definizione di attività per il container del proxy Service Connect.
- Se prevedi di creare più di 100 servizi Service Connect nello spazio dei nomi o 2.000 attività in totale su tutti i servizi Amazon ECS all'interno dello spazio dei nomi, ti consigliamo di aggiungere 128 MiB di memoria alla memoria dell'attività per il container del proxy Service Connect. È necessario eseguire questa operazione in ogni definizione di attività utilizzata da tutti i servizi Amazon ECS nello spazio dei nomi.

Configurazione del proxy

Le applicazioni si connettono al proxy nel container sidecar nella stessa attività in cui si trova l'applicazione. Amazon ECS configura l'attività e i contenitori in modo che le applicazioni si connettano al proxy solo quando l'applicazione è connessa ai nomi degli endpoint nello stesso spazio dei nomi. Tutto il resto del traffico non utilizza il proxy. L'altro traffico include indirizzi IP nello stesso VPC, endpoint di AWS servizio e traffico esterno.

Bilanciamento del carico

Service Connect configura il proxy per utilizzare la strategia round-robin per il bilanciamento del carico tra le attività in un endpoint Service Connect. Il proxy locale, che si trova nell'attività da cui proviene la connessione, seleziona una delle attività del servizio client-server che fornisce l'endpoint.

Ad esempio, si consideri un'attività eseguita WordPress in un servizio configurato come servizio client in uno spazio dei nomi chiamato locale. È presente un altro servizio con 2 attività che eseguono il database MySQL. Questo servizio è configurato per fornire un endpoint denominato `mysql` tramite Service Connect nello stesso spazio dei nomi. Nell'WordPress operazione, l'WordPress applicazione si connette al database utilizzando il nome dell'endpoint. Le connessioni a questo nome vanno al proxy che viene eseguito in un contenitore laterale durante la stessa attività. Quindi, il proxy può connettersi a una delle attività MySQL utilizzando la strategia round-robin.

Strategie di bilanciamento del carico: round-robin

Rilevamento di anomalie

Questa funzionalità utilizza i dati acquisiti dal proxy sulle connessioni non riuscite precedenti per evitare l'invio di nuove connessioni agli host che le contenevano. Service Connect configura la funzionalità di rilevamento delle anomalie del proxy per fornire controlli dell'integrità passivi.

Utilizzando l'esempio precedente, il proxy può connettersi a entrambe le attività MySQL. Se il proxy ha effettuato più connessioni a un'attività MySQL specifica e negli ultimi 30 secondi 5 di queste, o un numero maggiore, non sono andate a buon fine, il proxy evita di eseguire tale attività MySQL per un periodo compreso tra 30 e 300 secondi.

Tentativi

Service Connect configura il proxy in modo da tentare nuovamente le connessioni che passano attraverso il proxy e falliscono. Il secondo tentativo evita di utilizzare l'host delle connessioni precedenti. In questo modo si garantisce che ogni connessione attraverso Service Connect non si interrompa per motivi isolati.

Numero di tentativi: 2

Timeout

Service Connect configura il proxy in modo che attenda un tempo massimo per la risposta delle applicazioni client-server. Il valore di timeout predefinito è di 15 secondi, ma può essere aggiornato.

Parametri opzionali:

`idleTimeout` - La quantità di tempo in secondi in cui una connessione rimane attiva mentre è inattiva. Un valore di `0` disabilita `idleTimeout`.

L'impostazione predefinita per `idleTimeout` per HTTP/HTTP2/GRPC è 5 minuti.

L'impostazione predefinita per `idleTimeout` per TCP è 1 ora.

`perRequestTimeout` - Il periodo di attesa che l'upstream risponda con una risposta completa per richiesta. Il valore di `0` si spegne. `perRequestTimeout` Questo può essere impostato solo quando il contenitore dell'applicazione `appProtocol` for è HTTP/HTTP2/GRPC. L'impostazione predefinita è 15 secondi.

 Note

Se `idleTimeout` è impostato su un tempo inferiore a `aperRequestTimeout`, la connessione si chiuderà quando `idleTimeout` viene raggiunto il `perRequestTimeout`.

Considerazioni

Quando utilizzi Service Connect, considera quanto segue:

- Le attività eseguite in Fargate devono utilizzare la versione della piattaforma Fargate Linux 1.4.0 o superiore per utilizzare Service Connect.
- La versione dell'agente Amazon ECS sull'istanza del contenitore deve essere uguale 1.67.2 o superiore.
- Per utilizzare Service Connect, le istanze di container devono eseguire la versione 20230428 dell'AMI di Amazon Linux 2023 ottimizzata per Amazon ECS, o versioni successive, o la versione 2.0.20221115 dell'AMI di Amazon Linux 2 ottimizzata per Amazon ECS. Oltre all'agente del container Amazon ECS, queste versioni contengono l'agente Service Connect. Per ulteriori informazioni sull'agente Service Connect, consulta [Amazon ECS Service Connect Agent](#) on GitHub.
- Le istanze di container devono disporre dell'autorizzazione `ecs:Poll` per la risorsa `arn:aws:ecs:region:0123456789012:task-set/cluster/*`. Se utilizzi il `ecsInstanceRole`, non è necessario aggiungere altre autorizzazioni. La policy gestita da `AmazonEC2ContainerServiceforEC2Role` dispone delle autorizzazioni necessarie. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).
- Solo i servizi che utilizzano implementazioni in sequenza sono supportati con Service Connect.
- Le attività che utilizzano la modalità di `bridge` rete e utilizzano Service Connect non supportano il parametro di definizione del `hostname` contenitore.
- Per utilizzare Service Connect, le definizioni di attività devono impostare il limite di memoria dell'attività. Per ulteriori informazioni, consulta [Proxy Service Connect](#).
- Le definizioni delle attività che impostano i limiti di memoria del contenitore non sono supportate.

È possibile impostare i limiti di memoria del container sui container, ma è necessario impostare il limite di memoria delle attività su un numero maggiore della somma dei limiti di memoria del container. La CPU e la memoria aggiuntive nei limiti delle attività che non vengono allocate nei limiti del container negli altri contenitori vengono utilizzate dal container del proxy Service Connect

e da altri container che non impostano limiti di container. Per ulteriori informazioni, consulta [Proxy Service Connect](#).

- È possibile configurare Service Connect per utilizzare qualsiasi spazio dei nomi AWS Cloud Map nella stessa regione nella stessa Account AWS
- Ogni servizio può appartenere a un solo namespace.
- Sono supportate solo le attività create dai servizi.
- Tutti gli endpoint devono essere univoci all'interno di uno spazio dei nomi.
- Tutti i nomi di rilevamento devono essere univoci all'interno di uno spazio dei nomi.
- È necessario ridistribuire i servizi esistenti prima che le applicazioni possano risolvere nuovi endpoint. I nuovi endpoint aggiunti allo spazio dei nomi dopo l'implementazione più recente non verranno aggiunti alla configurazione dell'attività. Per ulteriori informazioni, consulta [the section called "Componenti Service Connect"](#).
- Service Connect non elimina i namespace quando vengono eliminati i cluster. È necessario eliminare i namespace in. AWS Cloud Map
- Per impostazione predefinita, il traffico di Application Load Balancer viene instradato tramite l'agente Service Connect in modalità rete. `aws_vpc` Se desideri che il traffico non di servizio aggiri l'agente Service Connect, utilizza il [ingressPortOverride](#) parametro nella configurazione del servizio Service Connect.

Service Connect non supporta quanto segue:

- Container Windows
- HTTP 1.0
- Attività autonome
- Servizi che utilizzano i tipi di distribuzione blu/verde ed esterna
- Le istanze di container `External` per Amazon ECS Anywhere non sono supportate con Service Connect.
- PPv2

Panoramica della configurazione di Amazon ECS Service Connect

Quando si utilizza Service Connect, è necessario configurare dei parametri nelle risorse.

La tabella seguente descrive i parametri di configurazione per le risorse Amazon ECS.

Posizione dei parametri	Tipo di app	Descrizione	Richiesto
Definizione di attività	Client	Non sono disponibili modifiche per Service Connect nelle definizioni delle attività del client.	N/D
Definizione di attività	Client-server	I server devono aggiungere i campi name alle porte nelle portMappings dei container. Per ulteriori informazioni, consulta portMappings .	Sì
Definizione di attività	Client-server	I server possono opzionalmente fornire un protocollo applicativo (ad esempio, HTTP) per ricevere parametri specifici del protocollo per le loro applicazioni server (ad esempio, HTTP 5xx).	No
Definizioni di servizi	Client	I servizi client devono aggiungere una serviceConnectConfiguration per configurare lo spazio dei nomi a cui aderire. Questo spazio dei nomi deve contenere tutti i servizi del server che questo servizio deve individuare. Per ulteriori informazioni, consulta serviceConnectConfiguration .	Sì
Definizione di servizio	Client-server	I servizi server devono aggiungere una serviceConnectConfiguration per configurare i nomi DNS, i numeri di porta e lo spazio dei nomi da cui è disponibile il servizio. Per ulteriori informazioni, consulta serviceConnectConfiguration .	Sì
Cluster	Client	I cluster possono aggiungere uno spazio dei nomi Service Connect predefinito. Quando Service Connect è configurato	No

Posizione dei parametri	Tipo di app	Descrizione	Richiesto
		in un servizio, i nuovi servizi nel cluster ereditano lo spazio dei nomi.	
Cluster	Client-server	Non sono disponibili modifiche per Service Connect nei cluster che si applicano ai servizi server. Le definizioni e i servizi delle attività del server devono impostare la rispettiva configurazione.	N/D

Panoramica della procedura per configurare Service Connect

I passaggi seguenti forniscono una panoramica su come configurare Service Connect.

Important

- Service Connect crea AWS Cloud Map servizi nel tuo account. La modifica di queste risorse AWS Cloud Map mediante la registrazione/deregistrazione manuale delle istanze, la modifica degli attributi dell'istanza o l'eliminazione di un servizio può portare a un comportamento imprevisto per il traffico dell'applicazione o le implementazioni successive.
- Service Connect non supporta i collegamenti nella definizione dell'attività.

1. Aggiungi i nomi delle porte alle mappature delle porte nelle definizioni di attività. Per ottenere parametri aggiuntivi, puoi inoltre identificare il protocollo di livello 7 dell'applicazione.
2. Crea un cluster con uno spazio AWS Cloud Map dei nomi o crea lo spazio dei nomi separatamente. Per un'organizzazione semplice, crea un cluster con il nome che desideri per lo spazio dei nomi e specifica lo stesso nome per lo spazio dei nomi. In questo caso, Amazon ECS crea un nuovo spazio dei nomi HTTP con la configurazione necessaria. Service Connect non utilizza né crea zone ospitate DNS in Amazon Route 53.
3. Configura i servizi per creare endpoint Service Connect all'interno dello spazio dei nomi.
4. Implementa i servizi per creare gli endpoint. Amazon ECS aggiunge un container del proxy Service Connect a ogni attività e crea gli endpoint Service Connect in AWS Cloud Map. Questo container non è configurato nella definizione dell'attività e la definizione dell'attività può essere riutilizzata senza modifiche per creare più servizi nello stesso spazio dei nomi o in più spazi dei nomi.

5. Implementa le app client come servizi per connetterti agli endpoint. Amazon ECS li connette agli endpoint Service Connect tramite il proxy Service Connect in ogni attività.

Le applicazioni utilizzano il proxy solo per connettersi agli endpoint Service Connect. Non è disponibile alcuna configurazione aggiuntiva per l'utilizzo del proxy. Il proxy esegue il bilanciamento del carico round-robin, il rilevamento di anomalie ed effettua un nuovo tentativo. Per ulteriori informazioni sul proxy, consulta [Proxy Service Connect](#).

6. Monitora il traffico tramite il proxy Service Connect in Amazon CloudWatch.

Configurazione del cluster

È possibile impostare uno spazio dei nomi predefinito per Service Connect quando si crea o si aggiorna il cluster. Se si specifica un nome per lo spazio dei nomi che non esiste nella stessa Regione AWS e nello stesso account, viene creato un nuovo spazio dei nomi HTTP.

Se crei un cluster e specifichi uno spazio dei nomi Service Connect predefinito, il cluster attende lo stato PROVISIONING mentre Amazon ECS crea lo spazio dei nomi. Puoi vedere un attachment nello stato del cluster che mostra lo stato dello spazio dei nomi. Per impostazione predefinita, gli allegati non vengono visualizzati in. È necessario aggiungerli `--include ATTACHMENTS` per visualizzarli. AWS CLI

Configurazione del servizio

Service Connect è progettato per richiedere la configurazione minima. È necessario impostare un nome per ogni mappatura delle porte che si desidera utilizzare con Service Connect nella definizione di attività. Nel servizio, per creare un servizio client è necessario attivare Service Connect e selezionare uno spazio dei nomi. Per creare un servizio client-server, è necessario aggiungere una singola configurazione del servizio Service Connect che corrisponda al nome di una delle mappature delle porte. Amazon ECS riutilizza il numero di porta e il nome della porta dalla definizione dell'attività per definire il servizio e l'endpoint Service Connect. Per sovrascrivere questi valori, puoi utilizzare gli altri parametri Discovery, DNS e Port nella console o `discoveryName` e `clientAliases`, rispettivamente, nell'API Amazon ECS.

Crittografa il traffico Amazon ECS Service Connect

Amazon ECS Service Connect supporta la crittografia automatica del traffico con certificati Transport Layer Security (TLS) per i servizi Amazon ECS. Quando indirizzi i tuoi servizi Amazon ECS verso un [AWS Private Certificate Authority \(AWS Private CA\)](#), Amazon ECS fornisce automaticamente

certificati TLS per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect. Amazon ECS genera, ruota e distribuisce i certificati TLS utilizzati per la crittografia del traffico.

La crittografia automatica del traffico con Service Connect utilizza funzionalità di crittografia leader del settore per proteggere le comunicazioni tra servizi e soddisfare i requisiti di sicurezza. Supporta certificati AWS Private Certificate Authority TLS con crittografia. 256-bit ECDSA 2048-bit RSA. Hai anche il pieno controllo sui certificati privati e sulle chiavi di firma per aiutarti a soddisfare i requisiti di conformità. Per impostazione predefinita, TLS 1.3 è supportato, ma TLS 1.0 - 1.2 non lo sono. Service Connect supporta TLS 1.3 con i seguenti codici:

- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

Note

Per utilizzare TLS 1.3, è necessario abilitarlo sul listener sulla destinazione.
Solo il traffico in entrata e in uscita che passa attraverso l'agente Amazon ECS è crittografato.

AWS Private Certificate Authority certificati e Service Connect

È necessario disporre del ruolo IAM dell'infrastruttura. Per ulteriori informazioni su questo ruolo, consulta il [ruolo IAM dell'infrastruttura Amazon ECS](#).

AWS Private Certificate Authority modalità per Service Connect

AWS Private Certificate Authority può funzionare in due modalità: generica e di breve durata.

- Scopo generale - Certificati che possono essere configurati con qualsiasi data di scadenza.
- Di breve durata - Certificati con una validità massima di sette giorni.

Sebbene Amazon ECS supporti entrambe le modalità, consigliamo di utilizzare certificati di breve durata. Per impostazione predefinita, i certificati ruotano ogni cinque giorni e l'esecuzione in modalità di breve durata offre risparmi significativi sui costi rispetto agli usi generici.

Service Connect non supporta la revoca dei certificati e sfrutta invece certificati di breve durata con rotazione frequente dei certificati. Hai l'autorità per modificare la frequenza di rotazione, disabilitare

o eliminare i segreti utilizzando la [rotazione gestita](#) in [Secrets Manager](#), ma ciò può comportare le seguenti possibili conseguenze.

- Frequenza di rotazione più breve - Una frequenza di rotazione più breve comporta costi più elevati a AWS Private CA causa di Secrets Manager AWS KMS e Auto Scaling che subiscono un maggiore carico di lavoro per la rotazione.
- Frequenza di rotazione più lunga - Le comunicazioni delle applicazioni falliscono se la frequenza di rotazione supera i sette giorni.
- Eliminazione del segreto - L'eliminazione del segreto comporta un errore di rotazione e influisce sulle comunicazioni con le applicazioni dei clienti.

Se la rotazione segreta fallisce, viene pubblicato un `RotationFailed` evento in [AWS CloudTrail](#). Puoi anche impostare un [CloudWatch allarme](#) per `RotationFailed`.

Important

Non aggiungere regioni di replica ai segreti. In questo modo si impedisce ad Amazon ECS di eliminare il segreto, poiché Amazon ECS non dispone dell'autorizzazione per rimuovere le regioni dalla replica. Se hai già aggiunto la replica, esegui il seguente comando.

```
aws secretsmanager remove-regions-from-replication \  
--secret-id SecretId \  
--remove-replica-regions region-name
```

Autorità di certificazione subordinate

È possibile importare qualsiasi certificato AWS Private CA, root o subordinato, a Service Connect TLS per emettere certificati di entità finale per i servizi. L'emittente fornito viene considerato il firmatario e la fonte della fiducia ovunque. È possibile emettere certificati di entità finale per diverse parti dell'applicazione da diversi subordinati. CAs Quando utilizzi AWS CLI, fornisci l'Amazon Resource Name (ARN) della CA per stabilire la catena di fiducia.

Autorità di certificazione locali

Per utilizzare la CA locale, è necessario creare e configurare una CA subordinata in. AWS Private Certificate Authority Ciò garantisce che tutti i certificati TLS emessi per i carichi di lavoro Amazon

ECS condividano la catena di fiducia con i carichi di lavoro eseguiti in locale e siano in grado di connettersi in modo sicuro.

 Important

Aggiungi il tag richiesto nel tuo. `AmazonECSManaged` : `true` AWS Private CA

infrastruttura come codice

Quando si utilizzano Service Connect TLS con gli strumenti Infrastructure as Code (IaC), è importante configurare correttamente le dipendenze per evitare problemi, come i servizi bloccati. La AWS KMS chiave, se fornita, il ruolo IAM e AWS Private CA le dipendenze devono essere eliminati dopo il servizio Amazon ECS.

Service Connect e Secrets Manager

Quando si utilizza la crittografia Amazon ECS Service Connect with TLS, il servizio interagisce con Secrets Manager nei seguenti modi:

Service Connect utilizza il ruolo di infrastruttura fornito per creare segreti all'interno di Secrets Manager. Questi segreti vengono utilizzati per archiviare le chiavi private associate ai certificati TLS per crittografare il traffico tra i servizi Service Connect.

 Warning

La creazione e la gestione automatiche di questi segreti da parte di Service Connect semplificano il processo di implementazione della crittografia TLS per i tuoi servizi. Tuttavia, è importante essere consapevoli delle potenziali implicazioni per la sicurezza. Altri ruoli IAM che hanno accesso in lettura a Secrets Manager potrebbero essere in grado di accedere a questi segreti creati automaticamente. Ciò potrebbe esporre materiale crittografico sensibile a soggetti non autorizzati, se i controlli di accesso non sono configurati correttamente.

Per mitigare questo rischio, segui queste best practice:

- Gestisci e limita con attenzione l'accesso a Secrets Manager, in particolare per i segreti creati da Service Connect.
- Verifica regolarmente i ruoli IAM e le relative autorizzazioni per garantire il rispetto del principio del privilegio minimo.

Quando concedi l'accesso in lettura a Secrets Manager, valuta la possibilità di escludere le chiavi private TLS create da Service Connect. Puoi farlo utilizzando una condizione nelle tue policy IAM per escludere i segreti ARNs che corrispondono allo schema:

```
"arn:aws:secretsmanager:::secret:ecs-sc!"
```

Un esempio di policy IAM che nega l'GetSecretValueazione a tutti i segreti con il ecs-sc! prefisso:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "secretsmanager:GetSecretValue",
      "Resource": "arn:aws:secretsmanager:*:*:secret:ecs-sc!*"
    }
  ]
}
```

Note

Questo è un esempio generale e potrebbe essere necessario modificarlo in base al caso d'uso specifico e alla configurazione AWS dell'account. Testa sempre a fondo le tue policy IAM per assicurarti che forniscano l'accesso previsto mantenendo al contempo la sicurezza.

Comprendendo come Service Connect interagisce con Secrets Manager, puoi gestire meglio la sicurezza dei tuoi servizi Amazon ECS sfruttando al contempo i vantaggi della crittografia TLS automatica.

Service Connect e AWS Key Management Service

È possibile utilizzare [AWS Key Management Service](#) per crittografare e decrittografare le risorse Service Connect. AWS KMS è un servizio gestito AWS in cui è possibile creare e gestire chiavi crittografiche per proteggere i dati.

Quando si utilizza AWS KMS con Service Connect, è possibile scegliere di utilizzare una chiave di AWS proprietà che AWS gestisca per conto proprio oppure scegliere una AWS KMS chiave esistente. Puoi anche [creare una nuova AWS KMS chiave](#) da usare.

Fornire la propria chiave di crittografia

Puoi fornire i tuoi materiali chiave oppure puoi utilizzare un archivio di chiavi esterno tramite AWS Key Management Service. Importa la tua chiave in AWS KMS, quindi specificare l'Amazon Resource Name (ARN) di quella chiave in Amazon ECS Service Connect.

Di seguito è riportato un esempio AWS KMS di policy. Sostituisci i *user input* valori con i tuoi.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "id",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/role-name"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyPair"
      ],
      "Resource": "*"
    }
  ]
}
```

Per ulteriori informazioni sulle politiche chiave, consulta [Creazione di una politica chiave](#) nella Guida per gli AWS Key Management Service sviluppatori.

Note

Service Connect supporta solo chiavi di crittografia AWS KMS simmetriche. Non è possibile utilizzare nessun altro tipo di AWS KMS chiave per crittografare le risorse Service Connect. Per informazioni su come determinare se una AWS KMS chiave è una chiave di crittografia simmetrica, consulta [Identificare le chiavi KMS asimmetriche](#).

[Per ulteriori informazioni sulle chiavi di crittografia AWS Key Management Service simmetriche, consulta Chiavi di crittografia simmetriche nella Guida per gli sviluppatori. AWS KMS AWS Key Management Service](#)

Abilitazione del TLS per Amazon ECS Service Connect

La crittografia del traffico viene abilitata quando si crea o si aggiorna un servizio Service Connect.

Per abilitare la crittografia del traffico per un servizio in uno spazio dei nomi esistente utilizzando AWS Management Console

1. È necessario disporre del ruolo IAM dell'infrastruttura. Per ulteriori informazioni su questo ruolo, consulta il [ruolo IAM dell'infrastruttura Amazon ECS](#).
2. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
3. Nel riquadro di navigazione seleziona Namespaces (Spazio dei nomi).
4. Scegli il Namespace con il servizio per cui desideri abilitare la crittografia del traffico.
5. Scegli il servizio per cui desideri abilitare la crittografia del traffico.
6. Scegli Update Service nell'angolo in alto a destra e scorri verso il basso fino alla sezione Service Connect.
7. Scegli Attiva la crittografia del traffico tra le informazioni sul servizio per abilitare TLS.
8. Per il ruolo TLS di Service Connect, scegli un ruolo IAM dell'infrastruttura esistente o creane uno nuovo.
9. Per l'autorità di certificazione Signer, scegli un'autorità di certificazione esistente o creane una nuova.

Per ulteriori informazioni, consulta le [AWS Private Certificate Authority certificati e Service Connect](#).

10. Per Choose an AWS KMS key, scegli una chiave AWS proprietaria e gestita oppure puoi scegliere una chiave diversa. Puoi anche scegliere di crearne una nuova.

Per un esempio di utilizzo di TLS AWS CLI per configurare il servizio, [Configurazione di Amazon ECS Service Connect con AWS CLI](#).

La verifica che TLS sia abilitato per Amazon ECS Service Connect

Service Connect avvia TLS nell'agente Service Connect e lo termina nell'agente di destinazione. Di conseguenza, il codice dell'applicazione non vede mai le interazioni TLS. Utilizza i passaggi seguenti per verificare che TLS sia abilitato.

1. Includi la `openssl` CLI nell'immagine dell'applicazione.
2. Abilita [ECS Exec](#) sui tuoi servizi per connetterti alle tue attività tramite SSM. In alternativa, puoi avviare un' EC2 istanza Amazon nello stesso Amazon VPC del servizio.
3. Recupera l'IP e la porta di un'attività da un servizio che desideri verificare. È possibile recuperare l'indirizzo IP dell'attività nella AWS Cloud Map console. Le informazioni si trovano nella pagina dei dettagli del servizio relativa al namespace.
4. Accedi a una qualsiasi delle tue attività utilizzando `execute-command` like nell'esempio seguente. In alternativa, accedi all' EC2 istanza Amazon creata nel passaggio 2.

```
$ aws ecs execute-command --cluster cluster-name \  
  --task task-id \  
  --container container-name \  
  --interactive \  
  --command "/bin/sh"
```

Note

La chiamata diretta al nome DNS non rivela il certificato.

5. Nella shell connessa, utilizza la `openssl` CLI per verificare e visualizzare il certificato allegato all'attività.

Esempio:

```
openssl s_client -connect 10.0.147.43:6379 < /dev/null 2> /dev/null \  
| openssl x509 -noout -text
```

Risposta di esempio:

```
Certificate:  
  Data:  
    Version: 3 (0x2)  
    Serial Number:
```

```

    <serial-number>
Signature Algorithm: ecdsa-with-SHA256
Issuer: <issuer>
Validity
    Not Before: Jan 23 21:38:12 2024 GMT
    Not After : Jan 30 22:38:12 2024 GMT
Subject: <subject>
Subject Public Key Info:
    Public Key Algorithm: id-ecPublicKey
    Public-Key: (256 bit)
    pub:
        <pub>
    ASN1 OID: prime256v1
    NIST CURVE: P-256
X509v3 extensions:
    X509v3 Subject Alternative Name:
        DNS:redis.yelb-cftc
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Authority Key Identifier:
        keyid:<key-id>

    X509v3 Subject Key Identifier:
        1D:<id>
    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication, TLS Web Client Authentication
Signature Algorithm: ecdsa-with-SHA256
    <hash>

```

Configurazione di Amazon ECS Service Connect con AWS CLI

Puoi creare un servizio Amazon ECS per un'attività Fargate che utilizza Service Connect con. AWS CLI

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

Prerequisiti

Di seguito sono riportati i prerequisiti di Service Connect:

- Verificare che sia installata e configurata AWS CLI la versione più recente di. Per ulteriori informazioni, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy [Amazon ECS_FullAccess](#) IAM.
- Sono disponibili un VPC, una sottorete, una tabella di instradamento e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).
- Hai un ruolo di esecuzione delle attività con il nome `ecsTaskExecutionRole` e la policy gestita da `AmazonECSTaskExecutionRolePolicy` è associata al ruolo. Questo ruolo consente a Fargate di scrivere i log delle applicazioni NGINX e i log del proxy Service Connect su Amazon Logs. CloudWatch Per ulteriori informazioni, consulta [Creazione del ruolo di esecuzione attività](#).

Fase 1: Creazione del cluster

Utilizza la procedura seguente per creare un cluster e uno spazio dei nomi Amazon ECS.

Per creare il cluster e AWS Cloud Map lo spazio dei nomi Amazon ECS

1. Crea un cluster Amazon ECS denominato `tutorial` da utilizzare. Il parametro `--service-connect-defaults` imposta lo spazio dei nomi predefinito del cluster. Nell'output di esempio, uno AWS Cloud Map spazio dei nomi con il nome `service-connect` non esiste in questo account e Regione AWS quindi lo spazio dei nomi viene creato da Amazon ECS. Lo spazio dei nomi è creato in AWS Cloud Map nell'account ed è visibile insieme a tutti gli altri spazi dei nomi, quindi utilizza un nome che ne indichi lo scopo.

```
aws ecs create-cluster --cluster-name tutorial --service-connect-defaults  
namespace=service-connect
```

Output:

```
{  
  "cluster": {  
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",  
    "clusterName": "tutorial",
```

```

    "serviceConnectDefaults": {
      "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-EXAMPLE"
    },
    "status": "PROVISIONING",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [
      {
        "name": "containerInsights",
        "value": "disabled"
      }
    ],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": [],
    "attachments": [
      {
        "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "type": "sc",
        "status": "ATTACHING",
        "details": []
      }
    ],
    "attachmentsStatus": "UPDATE_IN_PROGRESS"
  }
}

```

2. Verifica che il cluster sia stato creato:

```
aws ecs describe-clusters --clusters tutorial
```

Output:

```

{
  "clusters": [
    {
      "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",
      "clusterName": "tutorial",

```

```

    "serviceConnectDefaults": {
      "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-EXAMPLE"
    },
    "status": "ACTIVE",
    "registeredContainerInstancesCount": 0,
    "runningTasksCount": 0,
    "pendingTasksCount": 0,
    "activeServicesCount": 0,
    "statistics": [],
    "tags": [],
    "settings": [],
    "capacityProviders": [],
    "defaultCapacityProviderStrategy": []
  },
  "failures": []
}

```

3. (Facoltativo) Verifica che lo spazio dei nomi sia stato creato in. AWS Cloud Map. È possibile utilizzare la configurazione AWS Management Console o la AWS CLI configurazione normale così come viene creata in. AWS Cloud Map

Ad esempio, utilizza la AWS CLI.

```
aws servicediscovery get-namespace --id ns-EXAMPLE
```

Output:

```

{
  "Namespace": {
    "Id": "ns-EXAMPLE",
    "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
EXAMPLE",
    "Name": "service-connect",
    "Type": "HTTP",
    "Properties": {
      "DnsProperties": {
        "SOA": {}
      },
      "HttpProperties": {

```

```
        "HttpName": "service-connect"
      },
    },
    "CreateDate": 1661749852.422,
    "CreatorRequestId": "service-connect"
  }
}
```

Fase 2: Creare il servizio per il server

La funzionalità Service Connect è progettata per l'interconnessione di più applicazioni su Amazon ECS. Almeno una di queste applicazioni deve fornire un servizio Web a cui connettersi. In questa fase viene creata:

- La definizione dell'attività che utilizza l'immagine del container NGINX ufficiale non modificata e include la configurazione di Service Connect.
- La definizione del servizio Amazon ECS che configura Service Connect per fornire il rilevamento del servizio e il proxy service mesh per il traffico verso questo servizio. La configurazione riutilizza lo spazio dei nomi predefinito della configurazione del cluster per ridurre la quantità di operazioni di configurazione del servizio effettuata per ciascun servizio.
- Il servizio Amazon ECS. Esegue un'attività utilizzando la definizione di attività e inserisce un container aggiuntivo per il proxy Service Connect. Il proxy Service Connect è in ascolto sulla porta dalla mappatura della porta del container nella definizione di attività. In un'applicazione client in esecuzione su Amazon ECS, il proxy nell'attività client ascolta le connessioni in uscita al nome della porta di definizione dell'attività, al nome della porta di individuazione del servizio o al nome alias del client del servizio e al numero di porta dall'alias del client.

Per creare il servizio Web con Service Connect

1. Registra una definizione di attività che sia compatibile con Fargate e utilizzi la modalità di rete awsvpc. Completare la procedura riportata di seguito.
 - a. Crea un file denominato `service-connect-nginx.json` con il contenuto della seguente definizione di attività.

Questa definizione di attività configura Service Connect aggiungendo i parametri `name` e `appProtocol` alla mappatura delle porte. Il nome della porta rende questa porta più

identificabile nella configurazione del servizio quando vengono utilizzate più porte. Il nome della porta viene utilizzato anche per impostazione predefinita come nome individuabile per essere utilizzato da altre applicazioni nello spazio dei nomi.

La definizione dell'attività contiene il ruolo IAM dell'attività perché il servizio ha abilitato l'esecuzione ECS.

 Important

Questa definizione di attività utilizza `logConfiguration` a per inviare l'output `nginx` da `stdout` e `stderr` verso Amazon Logs. CloudWatch Questo ruolo di esecuzione delle attività non dispone delle autorizzazioni aggiuntive necessarie per creare il CloudWatch gruppo di log Logs. Crea il gruppo di log in CloudWatch Logs utilizzando o. AWS Management Console AWS CLI Se non desideri inviare i log di `nginx` a Logs, puoi rimuovere CloudWatch il. `logConfiguration` Sostituisci l' Account AWS id nel ruolo di esecuzione dell'attività con il tuo id. Account AWS

```
{
  "family": "service-connect-nginx",
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
  "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskRole",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "webserver",
      "image": "public.ecr.aws/docker/library/nginx:latest",
      "cpu": 100,
      "portMappings": [
        {
          "name": "nginx",
          "containerPort": 80,
          "protocol": "tcp",
          "appProtocol": "http"
        }
      ],
      "essential": true,
      "logConfiguration": {
```

```
        "logDriver": "awslogs",
        "options": {
            "awslogs-group": "/ecs/service-connect-nginx",
            "awslogs-region": "region",
            "awslogs-stream-prefix": "nginx"
        }
    },
    ],
    "cpu": "256",
    "memory": "512"
}
```

- b. Registra la definizione di attività usando il file `service-connect-nginx.json`:

```
aws ecs register-task-definition --cli-input-json file://service-connect-nginx.json
```

2. Crea un servizio:

- a. Crea un file denominato `service-connect-nginx-service.json` con il contenuto del servizio Amazon ECS che stai per creare. Questo esempio usa la definizione di attività creata nella fase precedente. È obbligatorio l'uso di un parametro `awsvpcConfiguration` perché la definizione di attività di esempio usa la modalità di rete `awsvpc`.

Quando crei il servizio ECS, specifica il tipo di avvio Fargate e la versione della piattaforma LATEST che supporta Service Connect. I `securityGroups` e le `subnets` devono appartenere a un cloud VPC che soddisfi i requisiti per l'utilizzo di Amazon ECS. Puoi ottenere il gruppo di sicurezza e la sottorete IDs dalla console Amazon VPC.

Questo servizio configura Service Connect aggiungendo il parametro `serviceConnectConfiguration`. Lo spazio dei nomi non è richiesto perché il cluster ha uno spazio dei nomi predefinito configurato. Le applicazioni client in esecuzione in ECS nello spazio dei nomi si connettono a questo servizio utilizzando il `portName` e la porta negli `clientAliases`. Ad esempio, questo servizio è raggiungibile tramite `http://nginx:80/`, poiché nginx fornisce una pagina di benvenuto nella posizione `root /`. Le applicazioni esterne che non sono in esecuzione in Amazon ECS o che non si trovano nello stesso spazio dei nomi possono raggiungere questa applicazione tramite il proxy Service Connect utilizzando l'indirizzo IP dell'attività e il numero di porta indicato

nella definizione dell'attività. Per la `tls` configurazione, aggiungi il certificato `arn` per il `tuoawsPcaAuthorityArn`, il tuo `kmsKey` e il tuo `roleArn` ruolo IAM.

Questo servizio utilizza un `logConfiguration` per inviare l'output del proxy Service Connect da `stdout` e `stderr` verso Amazon CloudWatch Logs. Questo ruolo di esecuzione delle attività non dispone delle autorizzazioni aggiuntive necessarie per creare il gruppo di CloudWatch log Logs. Crea il gruppo di log in CloudWatch Logs utilizzando o. AWS Management Console AWS CLI Si consiglia di creare questo gruppo di log e di archiviare i log proxy in CloudWatch Logs. Se non desideri inviare i log del proxy a Logs, puoi CloudWatch rimuovere il `logConfiguration`

```
{
  "cluster": "tutorial",
  "deploymentConfiguration": {
    "maximumPercent": 200,
    "minimumHealthyPercent": 0
  },
  "deploymentController": {
    "type": "ECS"
  },
  "desiredCount": 1,
  "enableECSTags": true,
  "enableExecuteCommand": true,
  "launchType": "FARGATE",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "assignPublicIp": "ENABLED",
      "securityGroups": [
        "sg-EXAMPLE"
      ],
      "subnets": [
        "subnet-EXAMPLE",
        "subnet-EXAMPLE",
        "subnet-EXAMPLE"
      ]
    }
  },
  "platformVersion": "LATEST",
  "propagateTags": "SERVICE",
  "serviceName": "service-connect-nginx-service",
  "serviceConnectConfiguration": {
```

```

    "enabled": true,
    "services": [
      {
        "portName": "nginx",
        "clientAliases": [
          {
            "port": 80
          }
        ],
        "tls": {
          "issuerCertificateAuthority": {
            "awsPcaAuthorityArn": "certificateArn"
          },
          "kmsKey": "kmsKey",
          "roleArn": "iamRoleArn"
        }
      }
    ],
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-group": "/ecs/service-connect-proxy",
        "awslogs-region": "region",
        "awslogs-stream-prefix": "service-connect-proxy"
      }
    }
  },
  "taskDefinition": "service-connect-nginx"
}

```

- b. Crea un servizio utilizzando il `service-connect-nginx-service.json` file:

```
aws ecs create-service --cluster tutorial --cli-input-json file://service-connect-nginx-service.json
```

Output:

```

{
  "service": {
    "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/tutorial/service-connect-nginx-service",

```

```

    "serviceName": "service-connect-nginx-service",
    "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",
    "loadBalancers": [],
    "serviceRegistries": [],
    "status": "ACTIVE",
    "desiredCount": 1,
    "runningCount": 0,
    "pendingCount": 0,
    "launchType": "FARGATE",
    "platformVersion": "LATEST",
    "platformFamily": "Linux",
    "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
service-connect-nginx:1",
    "deploymentConfiguration": {
      "deploymentCircuitBreaker": {
        "enable": false,
        "rollback": false
      },
      "maximumPercent": 200,
      "minimumHealthyPercent": 0
    },
    "deployments": [
      {
        "id": "ecs-svc/3763308422771520962",
        "status": "PRIMARY",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/service-connect-nginx:1",
        "desiredCount": 1,
        "pendingCount": 0,
        "runningCount": 0,
        "failedTasks": 0,
        "createdAt": 1661210032.602,
        "updatedAt": 1661210032.602,
        "launchType": "FARGATE",
        "platformVersion": "1.4.0",
        "platformFamily": "Linux",
        "networkConfiguration": {
          "awsvpcConfiguration": {
            "assignPublicIp": "ENABLED",
            "securityGroups": [
              "sg-EXAMPLE"
            ],
            "subnets": [
              "subnet-EXAMPLEf",

```

```

        "subnet-EXAMPLE",
        "subnet-EXAMPLE"
    ]
},
"rolloutState": "IN_PROGRESS",
"rolloutStateReason": "ECS deployment ecs-
svc/3763308422771520962 in progress.",
"failedLaunchTaskCount": 0,
"replacedTaskCount": 0,
"serviceConnectConfiguration": {
    "enabled": true,
    "namespace": "service-connect",
    "services": [
        {
            "portName": "nginx",
            "clientAliases": [
                {
                    "port": 80
                }
            ]
        }
    ],
    "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
            "awslogs-group": "/ecs/service-connect-proxy",
            "awslogs-region": "us-west-2",
            "awslogs-stream-prefix": "service-connect-proxy"
        },
        "secretOptions": []
    }
},
"serviceConnectResources": [
    {
        "discoveryName": "nginx",
        "discoveryArn": "arn:aws:servicediscovery:us-
west-2:123456789012:service/srv-EXAMPLE"
    }
]
},
"roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",

```

```
    "version": 0,
    "events": [],
    "createdAt": 1661210032.602,
    "placementConstraints": [],
    "placementStrategy": [],
    "networkConfiguration": {
      "awsvpcConfiguration": {
        "assignPublicIp": "ENABLED",
        "securityGroups": [
          "sg-EXAMPLE"
        ],
        "subnets": [
          "subnet-EXAMPLE",
          "subnet-EXAMPLE",
          "subnet-EXAMPLE"
        ]
      }
    },
    "schedulingStrategy": "REPLICA",
    "enableECSManagedTags": true,
    "propagateTags": "SERVICE",
    "enableExecuteCommand": true
  }
}
```

La `serviceConnectConfiguration` fornita viene visualizzata all'interno della prima implementazione dell'output. Quando apporti modifiche al servizio ECS in modi che richiedono modifiche alle attività, Amazon ECS crea una nuova implementazione.

Fase 3: Verifica della riuscita della connessione

Per verificare che Service Connect sia configurato e funzionante, completa questa procedura per connetterti al servizio Web da un'applicazione esterna. Quindi, guarda le metriche aggiuntive nelle creazioni CloudWatch del proxy Service Connect.

Connessione al servizio Web da un'applicazione esterna

- Connessione all'indirizzo IP dell'attività e alla porta del container utilizzando l'indirizzo IP dell'attività

Usa il AWS CLI per ottenere l'ID dell'attività, utilizzando `aws ecs list-tasks --cluster tutorial`.

Se le sottoreti e il gruppo di sicurezza consentono il traffico proveniente dalla rete Internet pubblica sulla porta della definizione dell'attività, sarà possibile connettersi all'IP pubblico dal computer. Tuttavia, l'IP pubblico non è disponibile da `describe-tasks`, quindi i passaggi prevedono di andare su Amazon EC2 AWS Management Console o ottenere i dettagli dell'interfaccia AWS CLI di rete elastica.

In questo esempio, un' EC2 istanza Amazon nello stesso VPC utilizza l'IP privato dell'attività. L'applicazione è nginx, ma l'intestazione `server: envoy` mostra che viene utilizzato il proxy Service Connect. Il proxy Service Connect è in ascolto sulla porta del container a partire dalla definizione dell'attività.

```
$ curl -v 10.0.19.50:80/
* Trying 10.0.19.50:80...
* Connected to 10.0.19.50 (10.0.19.50) port 80 (#0)
> GET / HTTP/1.1
> Host: 10.0.19.50
> User-Agent: curl/7.79.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< server: envoy
< date: Tue, 23 Aug 2022 03:53:06 GMT
< content-type: text/html
< content-length: 612
< last-modified: Tue, 16 Apr 2019 13:08:19 GMT
< etag: "5cb5d3c3-264"
< accept-ranges: bytes
< x-envoy-upstream-service-time: 0
<
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
  body {
    width: 35em;
```



```
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Visualizzazione dei parametri di Service Connect

Il proxy Service Connect crea metriche dell'applicazione (connessione HTTP HTTP2, gRPC o TCP) nelle metriche. CloudWatch Quando usi la CloudWatch console, visualizza le dimensioni metriche aggiuntive di DiscoveryName, (, DiscoveryName ServiceName, ClusterName) e (TargetDiscoveryName, TargetDiscoveryName ServiceName, ClusterName) nello spazio dei nomi Amazon ECS. Per ulteriori informazioni su questi parametri e sulle dimensioni, consulta [Visualizza i parametri disponibili](#) nella Amazon CloudWatch Logs User Guide.

Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS

Il servizio Amazon ECS può facoltativamente essere configurato per l'uso della funzione di individuazione dei servizi di Amazon ECS. Service Discovery utilizza azioni AWS Cloud Map API per gestire i namespace HTTP e DNS per i tuoi servizi Amazon ECS. Per ulteriori informazioni, consulta [Che cos'è AWS Cloud Map?](#) nella Guida per gli sviluppatori di AWS Cloud Map .

L'individuazione dei servizi è disponibile nelle seguenti regioni: AWS

Nome della regione	Regione
US East (N. Virginia)	us-east-1
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti occidentali (California settentrionale)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Tokyo)	ap-northeast-1
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1

Nome della regione	Regione
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Stoccolma)	eu-north-1
Israele (Tel Aviv)	il-central-1
Europa (Spagna)	eu-south-2
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Medio Oriente (Bahrein)	me-south-1
Sud America (São Paulo)	sa-east-1
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1

Concetti relativi all'individuazione dei servizi

L'individuazione dei servizi è costituita dai componenti seguenti:

- **Service Discovery Namespace:** un gruppo logico di servizi di individuazione dei servizi che condividono lo stesso nome di dominio, ad esempio `example.com`, dove si desidera indirizzare il traffico. Puoi creare uno spazio dei nomi con una chiamata al `aws servicediscovery create-private-dns-namespace` comando o nella console Amazon ECS. Puoi utilizzare il comando `aws servicediscovery list-namespaces` per visualizzare le informazioni di riepilogo degli spazi dei nomi creati dall'account corrente. Per ulteriori informazioni sui comandi di scoperta dei servizi, consulta [create-private-dns-namespace](#) e [list-namespaces](#) nella Guida di riferimento AWS Cloud Map (service discovery) AWS CLI .

- Servizio di individuazione dei servizi: è incluso nello spazio dei nomi di individuazione dei servizi ed è costituito dal nome del servizio e dalla configurazione DNS per lo spazio dei nomi. Fornisce il componente principale seguente:
 - Registro dei servizi: consente di cercare un servizio tramite azioni DNS o AWS Cloud Map API e recuperare uno o più endpoint disponibili che possono essere utilizzati per connettersi al servizio.
- Istanza di individuazione dei servizi: esiste all'interno del servizio di individuazione dei servizi e include gli attributi associati a ogni servizio Amazon ECS nella directory del servizio.
- Attributi dell'istanza: i metadati seguenti vengono aggiunti come attributi personalizzati per ciascun servizio Amazon ECS configurato per l'utilizzo dell'individuazione dei servizi:
 - **AWS_INSTANCE_IPV4**— Per un A record, l'IPv4 indirizzo che Route 53 restituisce in risposta alle query DNS e AWS Cloud Map restituisce quando rileva i dettagli dell'istanza, ad esempio. 192.0.2.44
 - **AWS_INSTANCE_PORT**: il valore della porta associato al servizio di individuazione dei servizi.
 - **AVAILABILITY_ZONE**: la zona di disponibilità in cui il processo è stato avviato. Per le attività che utilizzano il tipo di EC2 avvio, questa è la zona di disponibilità in cui si trova l'istanza del contenitore. Per i processi che utilizzano il tipo di avvio Fargate, questa è la zona di disponibilità in cui è presente l'interfaccia di rete elastica.
 - **REGION**: la regione in cui è presente il processo.
 - **ECS_SERVICE_NAME**: il nome del servizio Amazon ECS a cui appartiene il processo.
 - **ECS_CLUSTER_NAME**: il nome del cluster Amazon ECS a cui appartiene il processo.
 - **EC2_INSTANCE_ID**: l'ID dell'istanza di container in cui è stata posizionato il processo. Questo attributo personalizzato non viene aggiunto se l'attività sta utilizzando il tipo di avvio Fargate.
 - **ECS_TASK_DEFINITION_FAMILY**: la famiglia della definizione di attività utilizzata dal processo.
 - **ECS_TASK_SET_EXTERNAL_ID**: se viene creato un set di processi per un'implementazione esterna ed è associato a un registro di individuazione dei servizi, l'attributo **ECS_TASK_SET_EXTERNAL_ID** includerà l'ID esterno del set di processi.
- Controlli dell'integrità di Amazon ECS: Amazon ECS esegue periodicamente controlli dell'integrità a livello di container. Se un endpoint non supera il controllo dello stato, viene rimosso dal routing DNS e contrassegnato come non integro.

Considerazioni relative all'individuazione dei servizi

Quando usi l'individuazione dei servizi, tieni presenti le considerazioni seguenti:

- L'individuazione dei servizi è supportata per i processi su Fargate che utilizzano la piattaforma versione 1.1.0 o successiva. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).
- I servizi configurati per l'utilizzo del rilevamento dei servizi Amazon ECS hanno un limite di 1.000 processi per servizio. Ciò è dovuto a una quota di servizio Route 53.
- Il flusso di lavoro Crea servizio nella console Amazon ECS supporta solo la registrazione dei servizi in spazi dei nomi DNS privati. Quando viene creato uno spazio dei nomi DNS AWS Cloud Map privato, verrà creata automaticamente una zona ospitata privata di Route 53.
- Per la corretta risoluzione DNS, gli attributi DNS del VPC devono essere configurati. Per ulteriori informazioni su come configurare gli attributi, consulta [Supporto DNS nel VPC](#) nella Guida per l'utente di Amazon VPC.
- I record DNS creati per un servizio DNS si registrano sempre con l'indirizzo IP privato per il processo piuttosto che l'indirizzo IP pubblico, anche quando vengono utilizzati gli spazi di nomi pubblici.
- L'individuazione dei servizi richiede che i processi specifichino la modalità di rete awsvpc, bridge o host (none non è supportata).
- Se la definizione di attività del servizio usa la modalità di rete awsvpc, puoi creare qualsiasi combinazione di record A o SRV per ciascuna attività di servizio. Se usi record SRV, è necessaria una porta.
- Se la definizione di attività di servizio usa la modalità di rete bridge o host, un record SRV è il solo tipo di record DNS supportato. Crea un record SRV per ciascuna operazione di servizio. Il record SRV deve specificare un nome di container e una combinazione di porte di container dalla definizione di attività.
- È possibile eseguire query sui record DNS per un servizio di individuazione dei servizi all'interno del VPC. Viene usato questo formato: <service-discovery-service-name>.<service-discovery-namespace>.
- Quando esegui una query DNS sul nome del servizio, i record A restituiscono un set di indirizzi IP corrispondenti alle attività. I record SRV restituiscono un set di indirizzi IP e le porte per ogni attività.
- Se dispone di otto o meno record integri, Route 53 risponde alle query DNS con tutti i record integri.
- Quando tutti i record non sono integri, Route 53 risponde alle query DNS con un massimo di otto record non integri.

- Puoi configurare il rilevamento dei servizi per un servizio protetto da un load balancer, ma il traffico del rilevamento dei servizi viene sempre instradato verso l'attività e non verso il load balancer.
- L'individuazione dei servizi non supporta l'utilizzo di Classic Load Balancer.
- Per il servizio di individuazione dei servizi consigliamo di utilizzare i controlli dell'integrità a livello dei container gestiti da Amazon ECS.
 - **HealthCheckCustomConfig**—Amazon ECS gestisce i controlli sanitari per tuo conto. Amazon ECS usa le informazioni ottenute dal container e dai controlli dell'integrità, insieme allo stato dell'attività, per aggiornare l'integrità con AWS Cloud Map. Questo comportamento viene specificato tramite il parametro `--health-check-custom-config` durante la creazione del servizio di individuazione dei servizi. Per ulteriori informazioni, consulta [HealthCheckCustomConfig](#) nella documentazione di riferimento dell'API AWS Cloud Map .
- Le AWS Cloud Map risorse create quando viene utilizzato il service discovery devono essere pulite manualmente.
- Le attività e le istanze vengono registrate UNHEALTHY fino a quando i controlli dello stato del contenitore non restituiscono un valore. Se i controlli sanitari hanno esito positivo, lo stato viene aggiornato a HEALTHY. Se i controlli di integrità del contenitore falliscono, l'istanza di rilevamento del servizio viene annullata.

Prezzi del servizio di individuazione dei servizi

I clienti che utilizzano l'individuazione dei servizi di Amazon ECS pagano per le risorse Route 53 e le operazioni API di individuazione AWS Cloud Map . Sono inclusi i costi per la creazione delle zone ospitate di Route 53 e per le query nel registro del servizio. Per ulteriori informazioni, consulta [Prezzi di AWS Cloud Map](#) nella Guida per gli sviluppatori di AWS Cloud Map .

Amazon ECS esegue controlli di integrità a livello di container e li espone a operazioni API di controllo dello stato AWS Cloud Map personalizzate. Questa opzione al momento viene fornita gratuitamente ai clienti. Se configuri controlli dell'integrità di rete aggiuntivi per le attività esposte pubblicamente, potrebbero esserti addebitati i relativi costi.

Creazione di un servizio Amazon ECS che utilizza Service Discovery

Scopri come creare un servizio contenente un'attività Fargate che utilizza il rilevamento dei servizi con. AWS CLI

Per un elenco della scoperta di Regioni AWS tale servizio di supporto, vedere [Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS](#).

Per informazioni sulle regioni che supportano Fargate, consulta [the section called “AWS Regioni di Fargate”](#).

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

Prerequisiti

Prima di iniziare questo tutorial, verifica che siano soddisfatti i seguenti requisiti preliminari:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).
- Le fasi descritte in [Configurazione per l'uso di Amazon ECS](#) sono complete.
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy [Amazon ECS_FullAccess](#) IAM.
- Hai creato almeno un VPC e un gruppo di sicurezza. Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).

Fase 1: Creare le risorse Service Discovery in AWS Cloud Map

Utilizza la procedura seguente per creare lo spazio dei nomi di individuazione dei servizi e il servizio di individuazione dei servizi.

1. Crea uno spazio dei nomi privato di individuazione dei servizi Cloud Map. In questo esempio viene creato uno spazio dei nomi denominato `tutorial`. Sostituisci `vpc-abcd1234` con l'ID di uno dei tuoi esistenti VPCs.

```
aws servicediscovery create-private-dns-namespace \  
  --name tutorial \  
  --vpc vpc-abcd1234
```

L'output di questo comando è il seguente:

```
{
```

```
{
  "OperationId": "h2qe3s6dxftvvt7riu6lfy2f6c3j1hf4-je6chs2e"
}
```

2. Usando `OperationId` dall'output della fase precedente, verifica che lo spazio dei nomi privato sia stato creato correttamente. Prendi nota dell'ID dello spazio dei nomi perché lo utilizzerai nei comandi successivi.

```
aws servicediscovery get-operation \
  --operation-id h2qe3s6dxftvvt7riu6lfy2f6c3j1hf4-je6chs2e
```

L'output è il seguente.

```
{
  "Operation": {
    "Id": "h2qe3s6dxftvvt7riu6lfy2f6c3j1hf4-je6chs2e",
    "Type": "CREATE_NAMESPACE",
    "Status": "SUCCESS",
    "CreateDate": 1519777852.502,
    "UpdateDate": 1519777856.086,
    "Targets": {
      "NAMESPACE": "ns-uejictsjen2i4eeg"
    }
  }
}
```

3. Usando l'ID `NAMESPACE` dall'output nella fase precedente, crea un servizio di individuazione dei servizi. Questo esempio crea un servizio denominato `myapplication`. Prendi nota dell'ID del servizio e dell'ARN perché li utilizzerai nei comandi successivi.

```
aws servicediscovery create-service \
  --name myapplication \
  --dns-config "NamespaceId="ns-uejictsjen2i4eeg",DnsRecords=[{Type="A",TTL="300"}]" \
  --health-check-custom-config FailureThreshold=1
```

L'output è il seguente.

```
{
  "Service": {
    "Id": "srv-utcrh6wavdkggqtk",

```



```

    "Arn": "arn:aws:servicediscovery:region:aws_account_id:service/srv-utcrh6wavdkggqtk",
    "Name": "myapplication",
    "DnsConfig": {
      "NamespaceId": "ns-uejictsjen2i4eeg",
      "DnsRecords": [
        {
          "Type": "A",
          "TTL": 300
        }
      ]
    },
    "HealthCheckCustomConfig": {
      "FailureThreshold": 1
    },
    "CreatorRequestId": "e49a8797-b735-481b-a657-b74d1d6734eb"
  }
}

```

Fase 2: Creazione delle risorse Amazon ECS

Utilizza la procedura seguente per creare il cluster Amazon ECS, la definizione di attività e il servizio.

1. Crea un cluster Amazon ECS. In questo esempio viene creato un cluster denominato `tutorial`.

```

aws ecs create-cluster \
  --cluster-name tutorial

```

2. Registra una definizione di attività che sia compatibile con Fargate e utilizzi la modalità di rete `awsvpc`. Completare la procedura riportata di seguito.
 - a. Crea un file denominato `fargate-task.json` con il contenuto della seguente definizione di attività.

```

{
  "family": "tutorial-task-def",
  "networkMode": "awsvpc",
  "containerDefinitions": [
    {
      "name": "sample-app",
      "image": "public.ecr.aws/docker/library/httpd:2.4",
      "portMappings": [

```

```

        {
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
        }
    ],
    "essential": true,
    "entryPoint": [
        "sh",
        "-c"
    ],
    "command": [
        "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
App</title> <style>body {margin-top: 40px; background-color: #333;} </style>
</head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample
App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a
container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/
htdocs/index.html && httpd-foreground\""
    ]
    },
    "requiresCompatibilities": [
        "FARGATE"
    ],
    "cpu": "256",
    "memory": "512"
}

```

- b. Registrazione della definizione dell'attività tramite `fargate-task.json`.

```

aws ecs register-task-definition \
    --cli-input-json file://fargate-task.json

```

3. Crea un servizio ECS completando la seguente procedura:

- a. Creazione di un file denominato `ecs-service-discovery.json` con il contenuto del servizio ECS che stai per creare. Questo esempio usa la definizione di attività creata nella fase precedente. È obbligatorio l'uso di un parametro `awsvpcConfiguration` perché la definizione di attività di esempio usa la modalità di rete `awsvpc`.

Quando crei il servizio ECS, specifica il tipo di avvio Fargate e la versione della piattaforma LATEST che supporta l'individuazione dei servizi. Quando viene creato il servizio di individuazione dei servizi in AWS Cloud Map, l'ARN restituito è `registryArn`.

`securityGroups` e `subnets` devono appartenere al VPC utilizzato per creare lo spazio dei nomi Cloud Map. Puoi ottenere il gruppo di sicurezza e la sottorete IDs dalla console Amazon VPC.

```
{
  "cluster": "tutorial",
  "serviceName": "ecs-service-discovery",
  "taskDefinition": "tutorial-task-def",
  "serviceRegistries": [
    {
      "registryArn":
"arn:aws:servicediscovery:region:aws_account_id:service/srv-utcrh6wavdkggqtk"
    }
  ],
  "launchType": "FARGATE",
  "platformVersion": "LATEST",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "assignPublicIp": "ENABLED",
      "securityGroups": [ "sg-abcd1234" ],
      "subnets": [ "subnet-abcd1234" ]
    }
  },
  "desiredCount": 1
}
```

- b. Crea il tuo servizio ECS utilizzando `ecs-service-discovery.json`.

```
aws ecs create-service \
  --cli-input-json file://ecs-service-discovery.json
```

Fase 3: Verifica Service Discovery in AWS Cloud Map

Puoi verificare che sia stato creato tutto correttamente eseguendo una query sulle informazioni di individuazione dei servizi. Dopo aver configurato l'individuazione del servizio, puoi utilizzare le operazioni AWS Cloud Map API o effettuare una chiamata `dig` da un'istanza all'interno del tuo VPC. Completare la procedura riportata di seguito.

1. Usando l'ID servizio di individuazione dei servizi, elenca le istanze di individuazione dei servizi. Prendi nota dell'ID istanza (contrassegnato in grassetto) per la pulizia delle risorse.

```
aws servicediscovery list-instances \  
  --service-id srv-utcrh6wavdkggqtk
```

L'output è il seguente.

```
{  
  "Instances": [  
    {  
      "Id": "16becc26-8558-4af1-9fbd-f81be062a266",  
      "Attributes": {  
        "AWS_INSTANCE_IPV4": "172.31.87.2"  
        "AWS_INSTANCE_PORT": "80",  
        "AVAILABILITY_ZONE": "us-east-1a",  
        "REGION": "us-east-1",  
        "ECS_SERVICE_NAME": "ecs-service-discovery",  
        "ECS_CLUSTER_NAME": "tutorial",  
        "ECS_TASK_DEFINITION_FAMILY": "tutorial-task-def"  
      }  
    }  
  ]  
}
```

2. Utilizza lo spazio dei nomi, il servizio di individuazione dei servizi e parametri aggiuntivi quali il nome del cluster ECS per ottenere i dettagli sulle istanze di individuazione dei servizi.

```
aws servicediscovery discover-instances \  
  --namespace-name tutorial \  
  --service-name myapplication \  
  --query-parameters ECS_CLUSTER_NAME=tutorial
```

3. Puoi eseguire query sui record DNS creati nella zona ospitata di Route 53 per il servizio di individuazione dei servizi con i seguenti comandi AWS CLI :
 - a. Con l'ID spazio dei nomi, ottieni le informazioni sullo spazio dei nomi che includono l'ID della zona ospitata di Route 53.

```
aws servicediscovery \  
  get-namespace --id ns-uejictsjen2i4eeg
```

L'output è il seguente.

```
{
  "Namespace": {
    "Id": "ns-uejictsjen2i4eeg",
    "Arn": "arn:aws:servicediscovery:region:aws_account_id:namespace/ns-uejictsjen2i4eeg",
    "Name": "tutorial",
    "Type": "DNS_PRIVATE",
    "Properties": {
      "DnsProperties": {
        "HostedZoneId": "Z35JQ4ZFDYPLV"
      }
    },
    "CreateDate": 1519777852.502,
    "CreatorRequestId": "9049a1d5-25e4-4115-8625-96dbda9a6093"
  }
}
```

- b. Utilizzando l'ID della zona ospitata di Route 53 dalla fase precedente (visualizza il testo in grassetto), ottieni il set di record delle risorse per la zona ospitata.

```
aws route53 list-resource-record-sets \
  --hosted-zone-id Z35JQ4ZFDYPLV
```

4. Puoi eseguire query sul DNS anche da un'istanza all'interno tramite dig.

```
dig +short myapplication.tutorial
```

Fase 4: pulizia

Una volta terminato questo tutorial, elimina le risorse associate in modo da evitare costi aggiuntivi per le risorse non utilizzate. Completare la procedura riportata di seguito.

1. Annulla la registrazione delle istanze del servizio di individuazione dei servizi utilizzando l'ID servizio e l'ID istanza annotati in precedenza.

```
aws servicediscovery deregister-instance \
  --service-id srv-utcrh6wavdkggqtk \
  --instance-id 16becc26-8558-4af1-9fbd-f81be062a266
```

L'output è il seguente.

```
{
  "OperationId": "xhu73bsertlyffhm3faq7kumsmx274n-jh0zimzv"
}
```

- Utilizzando l'OperationId dall'output della fase precedente, verifica che la registrazione delle istanze del servizio di individuazione dei servizi sia stata annullata correttamente.

```
aws servicediscovery get-operation \
  --operation-id xhu73bsertlyffhm3faq7kumsmx274n-jh0zimzv
```

```
{
  "Operation": {
    "Id": "xhu73bsertlyffhm3faq7kumsmx274n-jh0zimzv",
    "Type": "DEREGISTER_INSTANCE",
    "Status": "SUCCESS",
    "CreateDate": 1525984073.707,
    "UpdateDate": 1525984076.426,
    "Targets": {
      "INSTANCE": "16becc26-8558-4af1-9fbd-f81be062a266",
      "ROUTE_53_CHANGE_ID": "C5NSRG1J4I1FH",
      "SERVICE": "srv-utcrh6wavdkggqtk"
    }
  }
}
```

- Eliminazione del servizio di individuazione dei servizi utilizzando l'ID servizio.

```
aws servicediscovery delete-service \
  --id srv-utcrh6wavdkggqtk
```

- Eliminazione dello spazio dei nomi di individuazione dei servizi utilizzando l'ID spazio dei nomi.

```
aws servicediscovery delete-namespace \
  --id ns-uejictsjen2i4eeg
```

L'output è il seguente.

```
{
  "OperationId": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj"
}
```

5. Usando l'OperationId dall'output della fase precedente, verifica che lo spazio dei nomi di individuazione dei servizi sia stato eliminato correttamente.

```
aws servicediscovery get-operation \  
  --operation-id c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj
```

L'output è il seguente.

```
{  
  "Operation": {  
    "Id": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj",  
    "Type": "DELETE_NAMESPACE",  
    "Status": "SUCCESS",  
    "CreateDate": 1525984602.211,  
    "UpdateDate": 1525984602.558,  
    "Targets": {  
      "NAMESPACE": "ns-rymlehshst7hhukh",  
      "ROUTE_53_CHANGE_ID": "CJP2A2M86XW30"  
    }  
  }  
}
```

6. Aggiorna il conteggio desiderato per il servizio Amazon ECS a 0. Questa operazione deve essere eseguita per eliminare il servizio nella fase successiva.

```
aws ecs update-service \  
  --cluster tutorial \  
  --service ecs-service-discovery \  
  --desired-count 0
```

7. Elimina il servizio Amazon ECS.

```
aws ecs delete-service \  
  --cluster tutorial \  
  --service ecs-service-discovery
```

8. Elimina il cluster Amazon ECS.

```
aws ecs delete-cluster \  
  --cluster tutorial
```

Usa Amazon VPC Lattice per connettere, osservare e proteggere i tuoi servizi Amazon ECS

Amazon VPC Lattice è un servizio di rete di applicazioni gestito che i clienti di Amazon ECS utilizzano per osservare, proteggere e monitorare le applicazioni create su servizi di AWS elaborazione e account senza dover modificare il codice. VPCs

VPC Lattice utilizza gruppi target, che sono una raccolta di risorse di calcolo. Questi obiettivi eseguono la tua applicazione o il tuo servizio e possono essere EC2 istanze Amazon, indirizzi IP, funzioni Lambda e Application Load Balancer. Associando i propri servizi Amazon ECS a un gruppo target VPC Lattice, i clienti possono ora abilitare le attività di Amazon ECS come destinazioni IP in VPC Lattice. Amazon ECS registra automaticamente le attività nel gruppo target VPC Lattice quando vengono lanciate le attività per il servizio registrato.

Note

Quando si utilizzano cinque configurazioni VPC Lattice, il tempo di implementazione potrebbe essere leggermente più lungo rispetto a quando si utilizzano meno configurazioni.

Una regola listener viene utilizzata per inoltrare il traffico a un gruppo target specifico quando le condizioni sono soddisfatte. Un listener verifica le richieste di connessione utilizzando il protocollo sulla porta configurata. Un servizio indirizza le richieste verso le destinazioni registrate in base alle regole definite al momento della configurazione del listener.

Amazon ECS sostituisce automaticamente un'attività anche se non funziona correttamente in base ai controlli di integrità di VPC Lattice. Una volta associati a VPC Lattice, i clienti Amazon ECS possono anche sfruttare molte altre funzionalità di connettività, sicurezza e osservabilità cross-computing di VPC Lattice, come la connessione a servizi tra cluster e account con integrazione IAM per l'autorizzazione e l' VPCsautenticazione e funzionalità avanzate di AWS Resource Access Manager gestione del traffico.

I clienti Amazon ECS possono trarre vantaggio da VPC Lattice nei seguenti modi.

- Maggiore produttività degli sviluppatori - VPC Lattice aumenta la produttività degli sviluppatori consentendoti di concentrarti sulla creazione di funzionalità, mentre VPC Lattice gestisce le sfide di rete, sicurezza e osservabilità in modo uniforme su tutte le piattaforme di elaborazione.
- Migliore livello di sicurezza - VPC Lattice consente agli sviluppatori di autenticare e proteggere facilmente la comunicazione tra applicazioni e piattaforme di calcolo, applicare la crittografia in

transito e applicare controlli di accesso granulari con le politiche di autenticazione VPC Lattice. Ciò consente di adottare una posizione di sicurezza più solida che soddisfi i principali requisiti normativi e di conformità del settore.

- **Migliore scalabilità e resilienza delle applicazioni** - VPC Lattice consente di creare una rete di applicazioni distribuite con funzionalità come routing, autenticazione, autorizzazione e monitoraggio basati su percorsi, intestazioni e metodi. Questi vantaggi vengono forniti senza sovraccarico di risorse sui carichi di lavoro e possono supportare implementazioni multi-cluster che generano milioni di richieste al secondo senza aggiungere latenza significativa.
- **Flessibilità di implementazione con un'infrastruttura eterogenea** - VPC Lattice offre funzionalità coerenti in tutti i servizi di elaborazione come Amazon ECS, Fargate, Amazon, EC2 Amazon EKS e Lambda e consente all'organizzazione la flessibilità di scegliere l'infrastruttura adatta per ogni applicazione.

Come funziona VPC Lattice con altri servizi Amazon ECS

L'uso di VPC Lattice con Amazon ECS può cambiare il modo in cui utilizzi altri servizi Amazon ECS, mentre altri rimangono invariati.

Application Load Balancer

Non è più necessario creare uno specifico Application Load Balancer da utilizzare con il tipo di gruppo target Application Load Balancer in VPC Lattice che poi si collega al servizio Amazon ECS. In alternativa, devi solo configurare il tuo servizio Amazon ECS con un gruppo target VPC Lattice. Puoi anche scegliere di utilizzare Application Load Balancer con Amazon ECS contemporaneamente.

Implementazioni in sequenza di Amazon ECS

Solo le distribuzioni in sequenza di Amazon ECS funzionano con VPC Lattice e Amazon ECS inserisce le attività in modo sicuro e le rimuove dai servizi durante la distribuzione. La distribuzione del codice e le distribuzioni Blue/Green non sono supportate.

Per ulteriori informazioni su VPC Lattice, consulta la Amazon [VPC](#) Lattice User Guide.

Crea un servizio che utilizza VPC Lattice

È possibile utilizzare AWS Management Console o il AWS CLI per creare un servizio con VPC Lattice.

Prerequisiti

Prima di iniziare questo tutorial, verifica che siano soddisfatti i seguenti requisiti preliminari:

- La versione più recente di AWS CLI è installata e configurata. Per ulteriori informazioni, consulta l'argomento relativo all'[installazione di AWS Command Line Interface](#).

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

- Le fasi descritte in [Configurazione per l'uso di Amazon ECS](#) sono complete.
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di policy IAM. [Amazon ECS_FullAccess](#)

Crea un servizio che utilizza VPC Lattice con AWS Management Console

Segui questi passaggi per creare un servizio con VPC Lattice utilizzando AWS Management Console

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina di navigazione, scegli Cluster.
3. Nella pagina Cluster, scegli il cluster in cui creare il servizio.
4. Nella scheda Services (Servizi), scegli Create (Crea).

Se non hai mai creato un servizio prima, segui i passaggi riportati in [Creazione di un servizio Amazon ECS utilizzando la console](#), quindi continua con questi passaggi quando raggiungi la sezione VPC Lattice.

5. Scegli di attivare VPC Lattice selezionando il pulsante.
6. Per utilizzare un ruolo esistente, per il ruolo di infrastruttura ECS per Amazon ECS, scegline uno che hai già creato da utilizzare durante la creazione del gruppo target VPC Lattice. Per creare un nuovo ruolo, crea un ruolo di infrastruttura ECS.
7. Scegli il VPC.

Il VPC dipende dalla modalità di rete selezionata al momento della registrazione della definizione dell'attività. Se utilizzi la network modalità host o con il tipo di EC2 avvio, scegli il tuo VPC.

Per la `awsipc` modalità, il VPC viene selezionato automaticamente in base al VPC scelto in Rete e non può essere modificato.

8. In Target Groups scegli il gruppo o i gruppi target. Devi scegliere almeno un gruppo target e puoi averne al massimo cinque. Scegli Aggiungi gruppo target per aggiungere altri gruppi target. Scegli il nome della porta, il protocollo e la porta per ogni gruppo target che hai scelto. Per eliminare un gruppo target, scegli Rimuovi.

Note

- Se desideri aggiungere gruppi target esistenti, devi utilizzare il AWS CLI. Per istruzioni su come aggiungere gruppi target utilizzando il AWS CLI, consulta [register-targets](#) nel riferimento. AWS Command Line Interface
- Sebbene un servizio VPC Lattice possa avere più gruppi target, ogni gruppo target può essere aggiunto a un solo servizio.

9. A questo punto, accedi alla console VPC Lattice per continuare la configurazione. È qui che includi i tuoi nuovi gruppi target nell'azione predefinita del listener o nelle regole di un servizio VPC Lattice esistente.

Per ulteriori informazioni, consulta le [regole del listener per il servizio VPC Lattice](#).

Important

È necessario consentire il `vpc-lattice` prefisso della regola in entrata al gruppo o alle attività di sicurezza e i controlli di integrità potrebbero avere esito negativo.

Crea un servizio che utilizza VPC Lattice con AWS CLI

Usa AWS CLI per creare un servizio con VPC Lattice. Sostituisci ogni *user input placeholder* con le tue informazioni.

1. Crea un file di configurazione del gruppo target. L'esempio seguente è denominato `tg-config.json`

```
{  
  "ipAddressType": "IPv4",
```

```
"port": 443,  
"protocol": "HTTPS",  
"protocolVersion": "HTTP1",  
"vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
}
```

2. Usa il seguente comando per creare un gruppo target VPC Lattice.

```
aws vpc-lattice create-target-group \  
  --name my-lattice-target-group-ip \  
  --type IP \  
  --config file://tg-config.json
```

Output di esempio:

```
{  
  "arn": "arn:aws:vpc-lattice:us-east-2:123456789012:targetgroup/  
tg-0eaa4b9ab4EXAMPLE",  
  "config": {  
    "healthCheck": {  
      "enabled": true,  
      "healthCheckIntervalSeconds": 30,  
      "healthCheckTimeoutSeconds": 5,  
      "healthyThresholdCount": 5,  
      "matcher": {  
        "httpCode": "200"  
      },  
      "path": "/",  
      "protocol": "HTTPS",  
      "protocolVersion": "HTTP1",  
      "unhealthyThresholdCount": 2  
    },  
    "ipAddressType": "IPV4",  
    "port": 443,  
    "protocol": "HTTPS",  
    "protocolVersion": "HTTP1",  
    "vpcIdentifier": "vpc-f1663d9868EXAMPLE"  
  },  
  "id": "tg-0eaa4b9ab4EXAMPLE",  
  "name": "my-lattice-target-group-ip",  
  "status": "CREATE_IN_PROGRESS",  
  "type": "IP"  
}
```

```
}
```

3. Il seguente file json denominato *ecs-service-vpc-lattice.json* è un esempio utilizzato per collegare un servizio Amazon ECS a un gruppo target VPC Lattice. L'portNameesempio seguente è lo stesso che hai definito nella definizione dell'attività.

```
{
  "serviceName": "ecs-service-vpc-lattice",
  "taskDefinition": "ecs-task-def",
  "vpcLatticeConfigurations": [
    {
      "targetGroupArn": "arn:aws:vpc-lattice:us-
west-2:123456789012:targetgroup/tg-0eaa4b9ab4EXAMPLE",
      "portName": "testvpclattice",
      "roleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRoleVpcLattice"
    }
  ],
  "desiredCount": 5,
  "role": "ecsServiceRole"
}
```

Utilizza il seguente comando per creare un servizio Amazon ECS e collegarlo al gruppo target VPC Lattice utilizzando l'esempio json sopra riportato.

```
aws ecs create-service \
  --cluster clusterName \
  --serviceName ecs-service-vpc-lattice \
  --cli-input-json file://ecs-service-vpc-lattice.json
```

 Note

VPC Lattice non è supportato su Amazon ECS Anywhere.

Proteggi le tue attività di Amazon ECS dall'interruzione a causa di eventi di scalabilità

Puoi utilizzare la protezione scale-in delle attività di Amazon ECS per evitare che le tue attività vengano interrotte da eventi di scalabilità in base all'auto scalabilità del servizio o alle distribuzioni.

Alcune applicazioni richiedono un meccanismo per salvaguardare le attività strategiche dall'interruzione dovuta a eventi scalabili nei periodi di scarso utilizzo o durante le implementazioni dei servizi. Per esempio:

- Si tratta di un'applicazione asincrona di elaborazione delle code, ad esempio un processo di transcodifica video in cui alcune attività devono essere eseguite per ore anche quando l'utilizzo cumulativo del servizio è basso.
- Hai un'applicazione di gioco che esegue server di gioco come attività Amazon ECS che devono continuare a funzionare anche se tutti gli utenti si sono disconnessi per ridurre la latenza di avvio di un riavvio del server.
- Quando si implementa una nuova versione del codice, è necessario che le attività continuino a essere eseguite perché sarebbe costoso rielaborarle.

Per evitare che le attività appartenenti al servizio vengano terminate in caso di un evento di dimensionamento, imposta l'attributo `ProtectionEnabled` su `true`. Se l'impostazione è impostata su `true`, `ProtectionEnabled` le attività sono protette per 2 ore per impostazione predefinita. È quindi possibile personalizzare il periodo di protezione utilizzando l'`ExpiresInMinutes` attributo. Puoi proteggere le tue attività per un minimo di 1 minuto e fino a un massimo di 2.880 minuti (48 ore). Se utilizzi il AWS CLI, puoi specificare l'`--protection-enabled` opzione.

Dopo che un'attività ha terminato il lavoro richiesto, sarà possibile impostare l'attributo `ProtectionEnabled` su `false`, in modo che l'attività venga terminata da successivi eventi di scalabilità. Se si utilizza il AWS CLI, è possibile specificare l'`--no-protection-enabled` opzione.

Meccanismi di protezione scalabile delle attività

Puoi impostare e ottenere una protezione scalabile delle attività utilizzando l'endpoint dell'agente container Amazon ECS o l'API Amazon ECS.

- Endpoint dell'agente del container Amazon ECS

Consigliamo di utilizzare l'endpoint dell'agente container Amazon ECS per attività che possono determinare autonomamente la necessità di essere protette. Utilizza questo approccio per carichi di lavoro basati su code o di elaborazione dei processi

Quando un container inizia il lavoro di elaborazione, ad esempio consumando un messaggio SQS, è possibile impostare l'attributo `ProtectionEnabled` tramite il percorso dell'endpoint di protezione scalabile dell'attività `$ECS_AGENT_URI/task-protection/v1/state` dall'interno del container. Amazon ECS non terminerà questa attività durante gli eventi di scalabilità. Una volta che l'attività ha terminato il suo lavoro, puoi cancellare l'`ProtectionEnabled` attributo utilizzando lo stesso endpoint, in modo che l'attività possa essere interrotta durante i successivi eventi scalabili.

Per ulteriori informazioni sull'endpoint Amazon ECS Container Agent, consulta [Endpoint di protezione scalabile per attività Amazon ECS](#)

- API Amazon ECS

Puoi utilizzare l'API Amazon ECS per impostare e recuperare la protezione scalabile delle attività se l'applicazione dispone di un componente che monitora lo stato delle attività attive. Utilizza `UpdateTaskProtection` per contrassegnare una o più attività come protette. Utilizza per recuperare `GetTaskProtection` lo stato di protezione.

Un esempio di questo approccio potrebbe essere se l'applicazione ospita sessioni del server di gioco come attività Amazon ECS. Quando un utente accede a una sessione sul server (attività), è possibile contrassegnare l'attività come protetta. Dopo la disconnessione dell'utente, puoi annullare la protezione specifica per l'attività o annullare periodicamente la protezione per attività simili che non hanno più sessioni attive, a seconda se è necessario mantenere i server inattivi.

Per ulteriori informazioni, consulta [UpdateTaskProtection](#) e consulta il riferimento [GetTaskProtection](#) all'API di Amazon Elastic Container Service.

È possibile combinare entrambi gli approcci. Ad esempio, utilizza l'endpoint dell'agente Amazon ECS per impostare la protezione delle attività dall'interno di un container e utilizza l'API Amazon ECS per rimuovere la protezione delle attività dal servizio di controller esterno.

Considerazioni

Considera i seguenti punti prima di utilizzare la protezione scalabile delle attività:

- Ti consigliamo di utilizzare l'endpoint dell'agente del container Amazon ECS, in quanto l'agente Amazon ECS dispone di meccanismi di ripetizione integrati e di un'interfaccia più semplice.
- Puoi reimpostare il periodo di scadenza della protezione per il dimensionamento delle attività richiamando `UpdateTaskProtection` su un'attività che ha già la protezione attivata.
- Determina il tempo necessario a un'attività per completare il lavoro richiesto e imposta la proprietà `expiresInMinutes` di conseguenza. Se imposti la scadenza della protezione più a lungo del necessario, dovrai sostenere costi e ritardi nell'implementazione di nuove attività.
- La protezione per il dimensionamento delle attività è supportata sull'agente del container Amazon ECS 1.65.0 o versioni successive.

Puoi aggiungere il supporto per questa funzionalità sulle EC2 istanze Amazon che utilizzano versioni precedenti dell'agente container Amazon ECS aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

- Considerazioni sull'implementazione:
 - Se il servizio utilizza un aggiornamento in sequenza, verranno create nuove attività ma le attività che eseguono una versione precedente non verranno interrotte fino a quando `protectionEnabled` non viene annullato o scade. È possibile regolare il parametro `maximumPercentage` nella configurazione di implementazione su un valore che consenta di creare nuove attività quando le vecchie attività sono protette.
 - Se viene applicato un aggiornamento blu/verde, l'implementazione blu contenente le attività protette non verrà rimossa se le attività presentano l'opzione `protectionEnabled`. Il traffico verrà dirottato verso le nuove attività che si presentano e le attività precedenti verranno rimosse solo quando `protectionEnabled` vengono cancellate o scadute. A seconda del timeout degli CloudFormation aggiornamenti, la CodeDeploy distribuzione potrebbe scadere e le attività Blue precedenti potrebbero essere ancora presenti.
 - Se lo utilizzi CloudFormation, lo update-stack ha un timeout di 3 ore. Pertanto, se imposti la protezione delle attività per più di 3 ore, la CloudFormation distribuzione potrebbe causare errori e ripristini.

Durante il periodo in cui le vecchie attività sono protette, lo CloudFormation stack viene visualizzato. `UPDATE_IN_PROGRESS` Se la protezione per il dimensionamento delle attività viene rimossa o scade entro la finestra di 3 ore, l'implementazione avrà esito positivo e passerà allo stato `UPDATE_COMPLETE`. Se l'implementazione rimane bloccata in `UPDATE_IN_PROGRESS` per più di 3 ore, non riuscirà e mostrerà lo stato `UPDATE_FAILED`, quindi verrà ripristinato il vecchio set di attività.

- Amazon ECS invia gli eventi di servizio se le attività protette impediscono a un'implementazione (in sequenza o blu/verde) di raggiungere lo stato stazionario, in modo che sia possibile intraprendere azioni correttive. Durante il tentativo di aggiornare lo stato di protezione di un'attività, se viene ricevuto un messaggio di errore `DEPLOYMENT_BLOCKED`, allora significa che il servizio ha più attività protette rispetto al numero di attività desiderato per il servizio. Per risolvere questo errore, effettuare una delle seguenti operazioni:
 - Attendi la scadenza della protezione dell'attività corrente. Quindi imposta la protezione delle attività.
 - Determina quali attività possono essere arrestate. Quindi, usa `UpdateTaskProtection` con l'opzione `protectionEnabled` impostata su `false` per queste attività.
 - Aumenta il numero di attività desiderate del servizio portandolo a un numero maggiore del numero di attività protette.

Autorizzazioni IAM richieste per la protezione scalabile delle attività

L'attività deve avere il ruolo di attività di Amazon ECS con le seguenti autorizzazioni:

- `ecs:GetTaskProtection`: consente all'agente del container Amazon ECS di effettuare chiamate `GetTaskProtection`.
- `ecs:UpdateTaskProtection`: consente all'agente del container Amazon ECS di effettuare chiamate `UpdateTaskProtection`.

Endpoint di protezione scalabile per attività Amazon ECS

L'agente del container Amazon ECS inserisce automaticamente la variabile di ambiente `ECS_AGENT_URI` nei container delle attività di Amazon ECS per fornire un metodo per interagire con l'endpoint API dell'agente del container.

Consigliamo di utilizzare l'endpoint dell'agente container Amazon ECS per attività che possono determinare autonomamente la necessità di essere protette.

Quando un contenitore inizia a elaborare il lavoro, è possibile impostare l'`protectionEnabled` attributo utilizzando il percorso `$ECS_AGENT_URI/task-protection/v1/state` dell'endpoint di protezione scale-in dell'attività dall'interno del contenitore.

Utilizzate una richiesta PUT a questo URI dall'interno di un contenitore per impostare la protezione scalabile delle attività. Una richiesta GET a questo URI restituisce lo stato di protezione corrente di un'attività.

Parametri della richiesta di protezione scalabile in base alla dimensione dell'attività

È possibile impostare la protezione scalabile delle attività utilizzando l'endpoint `${ECS_AGENT_URI}/task-protection/v1/state` con i seguenti parametri di richiesta.

ProtectionEnabled

`true` Specificare di contrassegnare un'attività per la protezione. `false` Specificare se rimuovere la protezione e rendere l'attività idonea alla cessazione.

Tipo: Booleano

Campo obbligatorio: sì

ExpiresInMinutes

Il numero di minuti in cui l'attività è protetta. È possibile specificare da un minimo di 1 minuto a un massimo di 2.880 minuti (48 ore). Durante questo periodo di tempo, l'attività non verrà terminata da eventi di dimensionamento derivanti dal servizio Dimensionamento automatico o dalle implementazioni. Trascorso questo periodo di tempo, il parametro `protectionEnabled` viene impostato su `false`.

Se non si specifica l'ora, l'attività viene protetta automaticamente per 120 minuti (2 ore).

Tipo: integer

Campo obbligatorio: no

Gli esempi seguenti mostrano come impostare la protezione delle attività con durate diverse.

Esempio di come proteggere un'attività con il periodo di tempo predefinito

Questo esempio mostra come proteggere un'attività con il periodo di tempo predefinito di 2 ore.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-protection/v1/state --data '{"ProtectionEnabled":true}'
```

Esempio di protezione di un'attività per 60 minuti

Questo esempio mostra come proteggere un'attività per 60 minuti utilizzando il parametro `expiresInMinutes`.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-protection/v1/state --data '{"ProtectionEnabled":true,"ExpiresInMinutes":60}'
```

Esempio di protezione di un'attività per 24 ore

Questo esempio mostra come proteggere un'attività per 24 ore utilizzando il parametro `expiresInMinutes`.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-protection/v1/state --data '{"ProtectionEnabled":true,"ExpiresInMinutes":1440}'
```

La richiesta PUT restituisce la seguente risposta.

```
{
  "protection": {
    "ExpirationDate": "2023-12-20T21:57:44.837Z",
    "ProtectionEnabled": true,
    "TaskArn": "arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"
  }
}
```

Parametri di risposta alla protezione scalabili in base alle attività

La risposta in formato JSON dell'endpoint di protezione per il dimensionamento delle attività `${ECS_AGENT_URI}/task-protection/v1/state` restituisce le seguenti informazioni.

ExpirationDate

L'ora in cui scadrà la protezione per l'attività. Se l'attività non è protetta, questo valore è nullo.

ProtectionEnabled

Lo stato di protezione dell'attività. Se la protezione scalabile è abilitata per un'attività, il valore è `true`. In caso contrario è `false`.

TaskArn

L'Amazon Resource Name (ARN) completo dell'attività a cui appartiene il container.

L'esempio che segue mostra i dettagli restituiti per un'attività protetta.

```
curl --request GET ${ECS_AGENT_URI}/task-protection/v1/state
```

```
{
  "protection":{
    "ExpirationDate":"2023-12-20T21:57:44Z",
    "ProtectionEnabled":true,
    "TaskArn":"arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"
  }
}
```

Le seguenti informazioni vengono restituite quando si verifica un errore.

Arn

Il nome della risorsa Amazon (ARN) completo dell'attività.

Detail

I dettagli relativi all'errore.

Reason

Il motivo dell'errore.

L'esempio che segue mostra i dettagli restituiti per un'attività che non è protetta.

```
{
  "failure":{
    "Arn":"arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0",
    "Detail":null,
    "Reason":"TASK_NOT_VALID"
  }
}
```

Le seguenti informazioni vengono restituite quando si verifica un'eccezione.

requestID

L'ID della AWS richiesta per la chiamata API Amazon ECS che genera un'eccezione.

Arn

Il nome della risorsa Amazon (ARN) completo dell'attività o del servizio.

Code

Il codice di errore.

Message

Messaggio di errore.

Note

Se viene visualizzato un errore `RequestError` o `RequestTimeout`, è probabile che si tratti di un problema di rete. Prova a utilizzare gli endpoint VPC per Amazon ECS.

L'esempio che segue mostra i dettagli restituiti quando si verifica un errore.

```
{
  "requestID": "12345-abc-6789-0123-abc",
  "error": {
    "Arn": "arn:aws:ecs:us-west-2:555555555555:task/my-cluster-name/1234567890abcdef0",
    "Code": "AccessDeniedException",
    "Message": "User: arn:aws:sts::444455556666:assumed-role/my-ecs-task-role/1234567890abcdef0 is not authorized to perform: ecs:GetTaskProtection on resource: arn:aws:ecs:us-west-2:555555555555:task/test/1234567890abcdef0 because no identity-based policy allows the ecs:GetTaskProtection action"
  }
}
```

L'errore seguente viene visualizzato se l'agente Amazon ECS non è in grado di ottenere una risposta dall'endpoint Amazon ECS per motivi quali problemi di rete o il piano di controllo Amazon ECS inattivo.

```
{
  "error": {
    "Arn": "arn:aws:ecs:us-west-2:555555555555:task/my-cluster-name/1234567890abcdef0",
    "Code": "RequestCanceled",
    "Message": "Timed out calling Amazon ECS Task Protection API"
  }
}
```

```
}
```

Il seguente errore si verifica quando l'agente Amazon ECS riceve un'eccezione di limitazione da Amazon ECS.

```
{
  "requestID": "12345-abc-6789-0123-abc",
  "error": {
    "Arn": "arn:aws:ecs:us-west-2:555555555555:task/my-cluster-name/1234567890abcdef0",
    "Code": "ThrottlingException",
    "Message": "Rate exceeded"
  }
}
```

Usa l'iniezione di errori con i tuoi carichi di lavoro Amazon ECS e Fargate

I clienti possono utilizzare l'iniezione di errori con Amazon ECS sia su Amazon che su EC2 Fargate per testare la risposta della loro applicazione a determinati scenari di compromissione. Questi test forniscono informazioni che puoi utilizzare per ottimizzare le prestazioni e la resilienza dell'applicazione.

Quando l'iniezione dei guasti è abilitata, l'agente container Amazon ECS consente alle attività di accedere a nuovi endpoint di iniezione degli errori. È necessario attivare il consenso per utilizzare l'iniezione di errore impostando il valore del parametro di definizione dell'`enableFaultInjection` attività su `true`. Il valore predefinito è `false`.

```
{
  ...
  "enableFaultInjection": true
}
```

Note

L'iniezione dei guasti funziona solo con le attività che utilizzano le modalità `host` di rete `awsvpc` o.

L'iniezione di errori non è disponibile in Windows.

Per informazioni su come abilitare l'iniezione di errori in AWS Management Console, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Dovrai abilitare la funzionalità per il test in. AWS Fault Injection Service Per ulteriori informazioni, consulta [Utilizzare le azioni AWS FIS `aws:ecs:task`](#).

Note

Se non usi il nuovo Amazon ECS ottimizzato AMIs o disponi di un'AMI personalizzata, installa le seguenti dipendenze:

- `tc`
- `sch_netem` modulo kernel

Endpoint di iniezione di errori Amazon ECS

L'agente del container Amazon ECS inserisce automaticamente la variabile di ambiente `ECS_AGENT_URI` nei container delle attività di Amazon ECS per fornire un metodo per interagire con l'endpoint API dell'agente del container. Ogni endpoint include un endpoint `/start/stop`, e `/status`. Gli endpoint accettano solo richieste da attività che hanno abilitato l'iniezione di errori e ogni endpoint ha un limite di frequenza di 1 richiesta ogni 5 secondi per contenitore. Il superamento di questo limite genera un errore.

Note

Amazon ECS Agent version `1.88.0+` è necessario per utilizzare gli endpoint di iniezione dei guasti.

I tre endpoint da utilizzare con l'iniezione di guasti sono:

- [Endpoint della porta blackhole di rete](#)
- [Endpoint di perdita di pacchetti di rete](#)
- [Endpoint di latenza di rete](#)

Una richiesta riuscita genera un codice di risposta `200` contenente un messaggio che indica `running` quando si chiama `/start` endpoint, `stopped` per `/stop` endpoint `running` e/o `not-running` per l'endpoint. `/status`

```
{
```

```
"Status": <string>
}
```

Una richiesta non riuscita restituisce uno dei seguenti codici di errore:

- 400- Richiesta errata
- 409- La richiesta di inserimento dei guasti è in conflitto con un altro errore in esecuzione
- 429- La richiesta è stata limitata
- 500- Il server ha avuto un errore imprevisto

```
{
  "Error": <string message>
}
```

Note

È possibile iniettare un errore di latenza di rete o un errore di perdita di pacchetti di rete alla volta. Il tentativo di iniettarne più di uno comporta il rifiuto della richiesta.

Endpoint della porta blackhole di rete

L'`{ECS_AGENT_URI}/fault/v1/network-blackhole-portendpoint` elimina il traffico in entrata o in uscita per una porta e un protocollo specifici nello spazio dei nomi di rete di un'attività ed è compatibile con due modalità:

- `awsvpc` - le modifiche vengono applicate allo spazio dei nomi della rete delle attività
- `host` - le modifiche vengono applicate all'istanza predefinita del contenitore dei namespace di rete

`{ECS_AGENT_URI}/fault/v1/network-blackhole-port/start`

Questo endpoint avvia le iniezioni di errori nelle porte blackhole di rete e presenta i seguenti parametri:

Porta

La porta specificata da utilizzare per l'iniezione dei guasti nelle porte blackhole.

Tipo: integer

Campo obbligatorio: sì

Protocollo

Il protocollo da utilizzare per l'iniezione di errori nella porta blackhole.

Tipo: stringa

Valori validi: tcp | udp

Campo obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione dei guasti.

Tipo: stringa

Valori validi: ingress | egress

Campo obbligatorio: sì

SourcesToFilter

Un array JSON di IPv4 indirizzi o blocchi CIDR protetti dall'errore.

Tipo: matrice di stringhe

Campo obbligatorio: no

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'`startendpoint` (sostituisci *red* i valori con i tuoi):

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-blackhole-port/start
```

```
Http method:POST
```

```
Request payload:
```

```
{  
  "Port": 1234,
```

```
"Protocol": "tcp|udp",
"TrafficType": "ingress|egress"
"SourcesToFilter": ["${IP1}", "${IP2}", ...],
}
```

{ECS_AGENT_URI}/fault/v1/network-blackhole-port/stop

Questo endpoint blocca l'errore specificato nella richiesta. Questo endpoint ha i seguenti parametri:

Porta

La porta interessata dall'errore che deve essere interrotta.

Tipo: integer

Campo obbligatorio: sì

Protocollo

Il protocollo da usare per fermare l'errore.

Tipo: stringa

Valori validi: tcp | udp

Campo obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione del guasto.

Tipo: stringa

Valori validi: ingress | egress

Campo obbligatorio: sì

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'stopendpoint (sostituisci i *red* valori con i tuoi):

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-blackhole-port/stop
```

```
Http method: POST
```

```
Request payload:
```

```
{  
  "Port": 1234,  
  "Protocol": "tcp|udp",  
  "TrafficType": "ingress|egress",  
}
```

{ECS_AGENT_URI}/fault/v1/network-blackhole-port/status

Questo endpoint viene utilizzato per verificare lo stato dell'iniezione del guasto. Questo endpoint ha i seguenti parametri:

Porta

La porta interessata per verificare lo stato del guasto.

Tipo: integer

Campo obbligatorio: sì

Protocollo

Il protocollo da utilizzare per verificare lo stato dell'errore.

Tipo: stringa

Valori validi: tcp | udp

Campo obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione del guasto.

Tipo: stringa

Valori validi: ingress | egress

Campo obbligatorio: sì

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'`status` endpoint (sostituisci i *red* valori con i tuoi):

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-blackhole-port/status

Http method: POST

Request payload:
{
  "Port": 1234,
  "Protocol": "tcp|udp",
  "TrafficType": "ingress|egress",
}
```

Endpoint di latenza di rete

L'`{ECS_AGENT_URI}/fault/v1/network-latencyendpoint` aggiunge ritardo e instabilità all'interfaccia di rete dell'attività per il traffico verso fonti specifiche. L'endpoint è compatibile con due modalità:

- `awsvpc` - le modifiche vengono applicate all'interfaccia di rete delle attività
- `host` - le modifiche vengono applicate all'interfaccia di rete predefinita

`{ECS_AGENT_URI}/fault/v1/network-latency/start`

Questo `/start` endpoint avvia l'iniezione dei guasti di latenza di rete e presenta i seguenti parametri:

`DelayMilliseconds`

Il numero di millisecondi di ritardo da aggiungere all'interfaccia di rete da utilizzare per l'iniezione dei guasti.

Tipo: integer

Campo obbligatorio: sì

`JitterMilliseconds`

Il numero di millisecondi di jitter da aggiungere all'interfaccia di rete da utilizzare per l'iniezione dei guasti.

Tipo: integer

Campo obbligatorio: sì

Origini

Un array JSON di IPv4 indirizzi o blocchi CIDR destinati all'uso con l'iniezione di errori.

Tipo: matrice di stringhe

Campo obbligatorio: sì

SourcesToFilter

Un array JSON di IPv4 indirizzi o blocchi CIDR protetti dall'errore. `SourcesToFilter` ha la priorità su `Sources`.

Tipo: matrice di stringhe

Campo obbligatorio: no

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'endpoint `/start` (sostituisci i *red* valori con i tuoi):

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-latency/start
```

```
Http method: POST
```

```
Request payload:
```

```
{
  "DelayMilliseconds": 123,
  "JitterMilliseconds": 123,
  "Sources": ["${IP1}", "${IP2}", ...],
  "SourcesToFilter": ["${IP1}", "${IP2}", ...],
}
```

`{ECS_AGENT_URI}/fault/v1/network-latency/stop` and `/status`

L'endpoint `{ECS_AGENT_URI}/fault/v1/network-latency/stop` arresta l'errore e ne verifica lo stato. `{ECS_AGENT_URI}/fault/v1/network-latency/status` stato.

Di seguito sono riportati due esempi di richieste per l'utilizzo degli endpoint `/stop` e degli endpoint `/status`. Entrambi utilizzano il POST HTTP metodo.

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-latency/stop
```

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-latency/status
```

Endpoint di perdita di pacchetti di rete

L'`{ECS_AGENT_URI}/fault/v1/network-packet-loss` endpoint aggiunge la perdita di pacchetti all'interfaccia di rete specificata. Questo endpoint è compatibile con due modalità:

- `awsvpc` - le modifiche vengono applicate all'interfaccia di rete delle attività
- `host` - le modifiche vengono applicate all'interfaccia di rete predefinita

`{ECS_AGENT_URI}/fault/v1/network-packet-loss/start`

Questo `/start` endpoint avvia l'iniezione di errore in caso di perdita di pacchetti di rete e presenta i seguenti parametri:

LossPercent

La percentuale di perdita di pacchetti

Tipo: integer

Campo obbligatorio: sì

Origini

Una matrice JSON di IPv4 indirizzi o blocchi CIDR da utilizzare per i test di iniezione dei guasti.

Tipo: matrice di stringhe

Campo obbligatorio: sì

SourcesToFilter

Un array JSON di IPv4 indirizzi o blocchi CIDR protetti dall'errore. `SourcesToFilter` ha la priorità su `Sources`

Tipo: matrice di stringhe

Campo obbligatorio: no

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'`start` endpoint (sostituisci i *red* valori con i tuoi):

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-packet-loss/start
```

```
Http method: POST
```

```
{
  "LossPercent": 6,
  "Sources": ["${IP1}", "${IP2}", ...],
  "SourcesToFilter": ["${IP1}", "${IP2}", ...],
}
```

`{ECS_AGENT_URI}/fault/v1/network-packet-loss/stop` and `/status`

L'`{ECS_AGENT_URI}/fault/v1/network-packet-loss/stop` endpoint arresta l'errore e ne verifica lo `{ECS_AGENT_URI}/fault/v1/network-packet-loss/status` stato. È supportato solo un tipo di errore alla volta.

Di seguito sono riportati due esempi di richieste per l'utilizzo degli endpoint `/stop` e degli `/status` endpoint. Entrambi utilizzano il POST HTTP metodo.

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-packet-loss/stop
```

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-packet-loss/status
```

Esegui la migrazione di un ARN di servizio breve di Amazon ECS a un ARN lungo

Amazon ECS assegna un Amazon Resource Name (ARN) univoco a ciascun servizio. I servizi creati prima del 2021 hanno un formato ARN breve:

```
arn:aws:ecs:region:aws_account_id:service/service-name
```

Amazon ECS ha modificato il formato ARN per includere il nome del cluster. Questo è un formato ARN lungo:

`arn:aws:ecs:region:aws_account_id:service/cluster-name/service-name`

Il servizio deve avere il formato ARN lungo per poter etichettare il servizio.

È possibile migrare un servizio con un formato ARN breve in un formato ARN lungo senza dover ricreare il servizio. Puoi usare l'API, la CLI o la console. Non è possibile annullare l'operazione di migrazione.

Se desideri utilizzare per AWS CloudFormation etichettare un servizio con un formato ARN breve, devi migrare il servizio utilizzando l'API, la CLI o la console. Una volta completata la migrazione, puoi utilizzare AWS CloudFormation per etichettare il servizio.

Se desideri utilizzare Terraform per etichettare un servizio con un formato ARN breve, devi migrare il servizio utilizzando l'API, la CLI o la console. Al termine della migrazione, puoi utilizzare Terraform per etichettare il servizio.

Una volta completata la migrazione, il servizio presenta le seguenti modifiche:

- Il formato ARN lungo

`arn:aws:ecs:region:aws_account_id:service/cluster-name/service-name`

- Quando esegui la migrazione utilizzando la console, Amazon ECS aggiunge un tag al servizio con la chiave impostata su «ecs: serviceArnMigrated At» e il valore impostato sul timestamp della migrazione (formato UTC).

Questo tag viene conteggiato ai fini della tua quota di tag.

- Quando `PhysicalResourceId` in uno AWS CloudFormation stack rappresenta un ARN di servizio, il valore non cambia e continuerà a essere l'ARN del servizio breve.

Prerequisiti

Eseguire le seguenti operazioni prima di migrare l'ARN del servizio.

1. Per verificare se disponi di un ARN di servizio breve, visualizza i dettagli del servizio nella console Amazon ECS (viene visualizzato un avviso quando il servizio ha il formato ARN breve) o il parametro `return da. serviceArn describe-services`. Quando l'ARN non include il nome del cluster, si dispone di un ARN breve. Di seguito è riportato il formato di un ARN breve:

`arn:aws:ecs:region:aws_account_id:service/service-name`

2. Nota la data di creazione.

3. Se disponi di policy IAM che utilizzano il formato ARN breve, aggiornalo al formato ARN lungo.

Sostituisci ogni *user input placeholder* con le tue informazioni.

```
arn:aws:ecs:region:aws_account_id:service/cluster-name/service-name
```

Per ulteriori informazioni, consulta [Modifica delle politiche IAM nella Guida](#) per l' AWS Identity and Access Management utente.

4. Se disponi di strumenti che utilizzano il formato ARN breve, aggiornalo al formato ARN lungo.

Sostituisci ogni *user input placeholder* con le tue informazioni.

```
arn:aws:ecs:region:aws_account_id:service/cluster-name/service-name
```

5. Abilita il formato ARN lungo del servizio. Esegui `put-account-setting` con l'opzione `serviceLongArnFormat` impostata su `enabled`. Per ulteriori informazioni, [put-account-setting](#) consulta la sezione Amazon Elastic Container Service API Reference.

Esegui il comando come utente root quando il servizio ha una `createdAt` data sconosciuta.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled
```

Output di esempio

```
{
  "setting": {
    "name": "serviceLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:role/your-role",
    "type": "user"
  }
}
```

6. Abilita il formato ARN lungo dell'attività. Ciò consente alle attività eseguite come parte di un servizio di avere il formato ARN lungo. Esegui `put-account-setting` con l'opzione `taskLongArnFormat` impostata su `enabled`. Per ulteriori informazioni, [put-account-setting](#) consulta la sezione Amazon Elastic Container Service API Reference.

Esegui il comando come utente root quando il servizio ha una `createdAt` data sconosciuta.

```
aws ecs put-account-setting --name taskLongArnFormat --value enabled
```

Output di esempio

```
{
  "setting": {
    "name": "taskLongArnFormat",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:role/your-role",
    "type": "user"
  }
}
```

Procedura

Usa quanto segue per migrare l'ARN del servizio.

Console

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Clusters (Cluster), scegli il cluster.
3. Nella sezione Servizi, scegli un servizio con un avviso nella colonna ARN.

Viene visualizzata la pagina dei dettagli del servizio.

4. Scegli Migra a un ARN lungo.

Viene visualizzata la finestra di dialogo del servizio Migrate.

5. Scegliere Migrate (Migrazione).

CLI

Dopo aver completato i prerequisiti, puoi taggare il tuo servizio. Esegui il comando seguente:

Amazon ECS valuta la possibilità di passare il formato ARN lungo in `tag-resource` una richiesta API per un servizio con un ARN breve come segnale per migrare il servizio all'utilizzo del formato ARN lungo.

```
aws ecs tag-resource \
  --resource-arn arn:aws:ecs:region:aws_account_id:service/cluster-name/service-name
  --tags key=key1,value=value1
```

L'esempio seguente contrassegna MyService con un tag con una chiave impostata su "" e un valore impostato su TestService ": WebServers

```
aws ecs tag-resource \  
  --resource-arn arn:aws:ecs:us-east-1:123456789012:service/MyCluster/MyService \  
  --tags key=TestService1,value=WebServers
```

Terraform

Dopo aver completato i prerequisiti, puoi taggare il tuo servizio. Create una `aws_ecs_service` risorsa e impostate il `tags` riferimento. Per ulteriori informazioni, consulta [Resource: aws_ecs_service](#) nella documentazione di Terraform.

```
resource "aws_ecs_service" "MyService" {  
  name      = "example"  
  cluster = aws_ecs_cluster.MyService.id  
  
  tags = {  
    "Name"      = "MyService"  
    "Environment" = "Production"  
    "Department" = "QualityAssurance"  
  }  
}
```

Passaggi successivi

Puoi aggiungere tag al servizio. Per ulteriori informazioni, consulta [Aggiungere tag alle risorse Amazon ECS](#).

Se desideri che Amazon ECS propaghi i tag dalla definizione dell'attività o dal servizio all'attività, esegui `update-service` con il `propagateTags` parametro. Per ulteriori informazioni, consulta [update-service](#) nel Reference. AWS Command Line Interface

Risoluzione dei problemi

Alcuni utenti potrebbero riscontrare il seguente errore durante la migrazione dal formato ARN breve al formato ARN lungo.

There was an error while migrating the ARN of service *service-name*. The specified account does not have `serviceLongArnFormat` or `taskLongArnFormat` account settings enabled. Add account settings in order to enable tagging.

Se hai già abilitato l'impostazione dell'`serviceLongArnFormat` account ma continui a riscontrare questo errore, è possibile che le impostazioni dell'account per il formato ARN lungo non siano state abilitate per il principale IAM specifico che ha originariamente creato il servizio.

1. Identifica il principale che ha creato il servizio.

1. Nella console, le informazioni sono disponibili nel campo Creato da nella scheda Configurazione e rete nella pagina dei dettagli del servizio nella console Amazon ECS.
2. Per il AWS CLI, esegui il seguente comando:

Sostituisci *user-input* con i tuoi valori.

```
aws ecs describe-services --cluster cluster-name --services service-name --query 'services[0].{createdBy: createdBy}'
```

2. Abilita le impostazioni dell'account richieste per quel committente specifico. Questa operazione può essere eseguita in uno dei seguenti modi:

- a. Assumi l'utente o il ruolo IAM per quel principale. Quindi esegui `put-account-setting`.
- b. Usa l'utente root per eseguire il comando specificando il principale di creazione con.
`principal-arn`

Esempio.

Sostituisci *principal-arn* con il valore del passaggio 1.

```
aws ecs put-account-setting --name serviceLongArnFormat --value enabled --principal-arn arn:aws:iam::123456789012:role/jdoe
```

Entrambi i metodi abilitano l'impostazione `serviceLongArnFormat` dell'account richiesta sul principale che ha creato il servizio, il che consente di procedere con la migrazione ARN.

Logica di accelerazione del servizio Amazon ECS

Il pianificatore di servizi Amazon ECS ora include la logica che limita la frequenza dei tentativi di riavvio dei processi in caso di avvii non riusciti ripetuti.

Se le attività relative a un servizio non rientrano ripetutamente nello `RUNNING` stato corrispondente (passando direttamente da uno `STOPPED` stato `PENDING` a a uno), il tempo che intercorre tra i

successivi tentativi di riavvio viene aumentato in modo incrementale fino a un massimo di 27 minuti. Questo periodo massimo è soggetto a modifiche in futuro. Questo comportamento riduce l'effetto delle attività fallite nelle risorse del cluster Amazon ECS o sui costi dell'infrastruttura Fargate. Se il servizio attiva la logica di limitazione, riceverai il seguente [messaggio di evento relativo al servizio](#):

```
(service service-name) is unable to consistently start tasks successfully.
```

Amazon ECS non impedisce mai a un servizio guasto di riprovare. Inoltre, non prova a modificarlo in alcun modo se non aumentando il tempo tra i riavvii. La logica di limitazione del servizio non fornisce parametri regolabili dall'utente.

Se aggiorni il servizio per l'utilizzo di una nuova definizione di attività, il servizio torna immediatamente a uno stato normale non limitato. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Di seguito sono riportate alcune cause comuni che avviano questa logica. Ti consigliamo di intraprendere un'azione manuale per risolvere il problema:

- Una mancanza di risorse che possono ospitare l'attività, quali porte, memoria o unità di CPU nel cluster. In questo caso, viene anche visualizzato il [messaggio di evento relativo al servizio delle risorse insufficienti](#).
- L'agente del container di Amazon ECS non è in grado di recuperare l'immagine Docker dell'attività. Ciò potrebbe essere dovuto al nome di un'immagine del container non corretto, a un'immagine o un tag non corretti o alla mancanza di autenticazione o delle autorizzazioni dei registri privati. In questo caso, viene anche visualizzato `CannotPullContainerError` negli [errori di attività interrotte](#).
- Lo spazio su disco insufficiente nell'istanza di container per poter creare il container. In questo caso, viene anche visualizzato `CannotCreateContainerError` negli [errori di attività interrotte](#). Per ulteriori informazioni, consulta [Risolvi i problemi relativi al Docker in Amazon API error \(500\): devmapper ECS](#).

Important

Le attività interrotte dopo che hanno raggiunto lo stato `RUNNING` non attivano la logica di limitazione o il messaggio di evento relativo al servizio associato. Ad esempio, se i controlli dell'integrità di Elastic Load Balancing non riusciti per un servizio fanno sì che un processo venga contrassegnato come non integro e Amazon ECS annulla la registrazione dell'attività

e la interrompe. A questo punto, le attività non vengono limitate. Anche se il comando del container di un'attività termina immediatamente con un codice di uscita diverso zero, l'attività è già passata allo stato RUNNING. Le attività che non riescono immediatamente a causa di errori di comando non provocano la limitazione o il messaggio di evento relativo al servizio.

Parametri di definizione del servizio Amazon ECS

Una definizione di servizio definisce come eseguire il servizio Amazon ECS. I seguenti parametri possono essere specificati in una definizione di servizio.

Tipo di avvio

`launchType`

Tipo: stringa

Valori validi: EC2 | FARGATE | EXTERNAL

Campo obbligatorio: no

Il tipo di avvio con cui eseguire il servizio. Se non viene specificato un tipo di avvio, per impostazione predefinita viene utilizzato il `capacityProviderStrategy` di default. Per ulteriori informazioni, consulta [Tipi di avvio di Amazon ECS](#).

Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

Strategia del provider di capacità

`capacityProviderStrategy`

Tipo: matrice di oggetti

Campo obbligatorio: no

La strategia del fornitore di capacità da utilizzare per il servizio.

Una strategia del fornitore di capacità consiste di uno o più fornitori di capacità e dei valori base e `weight` da assegnare a essi. Un provider di capacità deve essere associato al cluster da

utilizzare in una strategia del provider di capacità. L' `PutClusterCapacityProviders` API viene utilizzata per associare un provider di capacità a un cluster. È possibile utilizzare solo i provider di capacità con uno stato `ACTIVE` o uno stato `UPDATING`.

Se viene specificato un `capacityProviderStrategy`, il parametro `launchType` deve essere omesso. Se non viene specificato `capacityProviderStrategy` o `launchType`, viene utilizzato `defaultCapacityProviderStrategy` per il cluster.

Se desideri specificare un provider di capacità che utilizzi un gruppo con scalabilità automatica, il provider di capacità deve essere già stato creato. È possibile creare nuovi fornitori di capacità con l'operazione `CreateCapacityProvider` API.

Per utilizzare un provider di capacità AWS Fargate, specificare `FARGATE` o i provider di `FARGATE_SPOT` capacità. I provider di capacità AWS Fargate sono disponibili per tutti gli account e devono essere associati solo a un cluster per essere utilizzati.

L'operazione `PutClusterCapacityProviders` API viene utilizzata per aggiornare l'elenco dei fornitori di capacità disponibili per un cluster dopo la creazione del cluster.

`capacityProvider`

Tipo: stringa

Campo obbligatorio: sì

Il nome breve o il nome della risorsa Amazon (ARN) del provider di capacità.

`weight`

Tipo: integer

Intervallo valido: numeri interi compresi tra 0 e 1.000.

Campo obbligatorio: no

Il valore del peso indica la percentuale relativa del numero totale di attività avviate che devono utilizzare il provider di capacità specificato.

Ad esempio, supponiamo di avere una strategia che contiene due provider di capacità, ognuno con un peso pari a uno. Una volta soddisfatta la base, le attività si dividono equamente tra i due provider di capacità. Usando la stessa logica, assumiamo di specificare un peso di 1 per `capacityProviderA` e un peso di 4 per `capacityProviderB`. Quindi, per ogni attività che viene eseguita utilizzando `capacityProviderA`, quattro attività utilizzano `capacityProviderB`.

base

Tipo: integer

Intervallo valido: numeri interi compresi tra 0 e 100.000.

Campo obbligatorio: no

Il valore di base indica il numero minimo di attività da eseguire nel provider di capacità specificato. Solo un provider di capacità in una strategia di provider di capacità può avere una base definita.

Definizione di attività

taskDefinition

Tipo: string

Campo obbligatorio: no

I valori `family` e `revision` (`family:revision`) o il nome della risorsa Amazon (ARN) completo della definizione di attività da eseguire nel servizio. Se non viene specificato un `revision`, viene utilizzata l'ultima revisione ACTIVE della famiglia specificata.

Una definizione di attività deve essere specificata quando si utilizza il controller di distribuzione di aggiornamento in sequenza (ECS).

Sistema operativo della piattaforma

platformFamily

Tipo: stringa

Obbligatorio: condizionale

Di default: Linux

Questo parametro è richiesto per i servizi Amazon ECS ospitati su Fargate.

Questo parametro viene ignorato per i servizi Amazon ECS ospitati su Amazon. EC2

Il sistema operativo sui container che esegue il servizio. I valori validi sono LINUX, WINDOWS_SERVER_2019_FULL, WINDOWS_SERVER_2019_CORE, WINDOWS_SERVER_2022_FULL e WINDOWS_SERVER_2022_CORE.

Il valore `platformFamily` per ogni processo specificato per il servizio deve corrispondere al valore `platformFamily` del servizio. Ad esempio, se hai impostato `platformFamily` su `WINDOWS_SERVER_2019_FULL`, il valore `platformFamily` per tutte le attività deve essere `WINDOWS_SERVER_2019_FULL`.

Versione della piattaforma

`platformVersion`

Tipo: string

Campo obbligatorio: no

La versione della piattaforma su cui sono in esecuzione le attività nel servizio. Viene specificata una versione della piattaforma solo per le attività con tipo di avvio Fargate. Se non è specificata, la versione più recente (LATEST) viene utilizzata di default.

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Quando specifichi la versione della piattaforma LATEST durante l'esecuzione di un'attività o la creazione di un servizio, ottieni la versione di piattaforma più aggiornata disponibile per le tue attività. Quando incrementi il servizio, tali attività riceveranno la versione della piattaforma specificata nell'implementazione corrente del servizio. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Note

Le versioni della piattaforma non sono specificate per le attività che utilizzano il tipo di avvio EC2.

Cluster

`cluster`

Tipo: string

Campo obbligatorio: no

Il nome breve o il nome della risorsa Amazon (ARN) completo del cluster in cui eseguire il servizio. Se non specifichi un cluster, viene utilizzato il cluster di default.

Nome servizio

`serviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del servizio. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. I nomi dei servizi devono essere univoci all'interno di un cluster, ma puoi avere servizi dai nomi simili in più cluster all'interno di una Regione o in più Regioni.

Strategia di pianificazione

`schedulingStrategy`

Tipo: stringa

Valori validi: REPLICa | DAEMON

Campo obbligatorio: no

La strategia di pianificazione da utilizzare. Se non viene specificata alcuna strategia di pianificazione, viene utilizzata la strategia REPLICa. Per ulteriori informazioni, consulta [Servizi Amazon ECS](#).

Sono disponibili due strategie del pianificatore del servizio:

- **REPLICa:** la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le decisioni riguardo al posizionamento delle attività. Per ulteriori informazioni, consulta [Strategia di replica](#).
- **DAEMON:** la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività

specificati nel cluster. Quando si utilizza questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta [Strategia daemon](#).

**Note**

I processi Fargate non supportano la strategia di pianificazione DAEMON

Conteggio desiderato

`desiredCount`

Tipo: integer

Campo obbligatorio: no

Il numero di istanze della definizione di attività specificata da posizionare e mantenere in esecuzione nel servizio.

Questo parametro è obbligatorio se si utilizza la strategia di pianificazione REPLICA. Se il servizio utilizza la strategia di pianificazione DAEMON, questo parametro è facoltativo.

Quando si utilizza il service auto scaling, quando si aggiorna un servizio attualmente in esecuzione con un numero di attività `desiredCount` inferiore al numero di attività attualmente in esecuzione, il servizio viene ridimensionato fino al valore specificato. `desiredCount`

Configurazione dell'implementazione

`deploymentConfiguration`

Tipo: oggetto

Campo obbligatorio: no

I parametri di distribuzione opzionali che determinano quante attività vengono eseguite durante l'implementazione e l'ordine di arresto e di avvio delle attività.

`maximumPercent`

Tipo: integer

Campo obbligatorio: no

Se il servizio utilizza il tipo di implementazione con aggiornamento in sequenza (ECS), il parametro `maximumPercent` rappresenta un limite superiore sul numero di attività del servizio che sono consentite nello stato `RUNNING`, `STOPPING` o `PENDING` durante un'implementazione. Viene espresso come percentuale del `desiredCount` arrotondato per difetto al valore intero più vicino. Questo parametro può essere utilizzato per definire le dimensioni del batch di implementazione. Ad esempio, se il servizio utilizza l'utilità di pianificazione del `REPLICA` servizio e ha un numero `desiredCount` di quattro attività e un `maximumPercent` valore del 200%, lo scheduler avvia quattro nuove attività prima di interrompere le quattro attività precedenti, a condizione che le risorse del cluster necessarie per questa operazione siano disponibili. Il valore predefinito `maximumPercent` per un servizio che utilizza il pianificatore di servizi `REPLICA` è del 200%.

Lo scheduler di Amazon ECS utilizza questo parametro per sostituire le attività non integre avviando prima le attività sostitutive e poi interrompendo quelle non integre, purché siano disponibili risorse del cluster per avviare attività sostitutive. Per ulteriori informazioni su come lo scheduler sostituisce le attività non integre, consulta [Servizi Amazon ECS](#).

Se il servizio sta utilizzando il tipo di pianificatore di servizi `DAEMON`, il parametro `maximumPercent` deve rimanere al 100%. Si tratta del valore di default.

Il numero massimo di attività durante un'implementazione è uguale a `desiredCount` moltiplicato per `maximumPercent/100`, arrotondato per difetto al numero intero più vicino.

Se un servizio utilizza i tipi e le attività di distribuzione `blue/green` (`CODE_DEPLOY`) o di `EXTERNAL` distribuzione che utilizzano il tipo di `EC2` avvio, il valore percentuale massimo viene impostato sul valore predefinito. Il valore viene utilizzato per definire il limite massimo del numero di attività del servizio che rimangono nello `RUNNING` stato mentre le istanze del contenitore si trovano nello stato `DRAINING`.

Note

Non è possibile specificare un `maximumPercent` valore personalizzato per un servizio che utilizza i tipi `blue/green` (`CODE_DEPLOY`) o `EXTERNAL` deployment e ha attività che utilizzano il `EC2` tipo di avvio.

Se il servizio utilizza il tipo blue/green (CODE_DEPLOY) o EXTERNAL deployment e le attività del servizio utilizzano il tipo di avvio Fargate, il valore percentuale massimo non viene utilizzato. Il valore viene comunque restituito quando si descrive il servizio.

`minimumHealthyPercent`

Tipo: integer

Campo obbligatorio: no

Se il servizio utilizza il tipo di implementazione aggiornamento in sequenza (ECS), il parametro `minimumHealthyPercent` rappresenta un limite inferiore sul numero di attività del servizio che devono restare nello stato RUNNING durante un'implementazione. Viene espresso come percentuale del `desiredCount` arrotondato per eccesso al valore intero più vicino. Questo parametro può essere utilizzato per eseguire l'implementazione senza utilizzare capacità aggiuntiva del cluster.

Ad esempio, se il servizio ha quattro attività, una `desiredCount` del 50% e una `minimumHealthyPercent` `maximumPercent` del 100%, il service scheduler interrompe due attività esistenti per liberare la capacità del cluster prima di iniziare due nuove attività.

Se alcune attività non sono integre e `maximumPercent` non consente allo scheduler di Amazon ECS di avviare attività sostitutive, lo scheduler interrompe le attività non integre, utilizzando il vincolo `minimumHealthyPercent` come vincolo, per liberare la capacità necessaria per avviare attività one-by-one sostitutive. Per ulteriori informazioni su come lo scheduler sostituisce le attività non salutari, consulta [Servizi Amazon ECS](#).

Per i servizi che non utilizzano un load balancer, considera quanto riportato di seguito:

- Un servizio è considerato integro se tutti i container essenziali all'interno delle attività del servizio superano i controlli dello stato.
- Se un'attività non dispone di container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà 40 secondi dopo che un'attività raggiunge uno stato RUNNING prima di contarla per la percentuale minima di integrità totale.
- Se un'attività ha uno o più container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà che l'attività raggiunga uno stato integro prima di contarla per la percentuale minima di integrità totale. Un'attività è considerata sana quando tutti i container essenziali al suo interno hanno superato i controlli dello stato. La quantità di tempo che il pianificatore del servizio può attendere è determinata dalle impostazioni di controllo dello stato del container. Per ulteriori informazioni, consulta [Controllo dello stato](#).

Per i servizi che utilizzano un load balancer, considera quanto riportato di seguito:

- Se un'attività non dispone di container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà che il controllo dell'integrità del gruppo di destinazione del load balancer restituisca uno stato integro prima di contarla nella percentuale minima di integrità totale.
- Se un'attività dispone di un container essenziale con un controllo dell'integrità definito, il pianificatore del servizio attenderà che l'attività raggiunga uno stato integro e il controllo dell'integrità del gruppo di destinazione del load balancer restituisca uno stato integro prima di contare l'attività nella percentuale minima di integrità totale.

Il valore predefinito di un servizio di replica per `minimumHealthyPercent` è 100%. Il `minimumHealthyPercent` valore predefinito per un servizio che utilizza la pianificazione del DAEMON servizio è 0% per AWS CLI AWS SDKs, il e APIs e 50% per. AWS Management Console

Il numero minimo di attività integre durante un'implementazione è uguale a `desiredCount` moltiplicato per `minimumHealthyPercent/100`, arrotondato per eccesso al numero intero più vicino.

Se un servizio utilizza il tipo blue/green (`CODE_DEPLOY`) o `EXTERNAL deployment` e esegue attività che utilizzano il tipo di EC2 avvio, il valore della percentuale minima integra viene impostato sul valore predefinito. Il valore viene utilizzato per definire il limite inferiore al numero di attività del servizio che rimangono nello `RUNNING` stato mentre le istanze del contenitore si trovano nello stato. `DRAINING`

 Note

Non è possibile specificare un `maximumPercent` valore personalizzato per un servizio che utilizza i tipi blue/green (`CODE_DEPLOY`) o `EXTERNAL deployment` e ha attività che utilizzano il EC2 tipo di avvio.

Se un servizio utilizza i tipi di implementazione blu/verde (`CODE_DEPLOY`) o `EXTERNAL` ed esegue processi che utilizzano il tipo di avvio Fargate, il valore della percentuale minima di integrità non viene utilizzato, anche se viene restituito nella descrizione del servizio.

Controller di implementazione

deploymentController

Tipo: oggetto

Campo obbligatorio: no

Il tipo di controller di distribuzione da utilizzare per il servizio. Se non viene specificato alcun controller di implementazione, viene utilizzato il controller ECS. Per ulteriori informazioni, consulta [Servizi Amazon ECS](#).

type

Tipo: stringa

Valori validi: ECS | CODE_DEPLOY | EXTERNAL

Obbligatorio: sì

Il tipo di controller di distribuzione da utilizzare. Esistono tre tipi di controller di distribuzione:

ECS

Il tipo di distribuzione aggiornamento in sequenza (ECS) prevede la sostituzione della versione attualmente in esecuzione del container con quella più recente. Il numero di container che Amazon ECS aggiunge o rimuove dal servizio durante un aggiornamento in sequenza è controllato regolando il numero minimo e massimo di integrità consentite durante l'implementazione di un servizio, come specificato in [deploymentConfiguration](#).

CODE_DEPLOY

Il tipo di distribuzione blue/green (CODE_DEPLOY) utilizza il modello di distribuzione blue/green powered by CodeDeploy, che consente di verificare una nuova implementazione di un servizio prima di inviargli traffico di produzione.

EXTERNAL

Utilizza il tipo di implementazione esterna quando desideri usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implementazione per un servizio Amazon ECS.

Posizionamento dei processi

placementConstraints

Tipo: matrice di oggetti

Campo obbligatorio: no

Serie di oggetti vincolo di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di 10 vincoli per attività. Questo limite include i vincoli nella definizione dell'attività e quelli specificati in fase di runtime. Se utilizzi il tipo di avvio Fargate, i vincoli di posizionamento delle attività non sono supportati.

type

Tipo: string

Campo obbligatorio: no

Il tipo di vincolo. Utilizza `distinctInstance` per accertarti che ogni attività in un determinato gruppo sia in esecuzione su un'istanza di container diversa. Utilizza `memberOf` per limitare la selezione a un gruppo di candidati validi. Il valore `distinctInstance` non è supportato nelle definizioni di attività.

expression

Tipo: string

Campo obbligatorio: no

Un'espressione del linguaggio di query del cluster da applicare al vincolo. Non puoi specificare un'espressione se il tipo di vincolo è `distinctInstance`. Per ulteriori informazioni, consulta [Crea espressioni per definire istanze di container per le attività di Amazon ECS](#).

placementStrategy

Tipo: matrice di oggetti

Campo obbligatorio: no

Gli oggetti strategia di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di quattro regole di strategia per ogni servizio.

type

Tipo: stringa

Valori validi: `random` | `spread` | `binpack`

Campo obbligatorio: no

Il tipo di strategia di posizionamento. La strategia di posizionamento `random` posiziona in modo casuale le attività sui candidati disponibili. La strategia di posizionamento `spread` distribuisce le attività in modo uniforme tra i candidati disponibili in base al parametro `field`. La strategia `binpack` posiziona le attività sui candidati disponibili che dispongono della quantità minima di risorsa specificata con il parametro `field`. Ad esempio, se adotti il binpacking della memoria, un'attività verrà posizionata sull'istanza con la quantità minima di memoria residua (ma ancora sufficiente a eseguire l'attività).

field

Tipo: string

Campo obbligatorio: no

Il campo a cui applicare la strategia di posizionamento. Per la strategia di posizionamento `spread`, i valori validi sono `instanceId` (o `host`, che ha lo stesso effetto) oppure qualsiasi attributo personalizzato o di piattaforma applicato a un'istanza di container, come `attribute:ecs.availability-zone`. Per la strategia di posizionamento `binpack`, i valori validi sono `cpu` e `memory`. Per la strategia di posizionamento `random` questo campo non viene utilizzato.

Tag

tags

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati applicati al servizio per aiutarti a catalogarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Quando un servizio viene eliminato, vengono eliminati anche i tag. Al servizio è possibile applicare un massimo di 50 tag. Per ulteriori informazioni, consulta [Taggare le risorse Amazon ECS](#).

key

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

value

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

enableECSTags

Tipo: Booleano

Valori validi: true | false

Campo obbligatorio: no

Specifica se usare i tag gestiti di Amazon ECS per i processi nel servizio. Se non viene specificato alcun valore, il valore predefinito è false. Per ulteriori informazioni, consulta [Utilizza i tag per la fatturazione](#).

propagateTags

Tipo: stringa

Valori validi: TASK_DEFINITION | SERVICE

Campo obbligatorio: no

Specifica se copiare i tag dalla definizione di attività o dal servizio nelle attività del servizio. Se non viene specificato alcun valore, i tag non vengono copiati. I tag possono essere copiati solo nelle attività all'interno del servizio durante la creazione del servizio. Per aggiungere tag a un processo dopo la creazione di un servizio o di un processo, utilizza l'operazione API TagResource.

Configurazione della rete

networkConfiguration

Tipo: oggetto

Campo obbligatorio: no

La configurazione di rete per il servizio. Questo parametro è obbligatorio per le definizioni di attività che utilizzano la modalità di rete awsvpc per ricevere la propria interfaccia di rete elastica e non è supportato per altre modalità. Con il tipo di avvio Fargate, la modalità di rete awsvpc è obbligatoria. Per ulteriori informazioni sul networking per il tipo di EC2 lancio di Amazon, consulta [Opzioni di task networking di Amazon ECS per il tipo di EC2 avvio](#). Per ulteriori informazioni sul networking per il tipo di lancio Fargate, consulta le [opzioni di task networking di Amazon ECS per il tipo di lancio Fargate](#).

awsvpcConfiguration

Tipo: oggetto

Campo obbligatorio: no

Un oggetto che rappresenta le sottoreti e i gruppi di sicurezza per un'attività o un servizio.

subnets

Tipo: matrice di stringhe

Campo obbligatorio: sì

Le sottoreti associate all'attività o al servizio. Esiste un limite di 16 sottoreti che possono essere specificate in base a awsvpcConfiguration.

securityGroups

Tipo: matrice di stringhe

Campo obbligatorio: no

I gruppi di sicurezza associati all'attività o al servizio. Se non specifichi un gruppo di sicurezza, viene utilizzato il gruppo di sicurezza predefinito dell'ambiente VPC. Esiste un limite di cinque gruppi di sicurezza che possono essere specificati in base a awsvpcConfiguration.

`assignPublicIP`

Tipo: stringa

Valori validi: ENABLED | DISABLED

Campo obbligatorio: no

Se l'interfaccia di rete elastica dell'attività riceve un indirizzo IP pubblico. Se non viene specificato alcun valore, viene utilizzato il valore predefinito DISABLED.

`healthCheckGracePeriodSeconds`

Tipo: integer

Campo obbligatorio: no

Periodo di tempo, in secondi, durante il quale lo scheduler del servizio Amazon ECS ignora i controlli non corretti di Elastic Load Balancing, VPC Lattice e dello stato dei container dopo il primo avvio di un'attività. Se non si specifica un valore del periodo di tolleranza per il controllo dello stato di salute, viene utilizzato il valore predefinito 0. Se non si utilizza nessuno dei controlli sanitari, non `healthCheckGracePeriodSeconds` viene utilizzato.

Se l'avvio e la risposta delle attività del servizio richiedono un po' di tempo, è possibile specificare un periodo di tolleranza per il controllo sanitario fino a 2.147.483.647 secondi (circa 69 anni). Durante questo periodo, lo scheduler del servizio Amazon ECS ignora lo stato del controllo dello stato di integrità. Questo periodo di tolleranza può evitare che il pianificatore del servizio contrassegni le attività come non integre e le interrompa prima che abbiano il tempo di iniziare.

`loadBalancers`

Tipo: matrice di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il load balancer da utilizzare con il tuo servizio. Per i servizi che utilizzano un Application Load Balancer o un Network Load Balancer, esiste un limite di cinque gruppi di destinazione che puoi collegare a un servizio.

Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. Puoi utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del bilanciamento del carico solo per il controller di distribuzione ECS mobile, non blu/verde o esterno. AWS CloudFormation AWS CodeDeploy Quando aggiungi, aggiorni o

rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento [UpdateService](#) all'API di Amazon Elastic Container Service.

Per gli Application Load Balancer e i Network Load Balancer, questo oggetto deve contenere l'ARN del gruppo di destinazione del load balancer, il nome del container (così come appare nella definizione di quest'ultimo) e la porta del container per accedere dal load balancer. Quando un'attività di questo servizio è posizionata su un'istanza di container, l'unione dell'istanza e della porta del container viene registrata come destinazione nel gruppo di destinazione specificato qui.

`targetGroupArn`

Tipo: string

Campo obbligatorio: no

Il nome della risorsa Amazon (ARN) completo del gruppo o dei gruppi di destinazione dell'Elastic Load Balancer associati a un servizio.

L'ARN del gruppo target viene specificato solo quando si utilizza un Application Load Balancer o un Network Load Balancer.

`loadBalancerName`

Tipo: string

Campo obbligatorio: no

Il nome del load balancer da associare al servizio.

Se utilizzi un Application Load Balancer o un Network Load Balancer, il parametro relativo al nome del load balancer deve essere omissivo.

`containerName`

Tipo: string

Campo obbligatorio: no

Il nome del container (così come appare nella definizione di quest'ultimo) da associare al load balancer.

containerPort

Tipo: integer

Campo obbligatorio: no

La porta del container da associare al load balancer. Questa porta deve corrispondere a un `containerPort` nella definizione di attività utilizzata dalle attività nel servizio. Per le attività che utilizzano il tipo di EC2 avvio, l'istanza del contenitore deve consentire il traffico in entrata sulla mappatura `hostPort` delle porte.

role

Tipo: string

Campo obbligatorio: no

Il nome breve o l'ARN completo del ruolo IAM che consente ad Amazon ECS di effettuare chiamate al load balancer per tuo conto. Questo parametro è consentito solo se utilizzi un load balancer con un singolo gruppo di destinazione e se la definizione di attività non utilizza la modalità di rete `awsvpc`. Se specifichi il parametro `role`, devi anche specificare un oggetto load balancer con il parametro `loadBalancers`.

Se il ruolo specificato ha un percorso diverso da `/`, devi specificare l'ARN del ruolo completo (scelta consigliata) o anteporre il percorso al nome del ruolo come prefisso. Ad esempio, se un ruolo con il nome `bar` ha il percorso `/foo/`, devi specificare `/foo/bar` come nome del ruolo. Per ulteriori informazioni, consulta [Nomi descrittivi e percorsi](#) nella Guida per l'utente IAM.

Important

Se il tuo account ha già creato il ruolo collegato al servizio Amazon ECS, tale ruolo viene utilizzato di default per il tuo servizio, a meno che tu non specifichi un ruolo qui. Il ruolo collegato al servizio è obbligatorio se la definizione di attività utilizza la modalità di rete `awsvpc`; in tal caso, non è necessario specificare un ruolo qui. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).

serviceConnectConfiguration

Tipo: oggetto

Campo obbligatorio: no

La configurazione di questo servizio per rilevare e connettersi ai servizi, ed essere rilevati e connessi da altri servizi all'interno di uno spazio nomi.

Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

enabled

Tipo: Booleano

Campo obbligatorio: sì

Specifica se utilizzare Service Connect con questo servizio.

namespace

Tipo: string

Campo obbligatorio: no

Il nome breve o il nome completo Amazon Resource Name (ARN) dello spazio dei nomi AWS Cloud Map da utilizzare con Service Connect. Lo spazio dei nomi deve trovarsi nella stessa Regione AWS del cluster e del servizio Amazon ECS. Il tipo di spazio dei nomi non influisce su Service Connect. Per ulteriori informazioni AWS Cloud Map, consulta [Working with Services](#) nella Developer Guide.

services

Tipo: matrice di oggetti

Campo obbligatorio: no

Un array degli oggetti del servizio Service Connect. Questi sono nomi e alias (detti anche endpoint) utilizzati da altri servizi Amazon ECS per connettersi a questo servizio.

Questo campo non è obbligatorio per un servizio Amazon ECS "client" che un membro di uno spazio dei nomi utilizza solo per connettersi ad altri servizi nello spazio dei nomi. Un esempio può essere un'applicazione front-end che accetta richieste in arrivo da un sistema di bilanciamento del carico collegato al servizio o con altri mezzi.

Un oggetto seleziona una porta dalla definizione dell'attività, assegna un nome al AWS Cloud Map servizio e una serie di alias (noti anche come endpoint) e porte alle applicazioni client per fare riferimento a questo servizio.

portName

Tipo: stringa

Campo obbligatorio: sì

portName deve corrispondere al name di una delle portMappings da tutti i container nella definizione di attività di questo servizio Amazon ECS.

discoveryName

Tipo: string

Campo obbligatorio: no

discoveryName È il nome del nuovo AWS Cloud Map servizio creato da Amazon ECS per questo servizio Amazon ECS. Deve essere univoco nello spazio dei nomi AWS Cloud Map .

Se questo campo non è specificato, viene utilizzato portName.

clientAliases

Tipo: matrice di oggetti

Campo obbligatorio: no

L'elenco degli alias client per questo servizio Service Connect. Si usano per assegnare nomi che possono essere utilizzati da applicazioni client. Il numero massimo di alias client che si può avere in questo elenco è 1.

Ogni alias ("endpoint") è un nome DNS completo e un numero di porta che altre attività Amazon ECS ("client") possono utilizzare per connettersi a questo servizio.

Ogni combinazione di nome e porta deve essere univoca nello spazio dei nomi.

Questi nomi sono configurati all'interno di ogni attività del servizio client, non in AWS Cloud Map. Le richieste DNS per risolvere questi nomi non lasciano l'attività e non vengono conteggiate ai fini della quota di richieste DNS al secondo per interfaccia di rete elastica.

port

Tipo: integer

Campo obbligatorio: sì

Il numero della porta di ascolto per il proxy Service Connect. Questa porta è disponibile all'interno di tutte le attività nello stesso spazio dei nomi.

Per evitare di modificare le applicazioni nei servizi client Amazon ECS, imposta la stessa porta utilizzata dall'applicazione client per impostazione predefinita.

`dnsName`

Tipo: string

Campo obbligatorio: no

Il `dnsName` è il nome utilizzato nelle applicazioni di attività client per connettersi a questo servizio. Il nome deve essere un'etichetta DNS valida.

Se questo campo non viene specificato, il valore predefinito sarà `discoveryName.namespace`. Se il valore `discoveryName` non viene specificato, viene utilizzato il valore `portName` della definizione dell'attività.

Per evitare di modificare le applicazioni nei servizi client Amazon ECS, imposta lo stesso nome utilizzato dall'applicazione client per impostazione predefinita. Ad esempio, alcuni nomi comuni sono `database`, `db` o il nome minuscolo di un database, ad esempio `mysql` o `redis`.

`ingressPortOverride`

Tipo: integer

Campo obbligatorio: no

(Facoltativo) Il numero della porta di ascolto per il proxy Service Connect.

Utilizza il valore di questo campo per ignorare il proxy per il traffico sul numero di porta specificato nella `portMapping` denominata nella definizione di attività di questa applicazione, quindi utilizzalo nei tuoi gruppi di sicurezza Amazon VPC per consentire il traffico nel proxy per questo servizio Amazon ECS.

In modalità `awsvpc`, il valore predefinito è il numero di porta del container specificato nella `portMapping` nella definizione di attività di questa applicazione. In modalità `bridge`, il valore predefinito è la porta temporanea dinamica del proxy Service Connect.

`logConfiguration`

Tipo: oggetto [LogConfiguration](#)

Campo obbligatorio: no

Questo definisce dove vengono pubblicati i log del proxy di Service Connect. Usa i log per il debug durante eventi imprevisti. Questa configurazione imposta il parametro `logConfiguration` nel container del proxy Service Connect in ogni attività di questo servizio Amazon ECS. Il container del proxy non è specificato nella definizione di attività.

Ti consigliamo di utilizzare la stessa configurazione di log dei container delle applicazioni della definizione di attività per questo servizio Amazon ECS. Infatti FireLens, questa è la configurazione del registro del contenitore dell'applicazione. Non è il contenitore del FireLens log router che utilizza l'immagine del `fluentd` contenitore `fluent-bit` o `serviceRegistries`

Tipo: matrice di oggetti

Campo obbligatorio: no

I dettagli della configurazione dell'individuazione per il tuo servizio. Per ulteriori informazioni, consulta [Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS](#).

`registryArn`

Tipo: string

Campo obbligatorio: no

Il nome della risorsa Amazon (ARN) del registro del servizio. Il registro dei servizi attualmente supportato è AWS Cloud Map. Per ulteriori informazioni, consulta [Utilizzo dei servizi](#) nella Guida per gli sviluppatori di AWS Cloud Map .

`port`

Tipo: integer

Campo obbligatorio: no

Il valore della porta utilizzato se l'individuazione dei servizi ha specificato un record SRV. Questo campo è obbligatorio se vengono utilizzati entrambi la modalità di rete `awsvpc` e i record SRV.

`containerName`

Tipo: string

Campo obbligatorio: no

Il valore che indica il nome del container da utilizzare per l'individuazione del servizio. Questo valore è specificato nella definizione dell'attività. Se la definizione di attività specificata dall'attività del servizio utilizza le modalità di rete `bridge` o `host`, devi specificare una combinazione di `containerName` e `containerPort` nella definizione di attività. Se la definizione di attività specificata dall'attività del servizio utilizza la modalità di rete `awsvpc` e utilizzi un record tipo DNS SRV, devi specificare una combinazione di `containerName` e `containerPort` o un valore `port`, ma non entrambi.

`containerPort`

Tipo: integer

Campo obbligatorio: no

Il valore della porta da utilizzare per l'individuazione del servizio. Questo valore è specificato nella definizione dell'attività. Se la definizione di attività specificata dall'attività del servizio utilizza le modalità di rete `bridge` o `host`, devi specificare una combinazione di `containerName` e `containerPort` nella definizione di attività. Se la definizione di attività specificata dall'attività del servizio utilizza la modalità di rete `awsvpc` e utilizzi un record tipo DNS SRV, devi specificare una combinazione di `containerName` e `containerPort` o un valore `port`, ma non entrambi.

Token client

`clientToken`

Tipo: string

Campo obbligatorio: no

L'identificatore univoco (con distinzione tra maiuscole e minuscole) che fornisci per assicurare l'idempotenza della richiesta. Può contenere fino a 32 caratteri ASCII.

Ribilanciamento della zona di disponibilità

`availabilityZoneRebalancing`

Tipo: string

Campo obbligatorio: no

Indica se il servizio utilizza il ribilanciamento della zona di disponibilità. I valori validi sono `ENABLED` `DISABLED`. Per ulteriori informazioni sul ribilanciamento della zona di disponibilità, vedere [Bilanciamento di un servizio Amazon ECS tra le zone di disponibilità](#).

Configurazioni dei volumi

`volumeConfigurations`

Tipo: oggetto

Campo obbligatorio: no

La configurazione che verrà utilizzata per creare volumi per le attività gestite dal servizio. Solo i volumi contrassegnati come `configuredAtLaunch` nella definizione dell'attività possono essere configurati utilizzando questo oggetto.

Questo oggetto è necessario per collegare i volumi Amazon EBS alle attività gestite da un servizio. Per ulteriori informazioni, consulta [Usa i volumi Amazon EBS con Amazon ECS](#).

`name`

Tipo: stringa

Campo obbligatorio: sì

Il nome di un volume configurato durante la creazione o l'aggiornamento di un servizio. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, caratteri di sottolineatura (`_`) e trattini (`-`). - Questo valore deve corrispondere al nome del volume specificato nella definizione dell'attività.

`managedEBSVolume`

Tipo: oggetto

Campo obbligatorio: no

La configurazione del volume utilizzata per creare volumi Amazon EBS collegati alle attività gestite da un servizio al momento della creazione o dell'aggiornamento del servizio. Viene allegato un volume per operazione.

encrypted

Tipo: Booleano

Campo obbligatorio: no

Valori validi: true|false

Specifica se crittografare ogni volume Amazon EBS creato. Se hai attivato la crittografia Amazon EBS per impostazione predefinita per un determinato Regione AWS prodotto Account AWS ma hai impostato questo parametro su `false`, questo parametro verrà sovrascritto e i volumi verranno crittografati con la chiave KMS specificata per la crittografia per impostazione predefinita. Per ulteriori informazioni sulla crittografia di Amazon EBS per impostazione predefinita, consulta [Enable Amazon EBS encryption by default](#) nella Amazon EBS User Guide. Per ulteriori informazioni sulla crittografia dei volumi Amazon EBS collegati alle attività di Amazon ECS, consulta [Crittografa i dati archiviati nei volumi Amazon EBS collegati alle attività di Amazon ECS](#)

kmsKeyId

Tipo: string

Campo obbligatorio: no

L'identificatore della chiave AWS Key Management Service (AWS KMS) da utilizzare per la crittografia Amazon EBS. Se `kmsKeyId` viene specificata, lo stato crittografato deve essere `true`.

La chiave specificata utilizzando questo parametro sostituisce la chiave KMS predefinita di Amazon EBS o qualsiasi chiave KMS a livello di cluster per la crittografia dello storage gestito di Amazon ECS che potresti aver specificato. Per ulteriori informazioni, consulta [Crittografa i dati archiviati nei volumi Amazon EBS collegati alle attività di Amazon ECS](#).

Puoi specificare la chiave KMS utilizzando uno dei seguenti metodi:

- ID chiave: ad esempio, `1234abcd-12ab-34cd-56ef-1234567890ab`.
- Alias chiave: ad esempio, `alias/ExampleAlias`.
- ARN chiave: ad esempio, `arn:aws:kms:us-east-1:012345678910:key/1234abcd-12ab-34cd-56ef-1234567890ab`
- Alias ARN: ad esempio, `arn:aws:kms:us-east-1:012345678910:alias/ExampleAlias`

 Important

AWS autentica la chiave KMS in modo asincrono. Pertanto, se si specifica un ID, un alias o un ARN non valido, l'azione può sembrare riuscita, ma alla fine non riesce. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi ai volumi allegati di Amazon EBS](#).

volumeType

Tipo: string

Campo obbligatorio: no

Valori validi: gp2 | gp3 | io1 | | io2 | sc1 | st1 standard

Il tipo di volume Amazon EBS. Per ulteriori informazioni sui tipi di volume, consulta i tipi di [volume di Amazon EBS](#) nella Amazon EBS User Guide. Il tipo di volume di default è gp3.

 Note

Il tipo di standard volume non è supportato per le attività Fargate.

sizeInGiB

Tipo: integer

Campo obbligatorio: no

Intervallo valido: numeri interi compresi tra 1 e 16.384

La dimensione del volume EBS in gibibyte (GiB). Se non fornisci un ID snapshot per configurare un volume per gli allegati, devi fornire un valore di dimensione. Se si configura un volume per l'allegato utilizzando un'istantanea, il valore predefinito è la dimensione dell'istantanea. È quindi possibile specificare una dimensione maggiore o uguale alla dimensione dell'istantanea.

Per gp2 i tipi di gp3 volume, l'intervallo valido è 1-16.384.

Per i tipi io1 di io2 volume, l'intervallo valido è 4-16.384.

Per i tipi `st1` di `sc1` volume, l'intervallo valido è 125-16.384.

Per il tipo di `standard` volume, l'intervallo valido è 1-1.024.

`snapshotId`

Tipo: `string`

Campo obbligatorio: no

L'ID dello snapshot di un volume Amazon EBS esistente che Amazon ECS utilizza per creare nuovi volumi da allegare. È necessario specificare `snapshotId` o `sizeInGiB`.

`volumeInitializationRate`

Tipo: `integer`

Campo obbligatorio: no

La velocità, in MiB/s, con cui i dati vengono recuperati da uno snapshot di un volume Amazon EBS esistente per creare nuovi volumi da allegare. Questa proprietà può essere specificata solo se si specifica un. `snapshotId` Per ulteriori informazioni su questo tasso di inizializzazione del volume, incluso l'intervallo di frequenze supportate per l'inizializzazione, consulta [Initialize Amazon EBS Volumes nella Amazon EBS User Guide](#).

`iops`

Tipo: `integer`

Campo obbligatorio: no

Il numero di operazioni I/O al secondo (IOPS). Per volumi `gp3`, `io1` e `io2`, questo rappresenta il numero di IOPS che sono assegnate al volume. Per `gp2` i volumi, questo valore rappresenta le prestazioni di base del volume e la velocità con cui il volume accumula crediti di I/O per il bursting. Questo parametro è obbligatorio per i volumi `io1` e `io2`. Questo parametro non è supportato per `gp2`, `st1`, `sc1` o volumi. `standard`

Per `gp3` i volumi, l'intervallo di valori valido è compreso tra 3.000 e 16.000.

Per `io1` i volumi, l'intervallo di valori valido è compreso tra 100 e 64.000.

Per `io2` i volumi, l'intervallo di valori valido è compreso tra 100 e 64.000.

`throughput`

Tipo: `integer`

Campo obbligatorio: no

La velocità effettiva necessaria per il provisioning dei volumi collegati alle attività gestite da un servizio.



Important

Questo parametro è supportato solo per i gp3 volumi.

roleArn

Tipo: stringa

Campo obbligatorio: sì

Il ruolo Amazon Resource ARN (ARN) dell'infrastruttura AWS Identity and Access Management (IAM) che fornisce le autorizzazioni Amazon ECS per gestire le risorse Amazon EBS per le tue attività. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

tagSpecifications

Tipo: oggetto

Campo obbligatorio: no

La specifica per i tag da applicare a ciascun volume Amazon EBS.

resourceType

Tipo: stringa

Campo obbligatorio: sì

Valori validi: volume

Il tipo di risorsa cui applicare tag al momento della creazione.

tags

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati che applichi ai volumi per aiutarti a classificarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi definiti dall'utente. `AmazonECSCreated` e `AmazonECSManaged` sono tag riservati che vengono aggiunti da Amazon ECS per tuo conto, quindi puoi specificare un massimo di 48 tag personalizzati. Quando un volume viene eliminato, vengono eliminati anche i tag. Per ulteriori informazioni, consulta [Taggare le risorse Amazon ECS](#).

`key`

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

`value`

Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

La parte opzionale di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

`propagateTags`

Tipo: stringa

Valori validi: `TASK_DEFINITION` | `SERVICE` | `NONE`

Campo obbligatorio: no

Specifica se copiare i tag dalla definizione dell'attività o dal servizio in un volume. Se `NONE` è specificato o non viene specificato alcun valore, i tag non vengono copiati.

`fileSystemType`

Tipo: string

Campo obbligatorio: no

Valori validi: `xfs` | `ext3` | `ext4` NTFS

Il tipo di file system su un volume. Il tipo di file system del volume determina il modo in cui i dati vengono archiviati e recuperati nel volume. Per i volumi creati da un'istantanea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea. Se il tipo di file system non corrisponde, l'operazione non verrà avviata.

I valori validi per Linux sono `ext3`, `xfs`, and `ext4`. L'impostazione predefinita per i volumi collegati alle attività di Linux è `XFS`.

I valori validi per Windows sono `NTFS`. L'impostazione predefinita per i volumi collegati alle attività di Windows è `NTFS`.

Modello di definizione del servizio

Di seguito viene illustrata la rappresentazione JSON di una definizione di servizio Amazon ECS.

Tipo di EC2 lancio di Amazon

```
{
  "cluster": "",
  "serviceName": "",
  "taskDefinition": "",
  "loadBalancers": [
    {
      "targetGroupArn": "",
      "loadBalancerName": "",
      "containerName": "",
      "containerPort": 0
    }
  ],
  "serviceRegistries": [
    {
      "registryArn": "",
      "port": 0,
      "containerName": "",
      "containerPort": 0
    }
  ],
}
```

```
"desiredCount": 0,
"clientToken": "",
"launchType": "EC2",
"capacityProviderStrategy": [
  {
    "capacityProvider": "",
    "weight": 0,
    "base": 0
  }
],
"platformVersion": "",
"role": "",
"deploymentConfiguration": {
  "deploymentCircuitBreaker": {
    "enable": true,
    "rollback": true
  },
  "maximumPercent": 0,
  "minimumHealthyPercent": 0,
  "alarms": {
    "alarmNames": [
      ""
    ],
    "enable": true,
    "rollback": true
  }
},
"placementConstraints": [
  {
    "type": "distinctInstance",
    "expression": ""
  }
],
"placementStrategy": [
  {
    "type": "binpack",
    "field": ""
  }
],
"networkConfiguration": {
  "awsvpcConfiguration": {
    "subnets": [
      ""
    ],
```

```

        "securityGroups": [
            ""
        ],
        "assignPublicIp": "DISABLED"
    }
},
"healthCheckGracePeriodSeconds": 0,
"schedulingStrategy": "REPLICA",
"deploymentController": {
    "type": "EXTERNAL"
},
"tags": [
    {
        "key": "",
        "value": ""
    }
],
"enableECSTags": true,
"propagateTags": "TASK_DEFINITION",
"enableExecuteCommand": true,
"availabilityZoneRebalancing": "ENABLED",
"serviceConnectConfiguration": {
    "enabled": true,
    "namespace": "",
    "services": [
        {
            "portName": "",
            "discoveryName": "",
            "clientAliases": [
                {
                    "port": 0,
                    "dnsName": ""
                }
            ],
            "ingressPortOverride": 0
        }
    ],
    "logConfiguration": {
        "logDriver": "journald",
        "options": {
            "KeyName": ""
        },
        "secretOptions": [
            {

```

```

        "name": "",
        "valueFrom": ""
    }
]
}
},
"volumeConfigurations": [
    {
        "name": "",
        "managedEBSVolume": {
            "encrypted": true,
            "kmsKeyId": "",
            "volumeType": "",
            "sizeInGiB": 0,
            "snapshotId": "",
            "volumeInitializationRate": 0,
            "iops": 0,
            "throughput": 0,
            "tagSpecifications": [
                {
                    "resourceType": "volume",
                    "tags": [
                        {
                            "key": "",
                            "value": ""
                        }
                    ],
                    "propagateTags": "NONE"
                }
            ],
            "roleArn": "",
            "filesystemType": ""
        }
    }
]
}

```

Tipo di avvio di Fargate

```

{
    "cluster": "",
    "serviceName": "",
    "taskDefinition": "",

```

```

"loadBalancers": [
  {
    "targetGroupArn": "",
    "loadBalancerName": "",
    "containerName": "",
    "containerPort": 0
  }
],
"serviceRegistries": [
  {
    "registryArn": "",
    "port": 0,
    "containerName": "",
    "containerPort": 0
  }
],
"desiredCount": 0,
"clientToken": "",
"launchType": "FARGATE",
"capacityProviderStrategy": [
  {
    "capacityProvider": "",
    "weight": 0,
    "base": 0
  }
],
"platformVersion": "",
"platformFamily": "",
"role": "",
"deploymentConfiguration": {
  "deploymentCircuitBreaker": {
    "enable": true,
    "rollback": true
  },
  "maximumPercent": 0,
  "minimumHealthyPercent": 0,
  "alarms": {
    "alarmNames": [
      ""
    ],
    "enable": true,
    "rollback": true
  }
},
},

```

```

"placementStrategy": [
  {
    "type": "binpack",
    "field": ""
  }
],
"networkConfiguration": {
  "awsvpcConfiguration": {
    "subnets": [
      ""
    ],
    "securityGroups": [
      ""
    ],
    "assignPublicIp": "DISABLED"
  }
},
"healthCheckGracePeriodSeconds": 0,
"schedulingStrategy": "REPLICA",
"deploymentController": {
  "type": "EXTERNAL"
},
"tags": [
  {
    "key": "",
    "value": ""
  }
],
"enableECSTags": true,
"propagateTags": "TASK_DEFINITION",
"enableExecuteCommand": true,
"availabilityZoneRebalancing": "ENABLED",
"serviceConnectConfiguration": {
  "enabled": true,
  "namespace": "",
  "services": [
    {
      "portName": "",
      "discoveryName": "",
      "clientAliases": [
        {
          "port": 0,
          "dnsName": ""
        }
      ]
    }
  ]
}

```

```

        ],
        "ingressPortOverride": 0
    }
],
"logConfiguration": {
    "logDriver": "journald",
    "options": {
        "KeyName": ""
    },
    "secretOptions": [
        {
            "name": "",
            "valueFrom": ""
        }
    ]
},
"volumeConfigurations": [
    {
        "name": "",
        "managedEBSVolume": {
            "encrypted": true,
            "kmsKeyId": "",
            "volumeType": "",
            "sizeInGiB": 0,
            "snapshotId": "",
            "volumeInitializationRate": 0,
            "iops": 0,
            "throughput": 0,
            "tagSpecifications": [
                {
                    "resourceType": "volume",
                    "tags": [
                        {
                            "key": "",
                            "value": ""
                        }
                    ]
                }
            ],
            "propagateTags": "NONE"
        },
        "roleArn": "",
        "filesystemType": ""
    }
]
}

```



```
    }  
  ]  
}
```

È possibile creare questo modello di definizione del servizio utilizzando il comando AWS CLI seguente.

```
aws ecs create-service --generate-cli-skeleton
```

Taggare le risorse Amazon ECS

Per semplificare la gestione delle risorse Amazon ECS, è possibile assegnare metadati personalizzati a ciascuna risorsa sotto forma di tag. Ciascun tag è formato da una chiave e da un valore facoltativo.

Puoi utilizzare i tag per categorizzare le risorse Amazon ECS in vari modi, ad esempio per scopo, proprietario o ambiente. Questa funzione è utile quando si dispone di numerose risorse dello stesso tipo. Puoi identificare velocemente una risorsa specifica in base ai tag a questa assegnati. Ad esempio, puoi definire un set di tag per le istanze del container Amazon ECS del tuo account. Questo ti aiuta a monitorare il proprietario dell'istanza e il livello dello stack.

Puoi utilizzare i tag per i report su costi e utilizzo. Puoi utilizzare questi report per analizzare il costo e l'utilizzo delle tue risorse di Amazon ECS. Per ulteriori informazioni, consulta [the section called "Report di utilizzo di"](#).

Warning

Ce ne sono molti APIs che restituiscono le chiavi dei tag e i loro valori. Negare l'accesso a `DescribeTags` non nega automaticamente l'accesso ai tag restituiti da altri. APIs Come da best practice, è consigliabile non includere dati sensibili nei tag.

Ti consigliamo di creare un set di chiavi di tag in grado di soddisfare i requisiti di ciascun tipo di risorsa. Tramite un set di chiavi di tag coerente la gestione delle risorse risulta notevolmente semplificata. Puoi cercare e filtrare le risorse in base ai tag aggiunti.

I tag non hanno alcun significato semantico per Amazon ECS e vengono interpretati rigorosamente come una stringa di caratteri. Puoi modificare chiavi e valori di tag e rimuovere tag da una risorsa in qualsiasi momento. È possibile impostare il valore di un tag su una stringa vuota, ma non su null. Se a una risorsa specifica aggiungi un tag con la stessa chiave di un tag esistente, il nuovo valore sovrascriverà quello precedente. Quando si elimina una risorsa, vengono eliminati anche tutti i tag relativi alla risorsa.

Se utilizzi AWS Identity and Access Management (IAM), puoi controllare quali utenti del tuo AWS account sono autorizzati a gestire i tag.

Assegnazione di tag alle risorse

Esistono diversi modi per assegnare tag ad attività, servizi, definizioni di attività e cluster Amazon ECS:

- Un utente tagga manualmente una risorsa utilizzando l' AWS Management Console API Amazon ECS AWS CLI, o un AWS SDK.
- Un utente crea un servizio o esegue un'attività autonoma e seleziona l'opzione dei tag gestiti da Amazon ECS.

Amazon ECS assegna automaticamente i tag a tutte le attività appena avviate. Per ulteriori informazioni, consulta [the section called “Tag gestiti da Amazon ECS”](#).

- Un utente crea una risorsa utilizzando la console. La console assegna automaticamente i tag alle risorse.

Questi tag vengono restituiti nelle AWS CLI risposte e AWS SDK e vengono visualizzati nella console. Non è possibile modificare o eliminare questi tag.

Per informazioni sui tag aggiunti, consulta la colonna Tag aggiunti automaticamente dalla console nella tabella Supporto del tagging per le risorse Amazon ECS.

Se specifichi dei tag quando crei una risorsa e i tag non possono essere applicati, Amazon ECS esegue il rollback del processo di creazione. Ciò fa sì che le risorse vengano create con i tag oppure che non vengano create affatto, nonché che nessuna risorsa sia mai sprovvista di tag. L'aggiunta di tag alle risorse in fase di creazione consente di evitare di eseguire script di tagging personalizzati dopo la creazione delle risorse.

La tabella seguente descrive le risorse Amazon ECS che supportano il tagging.

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Attività di Amazon ECS	Sì	Sì, dalla definizione di attività.	Chiave: <code>aws:ecs:clusterName</code> Value (Valore): <code>cluster-name</code>

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Servizi Amazon ECS	Sì	Sì, dalla definizione di attività o dal servizio alle attività del servizio.	Chiave: <code>ecs:service:stackId</code> Value <code>arn:aws:cloudformation: <i>arn</i></code>
Set di processi di Amazon ECS	Sì	No	N/D
Definizione dei processi di Amazon ECS	Sì	No	Chiave: <code>ecs:taskDefinition:createdFrom</code> Value (Valore): <code>ecs-console-v2</code>

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Cluster Amazon ECS	Sì	No	Chiave: <code>aws:cloudformation:logical-id</code> Value (Valore): <code>ECSCluster</code> Chiave: <code>aws:cloudformation:stack-id</code> Value (Valore): <code>arn:aws:cloudformation: <i>arn</i></code> Chiave: <code>aws:cloudformation:stack-name</code> Value (Valore): <code>ECS-Console-V2-Cluster- <i>EXAMPLE</i></code>
Istanze di container di Amazon ECS	Sì	Sì, dall' EC2 istanza Amazon. Per ulteriori informazioni, consulta Aggiungere tag a un'istanza di container Amazon ECS .	N/D
Istanze esterne di Amazon ECS	Sì	No	N/D

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Provider di capacità Amazon ECS	Sì. Non è possibile applicare tag ai provider di capacità predefiniti FARGATE e FARGATE_SPOT .	No	N/D

Applicazione di tag alle risorse durante la creazione

Le seguenti risorse supportano l'etichettatura alla creazione utilizzando l'API Amazon ECS o AWS CLI AWS SDK:

- Attività di Amazon ECS
- Servizi Amazon ECS
- Definizione dell'attività di Amazon ECS
- Set di processi di Amazon ECS
- Cluster Amazon ECS
- Istanze di container di Amazon ECS
- Provider di capacità Amazon ECS

Amazon ECS offre la possibilità di utilizzare l'autorizzazione all'assegnazione di tag per la creazione di risorse. Quando Account AWS è configurato per l'autorizzazione all'etichettatura, gli utenti devono disporre delle autorizzazioni per le azioni che creano la risorsa, ad esempio. `ecsCreateCluster`. Se specifichi i tag nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare se gli utenti o i ruoli dispongono delle autorizzazioni per creare tag. Pertanto, devi concedere le autorizzazioni esplicite per utilizzare l'operazione `ecs:TagResource`. Per ulteriori informazioni, consulta [the section called “Assegnazione di tag alle risorse al momento della creazione”](#). Per informazioni su come configurare l'opzione, consulta. [the section called “Autorizzazione all'assegnazione di tag”](#)

Restrizioni

Ai tag si applicano le limitazioni seguenti:

- È possibile associare un massimo di 50 tag a una risorsa.
- Le chiavi dei tag non possono essere ripetute per una risorsa. Ogni chiave di tag deve essere univoca e può avere un solo valore.
- Le chiavi di tag possono contenere fino a 128 caratteri in UTF-8.
- Ogni valore può contenere fino a 256 caratteri UTF-8.
- Se più risorse Servizi AWS e utilizzano lo schema di etichettatura, limita i tipi di caratteri che usi. Alcuni servizi potrebbero avere restrizioni sui caratteri consentiti. I caratteri generalmente consentiti sono lettere, numeri, spazi e i simboli seguenti: `+ - = . _ : / @`.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Non è possibile utilizzare `aws:`, `AWS:` o qualsiasi combinazione di maiuscole o minuscole di un tale prefisso per chiavi o valori. Questi sono riservati solo all' AWS uso. Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non rientrano nel tuo tags-per-resource limite.

Tag gestiti da Amazon ECS

Quando utilizzi tag gestiti da Amazon ECS, Amazon ECS etichetta automaticamente tutte le attività appena avviate e tutti i volumi Amazon EBS collegati alle attività con le informazioni sul cluster e i tag di definizione delle attività aggiunti dall'utente o i tag di servizio. Di seguito vengono descritti i tag aggiunti:

- Attività autonome: un tag con una Chiave come `aws:ecs:clusterName` e un Valore impostato sul nome del cluster. Tutti i tag di definizione delle attività aggiunti dagli utenti. Un volume Amazon EBS collegato a un'attività autonoma riceverà il tag con una chiave come `aws:ecs:clusterName` e un valore impostato sul nome del cluster. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi [Amazon EBS](#).
- Attività che fanno parte di un servizio: un tag con una Chiave come `aws:ecs:clusterName` e un Valore impostato sul nome del cluster. Un tag con una Chiave come `aws:ecs:serviceName` e un Valore impostato sul nome del servizio. Tag da una delle risorse seguenti:
 - Definizioni delle attività: tutti i tag di definizione delle attività aggiunti dagli utenti.
 - Servizi: tutti i tag di servizio aggiunti dagli utenti.

Un volume Amazon EBS collegato a un'attività che fa parte di un servizio riceverà un tag con Key as `aws:ecs:clusterName` e Value impostati sul nome del cluster e un tag con Key as `aws:ecs:serviceName` e Value impostato sul nome del servizio. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi [Amazon EBS](#).

Per questa funzionalità sono richieste le seguenti opzioni:

- Devi scegliere esplicitamente i nuovi formati per il nome della risorsa Amazon (ARN) e l'identificatore di risorsa (ID). Per ulteriori informazioni, consulta [Nomi di risorse Amazon \(ARNs\) e IDs](#).
- Quando utilizzi il APIs per creare un servizio o eseguire un'attività, devi `enableECSTags` impostarlo su `for e. true run-task create-service` Per ulteriori informazioni, consulta [create-service](#) e [run-task](#) nella Documentazione di riferimento delle API AWS Command Line Interface .
- Amazon ECS utilizza tag gestiti per determinare quando alcune funzionalità sono abilitate, ad esempio il dimensionamento automatico del cluster. Ti consigliamo di non modificare manualmente i tag in modo che Amazon ECS possa gestire efficacemente le funzionalità.

Utilizza i tag per la fatturazione

AWS fornisce uno strumento di reporting chiamato Cost Explorer che puoi utilizzare per analizzare il costo e l'utilizzo delle tue risorse Amazon ECS.

Utilizza Cost Explorer per visualizzare i grafici relativi all'utilizzo e ai costi. Puoi visualizzare i dati fino agli ultimi 13 mesi e prevedere le spese dei prossimi tre mesi. Puoi utilizzare Cost Explorer per visualizzare i modelli relativi a quanto spendi in risorse AWS nel tempo. Ad esempio, puoi utilizzarlo per identificare aree che richiedono ulteriore studio e visualizzare le tendenze che puoi utilizzare per comprendere i costi. Puoi anche specificare intervalli di tempo per i dati e visualizzare i dati temporali per mese o per giorno.

Puoi utilizzare i tag gestiti da Amazon ECS oppure tag aggiunti dall'utente per il tuo report su costi e utilizzo. Per ulteriori informazioni, consulta [Report di utilizzo di Amazon ECS](#).

Per visualizzare il costo delle risorse combinate, puoi organizzare le informazioni di fatturazione in base alle risorse con gli stessi valori di chiave di tag. Puoi ad esempio applicare tag a numerose risorse con un nome di applicazione specifico, quindi organizzare le informazioni di fatturazione per visualizzare il costo totale dell'applicazione in più servizi. Per ulteriori informazioni sulla

configurazione di un report di allocazione dei costi mediante i tag, consulta [Report di allocazione dei costi mensili](#) nella Guida per l'utente di AWS Billing .

Inoltre, puoi attivare Dati di ripartizione dei costi per inserire i dati sull'utilizzo della CPU e della memoria a livello di attività in Report di costi utilizzo. Per ulteriori informazioni, consulta [Report di costi e utilizzo a livello di attività](#).

Note

Se hai attivato il report, i dati relativi al mese corrente saranno disponibili per la visualizzazione entro 24 ore.

Aggiungere tag alle risorse Amazon ECS

Puoi taggare attività, servizi, definizioni di attività o cluster nuovi o esistenti. Per informazioni sull'etichettatura delle istanze del contenitore, consulta. [Aggiungere tag a un'istanza di container Amazon ECS](#)

Warning

Non aggiungere Informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei tag. I tag sono accessibili a molti AWS servizi, inclusa la fatturazione. I tag non sono destinati ad essere utilizzati per dati privati o sensibili.

Le risorse seguenti consentono di specificare i tag al momento della creazione della risorsa.

Attività	Console	AWS CLI	Operazione API
Esegui una o più operazioni.	Esecuzione di un'applicazione come attività Amazon ECS	run-task	RunTask
Crea un servizio.	Creazione di un servizio Amazon	create-service	CreateService

Attività	Console	AWS CLI	Operazione API
	ECS utilizzando la console		
Crea un set di attività.	Implementa i servizi Amazon ECS utilizzando un controller di terze parti	create-task-set	CreateTaskSet
Registra una definizione di attività.	the section called “Creazione di una definizione di attività attraverso la nuova console”	register-task-definition	RegisterTaskDefinition
Crea un cluster.	Creazione di un cluster Amazon ECS per il tipo di lancio Fargate	create-cluster	CreateCluster
Esegui una o più istanze di container.	Avvio di un'istanza di container Linux di Amazon ECS	run-instances	RunInstances

Aggiungere tag alle risorse esistenti (console Amazon ECS)

Puoi aggiungere o eliminare tag associati ai cluster, ai servizi, alle attività e alle definizioni delle attività direttamente dalla pagina della risorsa.

Modifica di un tag per una singola risorsa

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel riquadro di navigazione, seleziona un tipo di risorsa (ad esempio Cluster).

4. Seleziona la risorsa dall'elenco di risorse, scegli la scheda Tags (Tag), quindi seleziona Manage tags (Gestisci tag).
5. Configura i tag.

[Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:

- In Chiave, immetti il nome della chiave.
- In Valore, immetti il valore della chiave.

6. Scegli Save (Salva).

Aggiungere tag alle risorse esistenti (AWS CLI)

Puoi aggiungere o sovrascrivere uno o più tag utilizzando AWS CLI o un'API.

Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

- AWS CLI - [tag-risorsa](#)
- Azione API - [TagResource](#)

Aggiungere tag a un'istanza di container Amazon ECS

Puoi associare i tag alle istanze di container utilizzando uno dei seguenti metodi:

- Metodo 1: quando crei l'istanza del contenitore utilizzando l' EC2API, la CLI o la console Amazon, specifica i tag passando i dati utente all'istanza utilizzando il parametro di configurazione dell'agente contenitore. ECS_CONTAINER_INSTANCE_TAGS In questo modo vengono creati tag associati all'istanza del contenitore solo in Amazon ECS, che non possono essere elencati utilizzando l' EC2 API di Amazon. Per ulteriori informazioni, consulta [Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati](#).

⚠ Important

Se avvii le istanze di container utilizzando un gruppo Amazon EC2 Auto Scaling, dovresti usare il parametro di configurazione dell'agente ECS_CONTAINER_INSTANCE_TAGS per aggiungere tag. Ciò è dovuto al modo in cui i tag vengono aggiunti alle EC2 istanze Amazon avviate utilizzando i gruppi di Auto Scaling.

Di seguito è riportato un esempio di script di dati utente che associa i tag all'istanza di container:

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_CONTAINER_INSTANCE_TAGS={"tag_key": "tag_value"}
EOF
```

- Metodo 2: quando crei l'istanza del contenitore utilizzando l' EC2 API, la CLI o la console di Amazon, specifica innanzitutto i tag utilizzando il TagSpecification.N parametro. Quindi, passa i dati utente all'istanza utilizzando il parametro di configurazione dell'agente container ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM. In questo modo li propaga da Amazon EC2 ad Amazon ECS.

Di seguito è riportato un esempio di script di dati utente che propaga i tag associati a un' EC2 istanza Amazon e registra l'istanza con un cluster denominato. MyCluster

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM=ec2_instance
EOF
```

Per consentire l'accesso e consentire la propagazione dei tag delle istanze di container da Amazon EC2 ad Amazon ECS, aggiungi manualmente le seguenti autorizzazioni come policy in linea al ruolo IAM dell'istanza di container Amazon ECS. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- ec2:DescribeTags

Di seguito è riportata una policy di esempio utilizzata per aggiungere tali autorizzazioni.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeTags"
      ],
      "Resource": "*"
    }
  ]
}
```

Istanze di container esterni

Puoi associare i tag alle istanze di container esterne utilizzando uno dei seguenti metodi:

- Metodo 1: prima di eseguire lo script di installazione per registrare l'istanza esterna nel cluster, crea o modifica il file di configurazione dell'agente del container di Amazon ECS all'indirizzo `/etc/ecs/ecs.config` e aggiungi il parametro di configurazione dell'agente del container `ECS_CONTAINER_INSTANCE_TAGS`. Questo consente di creare i tag associati all'istanza esterna.

Di seguito è riportato un esempio di sintassi.

```
ECS_CONTAINER_INSTANCE_TAGS={"tag_key": "tag_value"}
```

- Metodo 2: dopo aver registrato l'istanza esterna nel cluster, puoi utilizzarla AWS Management Console per aggiungere tag. Per ulteriori informazioni, consulta [Aggiungere tag alle risorse esistenti \(console Amazon ECS\)](#).

Report di utilizzo di Amazon ECS

AWS fornisce uno strumento di reporting chiamato Cost Explorer che puoi utilizzare per analizzare il costo e l'utilizzo delle tue risorse Amazon ECS.

Utilizza Cost Explorer per visualizzare i grafici relativi all'utilizzo e ai costi. Puoi visualizzare i dati fino agli ultimi 13 mesi e prevedere le spese dei prossimi tre mesi. Puoi utilizzare Cost Explorer per visualizzare i modelli relativi a quanto spendi in risorse AWS nel tempo. Ad esempio, puoi utilizzarlo

per identificare aree che richiedono ulteriore studio e visualizzare le tendenze che puoi utilizzare per comprendere i costi. Puoi anche specificare intervalli di tempo per i dati e visualizzare i dati temporali per mese o per giorno.

I dati di misurazione nel report su costi e utilizzo mostrano l'utilizzo in tutti i processi Amazon ECS. I dati di misurazione includono l'utilizzo della CPU come vCPU-Hours e l'utilizzo della memoria come GB-Hours per ogni attività eseguita. Come i dati presentati dipendono dal tipo di avvio del processo.

Per i processi che utilizzano il tipo di avvio Fargate, la colonna `lineItem/Operation` mostrerà `FargateTask` e verrà visualizzato il costo associato a ciascun processo.

Per le attività che utilizzano il tipo di EC2 avvio, la `lineItem/Operation` colonna viene visualizzata `ECSTask-EC2` e le attività non hanno un costo diretto associato. I dati di misurazione mostrati nel report, come l'utilizzo della memoria, rappresenta le risorse totali prenotate dal processo nel periodo di fatturazione indicato. Puoi utilizzare questi dati per determinare il costo del cluster sottostante di EC2 istanze Amazon. I dati su costi e utilizzo per le tue EC2 istanze Amazon sono elencati separatamente nel EC2 servizio Amazon.

Puoi anche utilizzare i tag gestiti di Amazon ECS per identificare il servizio o il cluster a cui appartengono i singoli processi. Per ulteriori informazioni, consulta [Utilizza i tag per la fatturazione](#).

 Important

I dati di misurazione possono essere visualizzati solo per le attività avviate dopo il 16 novembre 2018. I processi avviati prima di questa data non mostrano i dati di misurazione.

Di seguito è illustrato un esempio dei campi in base a cui puoi ordinare i dati di allocazione dei costi utilizzando lo strumento Cost Explorer.

- Nome cluster
- Nome servizio
- Tag delle risorse
- Tipo di avvio
- Regione AWS
- Tipo di utilizzo

Per ulteriori informazioni sulla creazione di un report sui AWS costi e sull'utilizzo, consulta il report [AWS sui costi e sull'utilizzo](#) nella Guida per l'AWS Billing utente.

Report di costi e utilizzo a livello di attività

AWS Cost Management può fornire dati sull'utilizzo della CPU e della memoria AWS Cost and Usage Report per ogni attività su Amazon ECS, incluse le attività su Fargate e le attività su EC2. Questi dati sono denominati Dati di ripartizione dei costi. Puoi utilizzare questi dati per analizzare i costi e l'utilizzo delle applicazioni. Inoltre, puoi suddividere e allocare i costi a singole unità aziendali e team utilizzando tag di allocazione dei costi e categorie di costi. Per ulteriori informazioni sui dati di allocazione dei costi suddivisi, consulta [Comprendere i dati di allocazione dei costi suddivisi nella Guida per l'utente](#). AWS Cost and Usage Report

Puoi attivare Dati di ripartizione dei costi a livello di attività per l'account nella AWS Cost Management Console. Se disponi di un account di gestione (pagante), puoi scegliere di applicare questa configurazione a tutti gli account collegati.

Dopo aver impostato i dati di allocazione dei costi suddivisi, ci saranno colonne aggiuntive sotto l'header del report. Per ulteriori informazioni, consulta i [dettagli degli elementi di riga suddivisi nella Guida](#) per l'AWS Cost and Usage Report utente.

Per le attività relative EC2, questi dati suddividono il costo dell'EC2 istanza in base all'utilizzo delle risorse o alle prenotazioni e alle risorse rimanenti sull'istanza.

I seguenti sono i prerequisiti:

- Imposta il parametro di configurazione dell'agente ECS_DISABLE_METRICS Amazon ECS su `false`.

Quando questa impostazione è impostata su `false`, l'agente Amazon ECS invia i parametri ad Amazon CloudWatch. Su Linux, questa impostazione è `false` predefinita e le metriche vengono inviate a CloudWatch. In Windows, questa impostazione è `true` predefinita, quindi è necessario modificarla in modo da `false` inviare le metriche CloudWatch per AWS Cost Management utilizzarle. Per ulteriori informazioni sulla configurazione dell'agente di ECS, consulta [Configurazione dell'agente del container Amazon ECS](#).

- La versione Docker minima per parametri affidabili è v20.10.13 e successive, inclusa nell'AMI 20220607 ottimizzata per Amazon ECS e successive.

Per utilizzare Dati di ripartizione dei costi, devi creare un report e selezionare Dati di ripartizione dei costi. Per ulteriori informazioni, consulta [Creazione di report sui costi e sull'utilizzo](#) nella Guida per l'AWS Cost and Usage Report utente.

AWS Cost Management calcola i dati di suddivisione dei costi in base all'utilizzo della CPU e della memoria dell'attività. AWS Cost Management può utilizzare la CPU e la prenotazione della memoria dell'attività anziché l'utilizzo, se l'utilizzo non è disponibile. Se vedi che il CUR sta utilizzando le prenotazioni, verifica che le istanze del contenitore soddisfino i prerequisiti e che le metriche sull'utilizzo delle risorse dell'attività vengano visualizzate in CloudWatch.

Monitoraggio di Amazon ECS

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon ECS e delle tue AWS soluzioni. È necessario raccogliere i dati di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. Prima di iniziare a monitorare Amazon ECS, crea un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno utilizzati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

I parametri resi disponibili dipendono dal tipo di avvio dei processi e dei servizi nei cluster. Se utilizzi il tipo di avvio Fargate per i tuoi servizi, vengono forniti i parametri di utilizzo della memoria e della CPU per semplificare il monitoraggio dei servizi. Per il tipo di EC2 lancio di Amazon, possiedi e devi monitorare le EC2 istanze che compongono l'infrastruttura sottostante. Ulteriori parametri di prenotazione e utilizzo della CPU e della memoria sono disponibili presso il cluster, il servizio e l'attività.

La fase successiva consiste nello stabilire una baseline per le prestazioni normali di Amazon ECS nell'ambiente, misurando le prestazioni in diversi momenti e con condizioni di carico differenti. Quando monitori Amazon ECS, archivia i dati storici di monitoraggio in modo da poterli confrontare con i dati sulle prestazioni correnti e per poter identificare i normali modelli di prestazioni e le anomalie e individuare i metodi per risolvere i problemi.

Per stabilire una baseline, è necessario monitorare almeno gli elementi seguenti:

- I parametri di utilizzo e prenotazione di CPU e memoria per i cluster Amazon ECS
- I parametri di utilizzo di CPU e memoria per i tuoi servizi Amazon ECS

Per ulteriori informazioni, consulta [Visualizzazione dei parametri Amazon ECS](#).

Best practice per il monitoraggio di Amazon ECS

Utilizza le seguenti best practice per monitorare Amazon ECS.

- Rendi il monitoraggio una priorità per risolvere piccoli problemi prima che diventino grandi
- Crea un piano di monitoraggio che includa le risposte alla seguente domanda
 - Quali sono gli obiettivi del monitoraggio?
 - Di quali risorse si intende eseguire il monitoraggio?
 - Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
 - Quali strumenti di monitoraggio verranno utilizzati?
 - Chi eseguirà i processi di monitoraggio?
 - Chi deve ricevere una notifica quando si verifica un problema?
- Automatizza il monitoraggio il più possibile.
- Controlla i file di registro di Amazon ECS. Per ulteriori informazioni, consulta [Visualizzazione dei log degli agenti container Amazon ECS](#).
- Usa Runtime Monitoring per proteggere account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente. Per ulteriori informazioni, consulta [Identifica i comportamenti non autorizzati utilizzando Runtime Monitoring](#).

Strumenti di monitoraggio per Amazon ECS

AWS fornisce diversi strumenti che puoi utilizzare per monitorare Amazon ECS. Alcuni di questi strumenti possono essere configurati in modo che eseguano automaticamente il monitoraggio, mentre altri richiedono l'intervento manuale. Si consiglia di automatizzare il più possibile i processi di monitoraggio.

Strumenti di monitoraggio automatici

Per controllare Amazon ECS e segnalare l'eventuale presenza di problemi, puoi usare i seguenti strumenti di monitoraggio automatici:

- CloudWatch Allarmi Amazon: monitora una singola metrica in un periodo di tempo specificato ed esegui una o più azioni in base al valore della metrica rispetto a una determinata soglia in diversi periodi di tempo. L'azione è una notifica inviata a un argomento di Amazon Simple Notification Service (Amazon SNS) o a una politica di Amazon Auto EC2 Scaling. CloudWatch gli allarmi non

richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta [Monitora Amazon ECS utilizzando CloudWatch](#).

Per i servizi con attività che utilizzano il tipo di avvio Fargate, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le attività del servizio in base a CloudWatch metriche, come l'utilizzo della CPU e della memoria. Per ulteriori informazioni, consulta [Ridimensiona automaticamente il tuo servizio Amazon ECS](#).

Per i cluster con attività o servizi che utilizzano il tipo di EC2 avvio, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le istanze del contenitore in base a CloudWatch metriche, come la prenotazione della memoria del cluster.

Per le istanze di container che sono state lanciate con l'AMI Amazon Linux ottimizzata per Amazon ECS, puoi CloudWatch usare Logs per visualizzare diversi log delle tue istanze di container in un'unica comoda posizione. È necessario installare l'agente sulle istanze del contenitore. CloudWatch Per ulteriori informazioni, consulta [Scaricare e configurare l' CloudWatch agente utilizzando la riga di comando](#) nella Amazon CloudWatch User Guide. Devi inoltre aggiungere la policy ECS-CloudWatchLogs al ruolo `ecsInstanceRole`. Per ulteriori informazioni, consulta [Monitoraggio delle autorizzazioni delle istanze dei container](#).

- Amazon CloudWatch Logs: monitora, archivia e accedi ai file di registro dai contenitori nelle tue attività Amazon ECS specificando il driver di `awslogs` registro nelle definizioni delle attività. Per ulteriori informazioni, consulta [Invia i log di Amazon ECS a CloudWatch](#).

Puoi anche monitorare, archiviare e accedere al sistema operativo e ai file di log dell'agente del container Amazon ECS dalle istanze di container Amazon ECS. Questo metodo di accesso ai log può essere utilizzato per i container che utilizzano il tipo di avvio EC2.

- Amazon CloudWatch Events: abbina gli eventi e li indirizza a una o più funzioni o stream di destinazione per apportare modifiche, acquisire informazioni sullo stato e intraprendere azioni correttive. Per ulteriori informazioni, [Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge](#) consulta questa guida ed [EventBridge è l'evoluzione di Amazon CloudWatch Events](#) nella Amazon EventBridge User Guide.
- Container Insights: raccogli, aggrega e riepiloga metriche e log dalle tue applicazioni e microservizi containerizzati. Container Insights raccoglie dati come eventi di log delle prestazioni tramite Embedded Metric Format. Questi eventi del registro delle prestazioni sono voci che utilizzano uno schema JSON strutturato che consente di importare e archiviare dati ad alta cardinalità su larga scala. Da questi dati, CloudWatch crea metriche aggregate a livello di cluster, attività e

servizio come metriche. CloudWatch Le metriche raccolte da Container Insights sono disponibili nei dashboard CloudWatch automatici e sono visualizzabili anche nella sezione Metriche della console. CloudWatch

- AWS CloudTrail monitoraggio dei log: condividi i file di registro tra account, monitora i file di CloudTrail registro in tempo reale inviandoli a CloudWatch Logs, scrivi applicazioni di elaborazione dei log in Java e verifica che i file di registro non siano cambiati dopo la consegna da parte di. CloudTrail Per ulteriori informazioni, [Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail](#) consultate questa guida e [Lavorare con i file di CloudTrail registro](#) nella Guida per l'AWS CloudTrail utente.
- Monitoraggio del runtime: rileva le minacce per cluster e contenitori all'interno del tuo AWS ambiente. Runtime Monitoring utilizza un agente di GuardDuty sicurezza che aggiunge visibilità di runtime ai singoli carichi di lavoro Amazon ECS, ad esempio accesso ai file, esecuzione dei processi e connessioni di rete.

Strumenti di monitoraggio manuali

Un'altra parte importante del monitoraggio di Amazon ECS consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Le dashboard della AWS console CloudWatch Trusted Advisor, e altre, forniscono una at-a-glance panoramica dello stato del tuo ambiente. AWS Ti consigliamo inoltre di controllare i file di log sulle tue istanze di container e i container nei tuoi processi.

- Console Amazon ECS:
 - Metriche del cluster per il tipo di avvio EC2
 - Parametri del servizio
 - Stato di integrità dei servizi
 - Eventi di distribuzione del servizio
- CloudWatch home page:
 - Stato e allarmi attuali
 - Grafici degli allarmi e delle risorse
 - Stato di integrità dei servizi

Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:

- Creare [pannelli di controllo personalizzati](#) per monitorare i servizi di interesse.
- Creare grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.

- Cerca e sfoglia tutte le metriche AWS delle tue risorse.
- Crea e modifica gli allarmi per ricevere le notifiche dei problemi.
- Controllo dello stato del contenitore: si tratta di comandi che vengono eseguiti localmente su un contenitore e convalidano lo stato e la disponibilità delle applicazioni. Li configuri per contenitore nella definizione dell'attività.
- AWS Trusted Advisor può aiutarvi a monitorare AWS le vostre risorse per migliorare prestazioni, affidabilità, sicurezza ed economicità. Quattro Trusted Advisor controlli sono disponibili per tutti gli utenti; più di 50 controlli sono disponibili per gli utenti con un piano di supporto Business o Enterprise. Per ulteriori informazioni, consulta [AWS Trusted Advisor](#).

Trusted Advisor dispone dei seguenti controlli relativi ad Amazon ECS:

- Una tolleranza agli errori che indica che un servizio è in esecuzione in un'unica zona di disponibilità.
- Una tolleranza agli errori che indica che non è stata utilizzata la strategia di spread placement per più zone di disponibilità.
- AWS Compute Optimizer è un servizio che analizza le metriche di configurazione e utilizzo delle risorse. AWS Il servizio segnala se le risorse sono ideali e genera suggerimenti di ottimizzazione per ridurre i costi e migliorare le prestazioni dei carichi di lavoro.

Per ulteriori informazioni, consulta [AWS Compute Optimizer consigli per Amazon ECS](#).

Monitora Amazon ECS utilizzando CloudWatch

Puoi monitorare le tue risorse Amazon ECS utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi da Amazon ECS in metriche leggibili quasi in tempo reale. Queste statistiche vengono registrate per un periodo di due settimane in modo da permettere l'accesso a informazioni storiche e per offrire una prospettiva migliore sulle prestazioni di cluster e servizi. I dati metrici di Amazon ECS vengono inviati automaticamente CloudWatch in periodi di 1 minuto. Per ulteriori informazioni CloudWatch, consulta la [Amazon CloudWatch User Guide](#).

Amazon ECS fornisce parametri gratuiti per cluster e servizi. A un costo aggiuntivo, puoi attivare Amazon ECS CloudWatch Container Insights per il tuo cluster per parametri per attività, tra cui l'utilizzo di CPU, memoria e file system EBS. Per ulteriori informazioni su Container Insights, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#).

Considerazioni

Quando si utilizzano i CloudWatch parametri di Amazon ECS, è necessario considerare quanto segue.

- Qualsiasi servizio Amazon ECS ospitato su Fargate CloudWatch dispone automaticamente di metriche di utilizzo della CPU e della memoria, quindi non è necessario eseguire alcuna operazione manuale.
- Per qualsiasi attività o servizio Amazon ECS ospitato su EC2 istanze Amazon, l' EC2 istanza Amazon richiede una versione 1.4.0 o successiva (Linux) 1.0.0 o successiva (Windows) dell'agente container per generare le CloudWatch metriche. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#).
- La versione minima di Docker per CloudWatch parametri affidabili è la versione Docker e successive. 20.10.13
- Le tue EC2 istanze Amazon richiedono anche l'`ecs:StartTelemetrySession` autorizzazione per il ruolo IAM con cui avvii le tue EC2 istanze Amazon. Se hai creato il ruolo IAM dell'istanza del contenitore Amazon ECS prima che i CloudWatch parametri fossero disponibili per Amazon ECS, potresti dover aggiungere questa autorizzazione. Per informazioni sul ruolo IAM dell'istanza di contenitore e sull'associazione della politica IAM gestita per le istanze di container, consulta. [Ruolo IAM delle istanze di container Amazon ECS](#)
- Puoi disabilitare la raccolta di CloudWatch metriche sulle tue EC2 istanze Amazon ECS_DISABLE_METRICS=true impostando la configurazione dell'agente container Amazon ECS. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Parametri consigliati

Amazon ECS fornisce CloudWatch parametri gratuiti che puoi utilizzare per monitorare le tue risorse. La prenotazione di CPU e memoria e l'utilizzo di CPU, memoria e file system EBS nell'intero cluster e l'utilizzo di CPU, memoria e file system EBS sui servizi dei cluster possono essere misurati utilizzando questi parametri. Per i carichi di lavoro GPU, puoi misurare la prenotazione GPU nel cluster.

L'infrastruttura su cui sono ospitate le tue attività Amazon ECS nei tuoi cluster determina quali metriche sono disponibili. Per le attività ospitate sull'infrastruttura Fargate, Amazon ECS fornisce

parametri di utilizzo di CPU, memoria e file system EBS per facilitare il monitoraggio dei tuoi servizi. Per le attività ospitate su EC2 istanze, Amazon ECS fornisce metriche di prenotazione di CPU, memoria e GPU e metriche di utilizzo di CPU e memoria a livello di cluster e servizio. È necessario monitorare separatamente le EC2 istanze Amazon che costituiscono l'infrastruttura sottostante. Per ulteriori informazioni sul monitoraggio delle EC2 istanze Amazon, consulta [Monitoring Amazon EC2 nella Amazon EC2 User Guide](#).

Per informazioni sugli allarmi consigliati da utilizzare con Amazon ECS, consulta uno dei seguenti articoli nella Amazon CloudWatch Logs User Guide:

- [Amazon ECS](#)
- [Amazon ECS con Container Insights](#)

Visualizzazione dei parametri Amazon ECS

Una volta che le risorse sono in esecuzione nel cluster, puoi visualizzare i parametri su Amazon ECS e CloudWatch sulle console. La console Amazon ECS fornisce una visualizzazione massima, minima e media di 24 ore dei parametri del cluster e del servizio. La console CloudWatch fornisce una visualizzazione granulare e personalizzabile delle risorse, nonché del numero di attività in esecuzione in un servizio.

Console Amazon ECS

I parametri di utilizzo di CPU e memoria del servizio Amazon ECS sono disponibili nella console Amazon ECS. La visualizzazione offerta per i parametri del servizio mostra i valori medio, minimo e massimo per il precedente periodo di 24 ore con punti di dati disponibili a intervalli di 5 minuti. Per ulteriori informazioni, consulta [Metriche di utilizzo del servizio Amazon ECS](#).

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Seleziona il cluster di cui desideri visualizzare i parametri.
3. Determina le metriche da visualizzare.

Per visualizzare le metriche da	Fasi	
Cluster	Nella pagina dei dettagli del cluster, scegli la scheda	

Per visualizzare le metriche da	Fasi	
	Metriche. È inoltre disponibili un collegamento alla CloudWatch console per visualizzare le metriche di CloudWatch Container Insights, se le hai attivate.	
Servizi	Nella pagina dei dettagli del cluster, nella scheda Servizi, seleziona il servizio. Le metriche sono quindi disponibili nella scheda Health and metrics.	

CloudWatch console

Per il tipo di lancio di Fargate, i parametri del servizio Amazon ECS possono essere visualizzati anche sulla console. CloudWatch La console fornisce la vista più dettagliata dei parametri Amazon ECS e puoi personalizzare le visualizzazioni in base alle esigenze. È possibile visualizzare l'utilizzo del servizio e il conteggio delle attività del servizio IN ESECUZIONE.

Per il tipo di EC2 avvio, i parametri del cluster e del servizio Amazon ECS possono essere visualizzati anche sulla CloudWatch console. La console fornisce la vista più dettagliata dei parametri Amazon ECS e puoi personalizzare le visualizzazioni in base alle esigenze.

Per informazioni su come visualizzare le metriche, consulta [Visualizza i parametri disponibili](#) nella Amazon CloudWatch User Guide.

Metriche di Amazon ECS CloudWatch

Puoi utilizzare i parametri di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse del tuo account. Utilizza queste metriche per visualizzare l'utilizzo corrente del servizio su CloudWatch grafici e dashboard.

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
CPUReservation	La percentuale di unità CPU riservate nel cluster o nel servizio. La prenotazione della CPU (filtrata per <code>ClusterName</code>) viene misurata come il totale delle unità CPU riservate dalle attività di Amazon ECS sul cluster, diviso per le unità CPU totali per tutte le EC2 istanze Amazon registrate nel cluster. Solo EC2 le istanze Amazon con DRAINING stato ACTIVE o influiranno sui parametri di prenotazione della CPU. La metrica è supportata solo per le attività	<code>ClusterName</code> .	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	ospitate su un' EC2 istanza Amazon.			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
CPUUtilization	La percentuale di unità CPU utilizzata dal cluster o dal servizio. L'utilizzo della CPU a livello di cluster (filtrato per <code>ClusterName</code>) viene misurato come il totale delle unità CPU utilizzate dalle attività di Amazon ECS sul cluster, diviso per le unità CPU totali per tutte le EC2 istanze Amazon registrate nel cluster. Solo le EC2 istanze Amazon con DRAINING stato ACTIVE o influiranno sui parametri di prenotazione della CPU. La metrica a livello di cluster è supportata solo per le	<code>ClusterName</code> , <code>ServiceName</code>	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	attività ospitate su un'istanza Amazon. EC2 L'utilizzo della CPU a livello di servizio (filtrato per <code>ClusterName</code>, <code>ServiceName</code>) viene misurato come il totale delle unità CPU utilizzate dalle attività che appartengono al servizio, diviso per il numero totale di unità CPU riservate alle attività che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su EC2 istanze Amazon e Fargate.			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
MemoryReservation	La percentuale di memoria che viene riservata dai processi in esecuzione nel cluster. La prenotazione della memoria del cluster viene misurata come la memoria totale riservata dalle attività di Amazon ECS sul cluster, divisa per la quantità totale di memoria per tutte le EC2 istanze Amazon registrate e nel cluster. Questa metrica può essere filtrata solo per. ClusterName Solo EC2 le istanze Amazon in ACTIVE o in DRAINING stato influiranno sui parametri di prenotazione della memoria.	ClusterName	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	La metrica di prenotazione della memoria a livello di cluster è supportata solo per le attività ospitate su un' EC2 istanza Amazon.  Note Quando si calcola l'utilizzo della memoria, se MemoryReservation è specificato, viene utilizzato o nel calcolo al posto della memoria totale.			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
MemoryUtilization	La percentuale di memoria utilizzata dal cluster o dal servizio. L'utilizzo della memoria a livello di cluster (filtrato per <code>ClusterName</code>) viene misurato come la memoria totale utilizzata dalle attività di Amazon ECS sul cluster, divisa per la memoria totale di tutte le EC2 istanze Amazon registrate nel cluster. Solo EC2 le istanze Amazon in ACTIVE o in DRAINING stato influiranno sui parametri di utilizzo della memoria. La metrica a livello di cluster è supportata solo per le attività ospitate	<code>ClusterName</code> , <code>ServiceName</code>	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	su un'istanza Amazon. EC2 L'utilizzo della memoria a livello di servizio (filtrato per <code>ClusterName</code>, <code>ServiceName</code>) viene misurato come la memoria totale utilizzata dalle attività che appartengono al servizio, divisa per la memoria totale riservata alle attività che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su EC2 istanze Amazon e Fargate.			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
EBSFilesystemUtilization	La percentuale del file system Amazon EBS utilizzata dalle attività di un servizio. La metrica di utilizzo del file system EBS a livello di servizio (filtrata per <code>ClusterName</code>, <code>ServiceName</code>) viene misurata come la quantità totale del file system EBS utilizzata dalle attività che appartengono al servizio, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività che appartengono al servizio. La metrica di utilizzo del file system EBS a livello di servizio è disponibi	<code>ClusterName</code> , <code>ServiceName</code>	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	le solo per le attività ospitate su EC2 istanze Amazon (utilizzando la versione container agent) 1.79.0 e Fargate (utilizzando la versione della piattaforma) a cui è collegato un volume EBS. 1.4.0  Note Per le attività ospitate su Fargate, c'è spazio sul disco che viene utilizzato solo da Fargate. Non ci sono costi associati			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	allo spazio utilizzato da Fargate, ma vedrai questo spazio di archiviazione aggiuntivo o utilizzando strumenti come. df			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
GPUReservation	La percentuale del totale disponibile GPUs che viene riservato eseguendo attività nel cluster. La metrica di prenotazione GPU a livello di cluster viene misurata come il numero di attività GPUs riservate da Amazon ECS sul cluster, diviso per il numero totale di GPUs attività disponibili su tutte le EC2 istanze Amazon GPUs registrate nel cluster. Solo EC2 le istanze Amazon con ACTIVE o DRAINING stato influiranno sui parametri di prenotazione della GPU.	ClusterName	Media, minima, massima	Percentuale

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
ActiveConnectionCount	Il numero totale di connessioni simultanee e attive dai client ai proxy Amazon ECS Service Connect eseguite in attività che condividono il Discovery Name selezionato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect. Dimensioni valide: Discovery Name e Discovery Name, ServiceName, ClusterName .	Discovery Name e Discovery Name ServiceName , ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
NewConnectionCount	Il numero totale di nuove connessioni stabilite dai client ai proxy Amazon ECS Service Connect eseguite in attività che condividono il <code>DiscoveryName</code> selezionato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	<code>DiscoveryName</code> e <code>DiscoveryName</code> , <code>ServiceName</code> , <code>ClusterName</code>	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
Processed Bytes	Il numero totale di byte di traffico in entrata elaborato dai proxy Service Connect. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	Discovery Name e Discovery Name, ServiceName, ClusterName	Media, minimo, massimo, somma	Byte

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
RequestCount	Il numero richieste di traffico in entrata elaborato dai proxy Service Connect. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect. È inoltre necessario configurare appProtocol la mappatura delle porte nella definizione dell'attività.	DiscoveryName e DiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
GrpcRequestCount	Il numero di gRPC richieste di traffico in entrata elaborate dai proxy Service Connect. Questo parametro è disponibile solo se hai configurato Amazon ECS Service Connect e appProtocol è GRPC nella mappatura delle porte nella definizione dell'attività.	DiscoveryName e DiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
HTTPCode_ Target_2X X_Count	Il numero di codici di risposta HTTP con numeri da 200 a 299 generati dalle applicazioni in queste attività. Queste attività sono le destinazioni. Questo parametro conta solo le risposte inviate ai proxy Service Connect dalle applicazioni in queste attività, non le risposte inviate direttamente. Questo parametro è disponibile solo se hai configurato Amazon ECS Service Connect e appProtocol è HTTP o HTTP2 nella mappatura delle porte nella definizione dell'attività.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
	Dimensioni valide: .			

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
HTTPCode_ Target_3X X_Count	Il numero di codici di risposta HTTP con numeri da 300 a 399 generati dalle applicazioni in queste attività. Queste attività sono le destinazioni. Questo parametro conta solo le risposte inviate ai proxy Service Connect dalle applicazioni in queste attività, non le risposte inviate direttamente. Questo parametro è disponibile solo se hai configurato Amazon ECS Service Connect e appProtocol è HTTP o HTTP2 nella mappatura delle porte nella definizione dell'attività.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
HTTPCode_Target_4XX_Count	Il numero di codici di risposta HTTP con numeri da 400 a 499 generati dalle applicazioni in queste attività. Queste attività sono le destinazioni. Questo parametro conta solo le risposte inviate ai proxy Service Connect dalle applicazioni in queste attività, non le risposte inviate direttamente. Questo parametro è disponibile solo se hai configurato Amazon ECS Service Connect e appProtocol è HTTP o HTTP2 nella mappatura delle porte nella definizione dell'attività.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
HTTPCode_Target_5XX_Count	Il numero di codici di risposta HTTP con numeri da 500 a 599 generati dalle applicazioni in queste attività. Queste attività sono le destinazioni. Questo parametro conta solo le risposte inviate ai proxy Service Connect dalle applicazioni in queste attività, non le risposte inviate direttamente. Questo parametro è disponibile solo se hai configurato Amazon ECS Service Connect e appProtocol è HTTP o HTTP2 nella mappatura delle porte nella definizione dell'attività.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
RequestCountPerTarget	Il numero medio di richieste ricevute da ogni destinazione che condivide il DiscoveryName selezionato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media	Conteggio
TargetProcessedBytes	Il numero totale di byte elaborati dai proxy Service Connect. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Byte

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
TargetResponseTime	La latenza di elaborazione della richiesta dell'applicazione. Il tempo trascorso, in millisecondi, da quando la richiesta ha raggiunto il proxy Service Connect nell'attività di destinazione fino a quando il proxy di destinazione riceve una risposta dall'applicazione di destinazione. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	TargetDiscoveryName e TargetDiscoveryName, ServiceName, ClusterName	Media, minima, massima	Millisecondi

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
ClientTLSEstablishedConnectionsErrorCount	Il numero totale di volte in cui la connessione TLS non è riuscita. Questa metrica viene utilizzata solo quando TLS è abilitato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	DiscoveryName e, DiscoveryName, ServiceName, ClusterName	Media, minimo, massimo, somma	Conteggio

Parametro	Descrizione	Dimensione valida	Statistiche utili	Unità
TargetTLSNegotiationErrorCount	Il numero totale di volte in cui la connessione TLS non è riuscita a causa di certificati client mancanti, AWS Private CA verifiche non riuscite o verifiche SAN non riuscite. Questa metrica viene utilizzata solo quando TLS è abilitato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	ServiceName , e ClusterName TargetDiscoveryName e TargetDiscoveryName	Media, minimo, massimo, somma	Conteggio

Dimensioni per i parametri Amazon ECS

I parametri di Amazon ECS utilizzano lo spazio dei nomi AWS/ECS e forniscono i parametri per le seguenti dimensioni. Amazon ECS invia solo i parametri per le risorse contenenti attività nello stato RUNNING. Ad esempio, se si dispone di un cluster con un servizio al suo interno ma tale servizio non ha attività in uno stato RUNNING, non verranno inviate metriche a CloudWatch. Se si dispone di due servizi e uno di essi ha processi in esecuzione mentre l'altro no, verranno inviate solo le metriche per il servizio con i processi in esecuzione.

Dimensione	Definizione
ClusterName	Questa dimensione filtra i dati richiesti per tutte le risorse in un cluster specificato. Tutti i parametri di Amazon ECS vengono filtrati per ClusterName .
ServiceName	Questa dimensione filtra i dati richiesti per tutte le risorse in un servizio specificato all'interno di un cluster specificato.
DiscoveryName	Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato per tutti i cluster Amazon ECS. Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.
DiscoveryName, ServiceName, ClusterName	Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato tra le attività che hanno questo nome di rilevamento e che vengono create dal servizio in questo cluster. Se hai riutilizzato lo stesso nome di rilevamento in più servizi in spazi dei nomi diversi, utilizza questa dimensione per visualizzare i parametri del traffico in entrata per un servizio specifico. Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.
TargetDiscoveryName	Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di

Dimensione	Definizione
	rilevamento di Service Connect specificato per tutti i cluster Amazon ECS. A differenza di <code>DiscoveryName</code> , questi parametri del traffico misurano solo il traffico in entrata a questo <code>DiscoveryName</code> proveniente da altre attività di Amazon ECS che hanno una configurazione Service Connect in questo spazio dei nomi. Ciò include le attività eseguite dai servizi con una configurazione Service Connect solo client o client-server. Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

Dimensione	Definizione
TargetDiscoveryName, ServiceName, ClusterName	Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato, ma conta solo il traffico proveniente dalle attività create dal servizio in questo cluster. Utilizza questa dimensione per visualizzare i parametri del traffico in entrata provenienti da un client specifico in un altro servizio. A differenza di DiscoveryName, ServiceName, ClusterName, questi parametri del traffico misurano solo il traffico in entrata a questo DiscoveryName proveniente da altre attività di Amazon ECS che hanno una configurazione Service Connect in questo spazio dei nomi. Ciò include le attività eseguite dai servizi con una configurazione Service Connect solo client o client-server. Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

AWS Fargate metriche di utilizzo

Puoi utilizzare le metriche di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse del tuo account. Utilizza queste metriche per visualizzare l'utilizzo corrente del servizio su CloudWatch grafici e dashboard.

AWS Fargate le metriche di utilizzo corrispondono alle quote di servizio. AWS È possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sulle quote dei servizi Fargate, sugli [endpoint Amazon ECS](#) e sulle quote nel.. Riferimenti generali di Amazon Web Services

AWS Fargate pubblica le seguenti metriche nel namespace. AWS/Usage

Parametro	Descrizione
ResourceCount	Il numero totale delle risorse specificate in esecuzione nell'account. La risorsa è definita dalle dimensioni associate attraverso il parametro.

Le seguenti dimensioni vengono utilizzate per perfezionare i parametri di utilizzo pubblicati da AWS Fargate.

Dimensione	Descrizione
Service	Il nome del AWS servizio che contiene la risorsa. Per i parametri di utilizzo di AWS Fargate , il valore di questa dimensione è Fargate.
Type	Il tipo di entità che viene segnalato. Attualmente, l'unico valore valido per le metriche di AWS Fargate utilizzo è Resource.
Resource	Il tipo di risorsa in esecuzione. Il tipo di risorsa in esecuzione. Attualmente, l'unico valore valido per le metriche di AWS Fargate utilizzo è quello vCPU che restituisce informazioni sulle istanze in esecuzione.
Class	La classe della risorsa monitorata. La classe della risorsa monitorata. Per le metriche di AWS Fargate utilizzo con vCPU come valore della dimensione Resource, i valori Standard/OnDemand validi sono e. Standard/Spot

Puoi utilizzare la console Service Quotas per visualizzare l'utilizzo su un grafico e configurare allarmi che ti avvisano quando l' AWS Fargate utilizzo si avvicina a una quota di servizio. Per informazioni su come creare un CloudWatch allarme per avvisarti quando ti avvicini a una soglia di quota, consulta [Service Quotas and Amazon CloudWatch](#) alarms nella Guida per l'utente Service Quotas

Metriche di prenotazione dei cluster Amazon ECS

Le metriche di prenotazione del cluster vengono misurate come la percentuale di CPU e memoria e GPUs quelle riservate da tutte le attività di Amazon ECS su un cluster rispetto alla CPU e GPUs alla memoria aggregate registrate per ogni istanza di container attiva nel cluster. Solo le istanze di container nello stato ACTIVE o DRAINING influenzano i parametri di prenotazione del cluster. Questa metrica viene utilizzata solo su cluster con attività o servizi ospitati su istanze. EC2 Non è supportata nei cluster con attività ospitate su. AWS Fargate

$$\text{Cluster CPU reservation} = \frac{(\text{Total CPU units reserved by tasks in cluster}) \times 100}{(\text{Total CPU units registered by container instances in cluster})}$$

$$\text{Cluster memory reservation} = \frac{(\text{Total MiB of memory reserved by tasks in cluster} \times 100)}{(\text{Total MiB of memory registered by container instances in cluster})}$$

$$\text{Cluster GPU reservation} = \frac{(\text{Total GPUs reserved by tasks in cluster} \times 100)}{(\text{Total GPUs registered by container instances in cluster})}$$

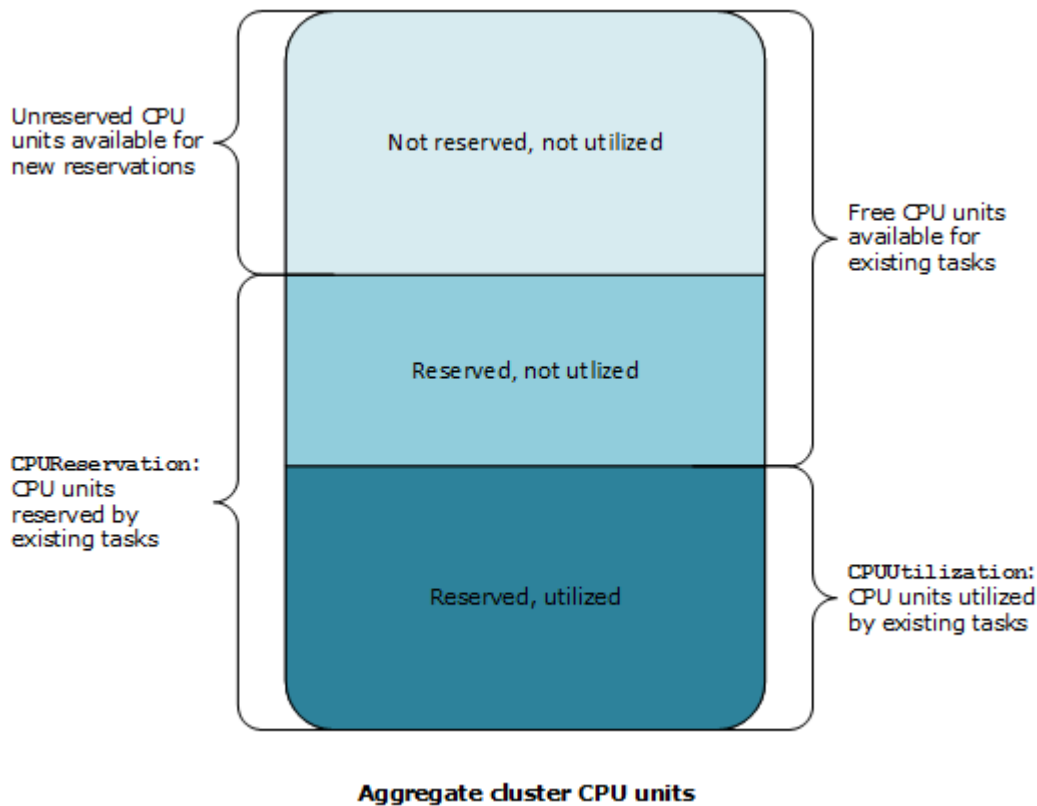
Quando esegui un'attività in un cluster, Amazon ECS analizza la definizione dell'attività e riserva le unità CPU aggregate, i MiB di memoria e GPUs quelle specificate nelle definizioni dei contenitori. Ogni minuto, Amazon ECS calcola il numero di unità CPU, MiB di memoria e GPUs che sono attualmente riservate per ogni attività in esecuzione nel cluster. Viene calcolata la quantità totale di CPU, memoria e risorse GPUs riservate per tutte le attività in esecuzione sul cluster e tali numeri vengono riportati CloudWatch come percentuale del totale delle risorse registrate per il cluster. Se nella definizione dell'attività specifichi un limite flessibile (`memoryReservation`), questo viene utilizzato per calcolare la quantità di memoria prenotata. In caso contrario, verrà utilizzato il limite rigido (`memory`). Il numero di MiB totali di memoria riservato dalle attività in un cluster include anche la dimensione del volume del file system (`tmpfs`) temporaneo e `sharedMemorySize` se definito

nella definizione delle attività. Per ulteriori informazioni sui limiti rigidi e flessibili, dimensione della memoria condivisa e del volume tmpfs, consulta [Parametri di definizione delle attività](#).

Ad esempio, un cluster ha due istanze di container attive registrate: un'istanza `c4.4xlarge` e un'istanza `c4.large`. L'istanza `c4.4xlarge` si registra nel cluster con 16.384 unità CPU e 30.158 MiB di memoria. L'istanza `c4.large` si registra con 2.048 unità CPU e 3.768 MiB di memoria. Le risorse aggregate di questo cluster sono 18.432 unità di CPU e 33.926 MiB di memoria.

Se una definizione di attività prenota 1.024 unità di CPU e 2.048 MiB di memoria e dieci attività vengono avviate con questa definizione attività su questo cluster (e nessun altra attività è attualmente in esecuzione), viene prenotato un totale di 10.240 unità di CPU e 20.480 MiB di memoria. Questo valore viene riportato CloudWatch come 55% di prenotazione della CPU e 60% di prenotazione di memoria per il cluster.

La figura seguente mostra il totale di unità di CPU registrate in un cluster e il significato della loro prenotazione e del loro utilizzo per le attività esistenti e il posizionamento di nuove attività. I blocchi inferiore (riservato, usato) e centrale (riservato, non utilizzato) rappresentano le unità CPU totali riservate alle attività esistenti in esecuzione sul cluster o alla `CPUReservation` CloudWatch metrica. Il blocco inferiore rappresenta le unità CPU riservate effettivamente utilizzate dalle attività in esecuzione sul cluster o la `CPUUtilization` CloudWatch metrica. Il blocco superiore rappresenta le unità di CPU che non sono prenotate dalle attività esistenti; queste unità di CPU sono disponibili per il nuovo posizionamento. Le attività esistenti possono usare queste unità di CPU non prenotate, se aumenta la loro necessità di risorse di CPU. Per ulteriori informazioni, consulta la documentazione del parametro della definizione di attività [cpu](#).



Metriche di utilizzo dei cluster Amazon ECS

I parametri di utilizzo del cluster sono disponibili per CPU, memoria e, quando è presente un volume EBS collegato alle attività, per l'utilizzo del file system EBS. Queste metriche sono disponibili solo per i cluster con attività o servizi ospitati su istanze Amazon EC2. Non sono supportati nei cluster con attività ospitate su AWS Fargate.

Parametri di utilizzo della CPU e della memoria a livello di cluster Amazon ECS

L'utilizzo della CPU e della memoria viene misurato come la percentuale di CPU e memoria utilizzata da tutte le attività su un cluster rispetto alla CPU e alla memoria aggregate registrate per ogni EC2 istanza Amazon attiva registrata nel cluster. Solo EC2 le istanze Amazon presenti ACTIVE o in DRAINING stato influiranno sui parametri di utilizzo del cluster.

$$\text{Cluster CPU utilization} = \frac{(\text{Total CPU units used by tasks in cluster}) \times 100}{(\text{Total CPU units registered by container instances in cluster})}$$

$$\text{Cluster memory utilization} = \frac{(\text{Total MiB of memory used by tasks in cluster} \times 100)}{(\text{Total MiB of memory registered by container instances in cluster})}$$

Ogni minuto, l'agente container Amazon ECS su ogni EC2 istanza Amazon calcola il numero di unità CPU e MiB di memoria attualmente utilizzati per ogni attività in esecuzione su quell' EC2 istanza Amazon e queste informazioni vengono riportate ad Amazon ECS. Viene calcolata la quantità totale di CPU e memoria utilizzata per tutte le attività in esecuzione sul cluster e tali numeri vengono riportati CloudWatch come percentuale del totale delle risorse registrate per il cluster.

Ad esempio, un cluster ha due EC2 istanze Amazon attive registrate, un'c4.4xlargeistanza e un'c4.largeistanza. L'c4.4xlargeistanza si registra nel cluster con unità 16,384 CPU e 30,158 MiB di memoria. L'c4.largeistanza viene registrata con unità 2,048 CPU e 3,768 MiB di memoria. Le risorse aggregate di questo cluster sono unità 18,432 CPU e 33,926 MiB di memoria.

Se su questo cluster sono in esecuzione dieci attività e ogni attività consuma unità 1,024 CPU e 2,048 MiB di memoria, nel cluster viene utilizzato un totale 10,240 di unità CPU e 20,480 MiB di memoria. Ciò corrisponde al CloudWatch 55% di utilizzo della CPU e al 60% di utilizzo della memoria per il cluster.

Utilizzo del file system Amazon EBS a livello di cluster Amazon ECS

La metrica di utilizzo del file system EBS a livello di cluster viene misurata come la quantità totale del filesystem EBS utilizzata dalle attività in esecuzione sul cluster, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività del cluster.

$$\text{Cluster EBS filesystem utilization} = \frac{(\text{Total GB of EBS filesystem used by tasks in cluster} \times 100)}{(\text{Total GB of EBS filesystem allocated to tasks in cluster})}$$

Metriche di utilizzo del servizio Amazon ECS

I parametri di utilizzo del servizio sono disponibili per la CPU, la memoria e, se alle tue attività è associato un volume EBS, l'utilizzo del file system EBS. Le metriche del livello di servizio sono supportate per i servizi con attività ospitate sia su EC2 istanze Amazon che su Fargate.

Utilizzo della CPU e della memoria a livello di servizio

L'utilizzo della CPU e della memoria viene misurato come la percentuale di CPU e memoria utilizzata dalle attività di Amazon ECS che appartengono a un servizio su un cluster rispetto alla CPU e alla memoria specificate nella definizione delle attività del servizio.

$$\text{Service CPU utilization} = \frac{(\text{Total CPU units used by tasks in service}) \times 100}{(\text{Total CPU units specified in task definition}) \times (\text{number of tasks in service})}$$

$$\text{Service memory utilization} = \frac{100 \times (\text{Total MiB of memory used by tasks in service})}{(\text{Total MiB of memory specified in task definition}) \times (\text{number of tasks in service})}$$

Amazon ECS raccoglie i parametri ogni 20 secondi. Ogni minuto, l'agente container Amazon ECS calcola il numero di unità CPU e MiB di memoria attualmente utilizzati per ogni attività in esecuzione di proprietà del servizio. Queste informazioni vengono riportate ad Amazon ECS. La quantità totale di CPU e memoria utilizzata per tutte le attività di proprietà del servizio in esecuzione nel cluster viene calcolata e tali numeri vengono segnalati a CloudWatch come una percentuale delle risorse totali specificate per il servizio nella definizione di attività del servizio. I valori minimo e massimo sono i più piccoli e i più grandi tra i parametri di 20 secondi. La media è l'aggregato dei 3 valori.

Se specifichi un limite flessibile (`memoryReservation`), viene utilizzato per calcolare la quantità di memoria prenotata. In caso contrario, verrà utilizzato il limite rigido (`memory`). Per ulteriori informazioni sui limiti rigidi e flessibili, vedere [Dimensioni processo](#).

Ad esempio, la definizione di attività per un servizio specifica un totale di 512 unità di CPU e 1.024 MiB di memoria (con il parametro `memory` di limite rigido) per tutti i suoi container. Il servizio dispone

di un conteggio desiderato di 1 attività in esecuzione, il servizio è in esecuzione in un cluster con 1 istanza di container `c4.large` (con 2.048 unità di CPU e 3.768 MiB di memoria totale) e non sono presenti altre attività in esecuzione nel cluster. Anche se l'attività specifica 512 unità di CPU, perché è l'unica attività in esecuzione in un'istanza di container con 2.048 unità di CPU, può utilizzare fino a quattro volte la quantità specificata (2.048/512). Tuttavia, la memoria specificata di 1.024 MiB è un limite insuperabile, perciò in questo caso, l'utilizzo della memoria di servizio non può superare il 100%.

Se l'esempio precedente utilizzasse il parametro `memoryReservation` di limite flessibile anziché il parametro `memory` di limite rigido, le attività del servizio potrebbero utilizzare una quantità superiore a 1.024 MB di memoria, in base alle necessità. In questo caso, l'utilizzo della memoria di servizio di utilizzo potrebbe superare il 100%.

Se la tua applicazione presenta un picco improvviso nell'utilizzo della memoria per un breve periodo di tempo, non vedrai aumentare l'utilizzo della memoria di servizio perché Amazon ECS raccoglie più punti dati ogni minuto e poi li aggrega in un punto dati a cui viene inviato. CloudWatch

Se questa attività esegue un lavoro a uso intensivo di CPU per un periodo e utilizza tutte le 2.048 unità di CPU e tutti i 512 MiB di memoria disponibili, il servizio segnala il 400% di utilizzo di CPU e il 50% di utilizzo di memoria. Se l'attività è inattiva e l'utilizzo di 128 unità di CPU e 128 MiB di memoria, il servizio segnala il 25% di utilizzo di CPU e il 12,5% di utilizzo della memoria.

Note

In questo esempio, l'utilizzo della CPU supererà il 100% solo quando le unità CPU sono definite a livello di container. Se definisci le unità CPU a livello di processo, l'utilizzo non supererà il limite definito a livello di processo.

Utilizzo del file system EBS a livello di servizio

L'utilizzo del file system EBS a livello di servizio viene misurato come la quantità totale del file system EBS utilizzata dalle attività che appartengono al servizio, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività che appartengono al servizio.

$$\text{Service EBS filesystem utilization} = \frac{\text{(Total GB of EBS filesystem used by tasks in the service)} \times 100}{\text{Total GB of EBS filesystem allocated for all tasks in the service}}$$

(Total GB of EBS filesystem allocated to tasks
in the service)

Conteggio dei processi **RUNNING** del servizio

Puoi utilizzare le CloudWatch metriche per visualizzare il numero di attività dei tuoi servizi che si trovano nello stato. **RUNNING** Ad esempio, puoi impostare un CloudWatch allarme per questa metrica per avvisarti se il numero di attività in esecuzione nel tuo servizio scende al di sotto di un valore specificato.

Numero di **RUNNING** attività di servizio in Amazon ECS CloudWatch Container Insights

Una metrica «Number of Running Tasks» (RunningTaskCount) è disponibile per cluster e per servizio quando utilizzi Amazon ECS CloudWatch Container Insights. Puoi utilizzare Container Insights per tutti i nuovi cluster creati attivando l'impostazione dell'`containerInsightsAccount`, sui singoli cluster attivando le impostazioni del cluster durante la creazione del cluster o sui cluster esistenti utilizzando l'API. `UpdateClusterSettings` Le metriche raccolte da CloudWatch Container Insights vengono addebitate come metriche personalizzate. Per ulteriori informazioni sui prezzi di CloudWatch , consulta [Prezzi di CloudWatch](#).

Per visualizzare questo parametro, consulta i [parametri di Amazon ECS Container Insights](#) nella Amazon CloudWatch User Guide.

Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge

Con Amazon EventBridge, puoi automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Le azioni che possono essere configurate automaticamente includono le seguenti:

- Aggiungere eventi ai gruppi di log in CloudWatch Logs
- Invocare una funzione AWS Lambda
- Richiamo del comando Amazon EC2 Run
- Inoltro dell'evento a Amazon Kinesis Data Streams
- Attivazione di una macchina a stati AWS Step Functions

- Notifica di un argomento Amazon SNS o una coda Amazon Simple Queue Service (Amazon SQS)

Per ulteriori informazioni, consulta la sezione Guida [introduttiva ad Amazon EventBridge](#) nella Amazon EventBridge User Guide.

Puoi utilizzare gli eventi Amazon ECS per EventBridge ricevere notifiche quasi in tempo reale sullo stato attuale dei tuoi cluster Amazon ECS. Se le tue attività utilizzano il tipo di EC2 avvio, puoi vedere lo stato di entrambe le istanze del contenitore e lo stato corrente di tutte le attività in esecuzione su tali istanze di container. Se le tue attività utilizzano il tipo di avvio Fargate, puoi vedere lo stato delle istanze del contenitore.

Utilizzando EventBridge, puoi creare pianificatori personalizzati su Amazon ECS responsabili dell'orchestrazione delle attività tra i cluster e del monitoraggio dello stato dei cluster quasi in tempo reale. Puoi eliminare il codice di pianificazione e monitoraggio che interroga continuamente il servizio Amazon ECS per le modifiche di stato e gestire invece le modifiche di stato di Amazon ECS in modo asincrono utilizzando qualsiasi destinazione. EventBridge Gli obiettivi potrebbero includere AWS Lambda Amazon Simple Queue Service, Amazon Simple Notification Service o Amazon Kinesis Data Streams.

Un flusso di eventi Amazon ECS assicura che ogni evento venga rilasciato almeno una volta. Se vengono inviati eventi duplicati, l'evento fornisce informazioni sufficienti a identificare i duplicati. Per ulteriori informazioni, consulta [Gestione degli eventi Amazon ECS](#).

Gli eventi sono relativamente ordinati, in modo che tu possa facilmente stabilire quando un evento si verifica in relazione ad altri eventi.

Argomenti

- [Eventi Amazon ECS](#)
- [Gestione degli eventi Amazon ECS](#)

Eventi Amazon ECS

Amazon ECS monitora lo stato di ogni processo e servizio. Se lo stato di un'attività o di un servizio cambia, viene generato un evento che viene inviato ad Amazon EventBridge. Questi eventi sono classificati come eventi di modifica dello stato delle attività ed eventi di operazioni di servizio. Questi eventi e le loro possibili cause vengono descritti in modo più dettagliato nelle sezioni seguenti.

Amazon ECS genera e invia i seguenti tipi di eventi a EventBridge:

- Modifica dello stato dell'istanza del contenitore
- Modifica dello stato dell'attività
- Deployment state change (Cambio stato distribuzione)
- Azione di servizio

Note

Amazon ECS potrebbe aggiungere altri tipi di eventi, fonti e dettagli in futuro. Se stai deserializzando i dati JSON degli eventi in codice, assicurati che l'applicazione sia pronta a gestire proprietà sconosciute per evitare problemi se e quando vengono aggiunte queste proprietà aggiuntive.

In alcuni casi, per la stessa attività vengono generati più eventi. Ad esempio, quando un'attività viene avviata su un'istanza di container, viene generato un evento di modifica dello stato delle attività per la nuova attività. Viene generato un evento di modifica dello stato delle istanze di container per verificare la modifica nelle risorse disponibili, ad esempio CPU, memoria e porte disponibili, sull'istanza di container. Analogamente, se un'istanza di container viene terminata, vengono generati eventi per l'istanza di container, per lo stato di connessione dell'agente del container e per tutte le attività che erano in esecuzione sull'istanza di container.

Gli eventi di modifica dello stato del container e di modifica dello stato delle attività contengono due campi `version`: uno nel corpo principale dell'evento e uno nell'oggetto `detail` dell'evento. Di seguito vengono descritte le differenze tra questi due campi:

- Il campo `version` nel corpo principale dell'evento è impostato su `0` per tutti gli eventi. Per ulteriori informazioni sui EventBridge parametri, consulta i [metadati degli eventi di AWS servizio](#) nella Amazon EventBridge User Guide.
- Il campo `version` nell'oggetto `detail` dell'evento descrive la versione della risorsa associata. Ogni volta che una risorsa cambia stato, tale versione viene incrementata. Poiché gli eventi possono essere inviati più volte, questo campo consente di identificare gli eventi duplicati. Gli eventi duplicati hanno la stessa versione nell'oggetto `detail`. Se stai replicando l'istanza del container Amazon ECS e lo stato dell'attività con EventBridge, puoi confrontare la versione di una risorsa riportata da Amazon ECS APIs con la versione riportata EventBridge per la risorsa (all'interno dell'`detail`oggetto) per verificare che la versione nel tuo flusso di eventi sia attuale.

Gli eventi di operazioni di servizio contengono solo il campo `version` nel corpo principale.

Gli eventi di azione del servizio specificano il servizio in 2 campi diversi:

- Per gli eventi generati da `create-service`, il servizio è sul `serviceName` campo.
- Per gli eventi generati da `update-service`, il servizio è sul `service` campo.

Se si utilizzano strumenti automatizzati per gli eventi di servizio, è necessario codificare per entrambi i campi.

Per informazioni su come creare una regola per gli eventi di intervento di servizio, vedere [Eventi di azione del servizio Amazon ECS](#).

Per ulteriori informazioni su come integrare Amazon ECS e EventBridge, consulta [Integrating Amazon and EventBridge Amazon ECS](#).

Eventi di modifica dello stato dell'istanza del contenitore Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle istanze di container.

Puoi chiamare le operazioni `StartTaskRunTask`, o `StopTask` API, direttamente o con l' AWS Management Console o SDKs.

Collocare o interrompere le attività su un'istanza di container comporta la modifica delle risorse disponibili sull'istanza di container, come la CPU, la memoria e le porte disponibili.

Il pianificatore del servizio Amazon ECS inizia o interrompe un processo.

Collocare o interrompere le attività su un'istanza di container comporta la modifica delle risorse disponibili sull'istanza di container, come la CPU, la memoria e le porte disponibili.

L'agente del container Amazon ECS chiama l'operazione API `SubmitTaskStateChange` con uno stato `STOPPED` per un processo con uno stato desiderato di `RUNNING`.

L'agente del container di Amazon ECS monitora lo stato dei processi nelle istanze di container e segnala qualsiasi modifica dello stato. Se un'attività che dovrebbe essere `RUNNING` viene trasferita a `STOPPED`, l'agente rilascia le risorse che erano allocate all'attività arrestata, come la CPU, la memoria e le porte disponibili.

Si annulla la registrazione dell'istanza del contenitore con l'operazione `DeregisterContainerInstance` API, direttamente o con o. AWS Management Console SDKs

L'annullamento della registrazione di un'istanza di container comporta la modifica dello stato dell'istanza di container e dello stato di connessione dell'agente del container Amazon ECS.

Un'attività è stata interrotta quando è stata interrotta un' EC2 istanza.

Quando interrompi un'istanza di container, le attività in esecuzione vengono trasferite in stato STOPPED.

L'agente del container di Amazon ECS registra un'istanza di container per la prima volta.

La prima volta che l'agente del container Amazon ECS registra un'istanza di container (all'avvio durante la prima esecuzione manuale), viene creato un evento di modifica dello stato per l'istanza.

L'agente del container Amazon ECS si connette o si disconnette da Amazon ECS.

Quando l'agente del container Amazon ECS si connette o si disconnette dal back-end di Amazon ECS modifica lo stato `agentConnected` dell'istanza di container.

 Note

L'agente del container di Amazon ECS si disconnette e si ricollega più volte all'ora come parte del normale funzionamento, pertanto è necessario prevedere gli eventi di connessione dell'agente. Questi eventi non indicano l'esistenza di un problema con l'agente container o l'istanza di container.

Puoi aggiornare l'agente del container Amazon ECS su un'istanza.

Il dettaglio di un'istanza di container contiene un oggetto per la versione dell'agente del container. Se aggiorni l'agente, le informazioni sulla versione cambiano e generano un evento.

Example Evento di modifica dello stato delle istanze di container

Gli eventi di modifica dello stato delle istanze di container vengono forniti nel formato seguente. La `detail` sezione seguente è simile all'[ContainerInstance](#) oggetto restituito da un'operazione [DescribeContainerInstances](#) API nell'Amazon Elastic Container Service API Reference. Per ulteriori informazioni sui EventBridge parametri, consulta i [metadati degli eventi di AWS servizio](#) nella Amazon EventBridge User Guide.

```
{
  "version": "0",
  "id": "8952ba83-7be2-4ab5-9c32-6687532d15a2",
  "detail-type": "ECS Container Instance State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2016-12-06T16:41:06Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ecs:us-east-1:111122223333:container-instance/
b54a2a04-046f-4331-9d74-3f6d7f6ca315"
  ],
  "detail": {
    "agentConnected": true,
    "attributes": [
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
      },
      {
        "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
      },
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
      },
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
      },
      {
        "name": "com.amazonaws.ecs.capability.privileged-container"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
      },
      {
        "name": "com.amazonaws.ecs.capability.ecr-auth"
      },
      {
```

```

    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
  },
  {
    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
  },
  {
    "name": "com.amazonaws.ecs.capability.task-iam-role"
  }
],
"clusterArn": "arn:aws:ecs:us-east-1:111122223333:cluster/default",
"containerInstanceArn": "arn:aws:ecs:us-east-1:111122223333:container-instance/
b54a2a04-046f-4331-9d74-3f6d7f6ca315",
"ec2InstanceId": "i-f3a8506b",
"registeredResources": [
  {
    "name": "CPU",
    "type": "INTEGER",
    "integerValue": 2048
  },
  {
    "name": "MEMORY",
    "type": "INTEGER",
    "integerValue": 3767
  },
  {
    "name": "PORTS",
    "type": "STRINGSET",
    "stringSetValue": [
      "22",
      "2376",
      "2375",
      "51678",
      "51679"
    ]
  }
],
{
  "name": "PORTS_UDP",
  "type": "STRINGSET",

```

```
    "stringSetValue": []
  },
],
"remainingResources": [
  {
    "name": "CPU",
    "type": "INTEGER",
    "integerValue": 1988
  },
  {
    "name": "MEMORY",
    "type": "INTEGER",
    "integerValue": 767
  },
  {
    "name": "PORTS",
    "type": "STRINGSET",
    "stringSetValue": [
      "22",
      "2376",
      "2375",
      "51678",
      "51679"
    ]
  },
  {
    "name": "PORTS_UDP",
    "type": "STRINGSET",
    "stringSetValue": []
  }
],
"status": "ACTIVE",
"version": 14801,
"versionInfo": {
  "agentHash": "aebcbca",
  "agentVersion": "1.13.0",
  "dockerVersion": "DockerVersion: 1.11.2"
},
"updatedAt": "2016-12-06T16:41:06.991Z"
}
```

Eventi di modifica dello stato delle attività di Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle attività:

Puoi chiamare `StartTask` le `RunTask` operazioni o le `StopTask` API, direttamente o con AWS Management Console AWS CLI, o SDKs.

L'avvio o l'arresto delle attività crea nuove risorse delle attività o modifica lo stato delle risorse delle attività esistenti.

Il pianificatore del servizio Amazon ECS inizia o interrompe un processo.

L'avvio o l'arresto delle attività crea nuove risorse delle attività o modifica lo stato delle risorse delle attività esistenti.

L'agente del container Amazon ECS chiama l'operazione API `SubmitTaskStateChange`

Per il tipo di EC2 avvio di Amazon, l'agente container Amazon ECS monitora lo stato delle attività sulle istanze di container. L'agente container Amazon ECS segnala eventuali modifiche di stato. Le modifiche dello stato potrebbero includere modifiche da `PENDING` a `RUNNING` o da `RUNNING` a `STOPPED`.

Imponi la cancellazione dell'istanza del contenitore sottostante con l'operazione `DeregisterContainerInstance` API e il `force` flag, direttamente o con l'opzione `or`. AWS Management Console SDKs

L'annullamento della registrazione di un'istanza di container comporta la modifica dello stato dell'istanza di container e dello stato di connessione dell'agente del container Amazon ECS. Se le attività sono in esecuzione sull'istanza di container, il flag `force` deve essere impostato per consentire l'annullamento della registrazione. Ciò arresta tutte le attività dell'istanza.

L'istanza di container sottostante viene arrestata o terminata.

Quando interrompi o termini un'istanza di container, le attività in esecuzione su di essa vengono trasferite in stato `STOPPED`.

Un container dell'attività cambia lo stato.

L'agente del container Amazon ECS monitora lo stato dei container all'interno dei processi. Ad esempio, se un container in esecuzione all'interno di un'attività viene arrestato, tale modifica dello stato del container genera un evento.

Un'attività che utilizza il provider di capacità Fargate Spot riceve un avviso di cessazione.

Quando un'attività utilizza il provider di capacità FARGATE_SPOT e viene interrotta a causa di un'interruzione Spot, viene generato un evento di modifica dello stato delle attività.

Example Evento di modifica dello stato dei processi

Gli eventi di modifica dello stato delle attività vengono forniti nel formato seguente. Tieni presente quanto segue sui campi:

- Lo stato di integrità dell'evento non è disponibile nell'evento di modifica dello stato dell'attività. Se è necessario lo stato di integrità dell'attività, è possibile eseguire [describe-tasks](#).
- Quando i contenitori utilizzano un'immagine ospitata con Amazon ECR, il `imageDigest` campo viene restituito.
- I valori per i `updatedAt` `campicreatedAt`, `connectivityAt`, `pullStartedAt` `startedAt` `pullStoppedAt`, e sono timestamp di stringhe ISO.
- Il `detail-type` valore è «ECS Task State Change».
- Quando l'evento viene generato per un'attività interrotta, i `stoppedCode` campi `stoppedReason` and forniscono informazioni aggiuntive sul motivo per cui l'attività è stata interrotta (ad esempio, «Iniziata dall'utente»).

Per ulteriori informazioni sui EventBridge parametri, consulta i [metadati degli eventi di AWS servizio](#) in Amazon EventBridge Events Reference.

Per informazioni su come configurare una regola di EventBridge eventi Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita perché uno dei suoi contenitori essenziali è terminato, consulta [Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS](#)

```
{
  "version": "0",
  "id": "105f6bb1-4da6-c630-4965-35383018cbca",
  "detail-type": "ECS Task State Change",
  "source": "aws.ecs",
  "account": "123456789012",
  "time": "2025-05-06T11:02:34Z",
  "region": "us-east-1",
  "resources": [
```

```

    "arn:aws:ecs:us-east-1:123456789012:task/example-cluster/a1173316d40a45dea9"
  ],
  "detail": {
    "attachments": [
      {
        "id": "fe3a9a46-6a47-40ee-afd9-7952ae90a75a",
        "type": "eni",
        "status": "ATTACHED",
        "details": [
          {
            "name": "subnetId",
            "value": "subnet-0d0eab1bb38d5ca64"
          },
          {
            "name": "networkInterfaceId",
            "value": "eni-0103a2f01bad57d71"
          },
          {
            "name": "macAddress",
            "value": "0e:50:d1:c1:77:81"
          },
          {
            "name": "privateDnsName",
            "value": "ip-10-0-1-163.ec2.internal"
          },
          {
            "name": "privateIPv4Address",
            "value": "10.0.1.163"
          }
        ]
      }
    ],
    "attributes": [
      {
        "name": "ecs.cpu-architecture",
        "value": "x86_64"
      }
    ],
    "availabilityZone": "us-east-1b",
    "capacityProviderName": "FARGATE",
    "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example-cluster",
    "connectivity": "CONNECTED",
    "connectivityAt": "2025-05-06T11:02:17.19Z",
    "containers": [

```

```

    {
      "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/example-
cluster/a1173316d40a45dea9/a0a99b87-baa8-4bf6-b9f1-a9a95917a635",
      "lastStatus": "RUNNING",
      "name": "web",
      "image": "nginx",
      "imageDigest":
"sha256:c15da6c91de8d2f436196f3a768483ad32c258ed4e1beb3d367a27ed67253e66",
      "runtimeId": "a1173316d40a45dea9-0265927825",
      "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example-cluster/
a1173316d40a45dea9",
      "networkInterfaces": [
        {
          "attachmentId": "fe3a9a46-6a47-40ee-afd9-7952ae90a75a",
          "privateIpv4Address": "10.0.1.163"
        }
      ],
      "cpu": "99",
      "memory": "100"
    },
    {
      "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/example-
cluster/a1173316d40a45dea9/a2010e2d-ba7c-4135-8b79-e0290ff3cd8c",
      "lastStatus": "RUNNING",
      "name": "aws-guardduty-agent-nm401c",
      "imageDigest":
"sha256:bf9197abdf853607e5fa392b4f97ccdd6ca56dd179be3ce8849e552d96582ac8",
      "runtimeId": "a1173316d40a45dea9-2098416933",
      "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example-cluster/
a1173316d40a45dea9",
      "networkInterfaces": [
        {
          "attachmentId": "fe3a9a46-6a47-40ee-afd9-7952ae90a75a",
          "privateIpv4Address": "10.0.1.163"
        }
      ],
      "cpu": "null"
    },
    {
      "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/example-
cluster/a1173316d40a45dea9/dccf0ca2-d929-471f-a5c3-98006fd4379e",
      "lastStatus": "RUNNING",
      "name": "aws-otel-collector",
      "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.32.0",

```



```

        "imageDigest":
          "sha256:7a1b3560655071bcacd66902c20ebe9a69470d5691fe3bd36baace7c2f3c4640",
          "runtimeId": "a1173316d40a45dea9-4027662657",
          "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example-cluster/
a1173316d40a45dea9",
          "networkInterfaces": [
            {
              "attachmentId": "fe3a9a46-6a47-40ee-afd9-7952ae90a75a",
              "privateIpv4Address": "10.0.1.163"
            }
          ],
          "cpu": "0"
        }
      ],
      "cpu": "256",
      "createdAt": "2025-05-06T11:02:13.877Z",
      "desiredStatus": "RUNNING",
      "enableExecuteCommand": false,
      "ephemeralStorage": {
        "sizeInGiB": 20
      },
      "group": "family:webserver",
      "launchType": "FARGATE",
      "lastStatus": "RUNNING",
      "memory": "512",
      "overrides": {
        "containerOverrides": [
          {
            "name": "web"
          },
          {
            "environment": [
              {
                "name": "CLUSTER_NAME",
                "value": "example-cluster"
              },
              {
                "name": "REGION",
                "value": "us-east-1"
              },
              {
                "name": "HOST_PROC",
                "value": "/host_proc"
              }
            ]
          }
        ]
      }
    ]
  }
}

```

```

        {
            "name": "AGENT_RUNTIME_ENVIRONMENT",
            "value": "ecsfgate"
        },
        {
            "name": "STAGE",
            "value": "prod"
        }
    ],
    "memory": 128,
    "name": "aws-guardduty-agent-nm401C"
},
{
    "name": "aws-otel-collector"
}
]
},
"platformVersion": "1.4.0",
"pullStartedAt": "2025-05-06T11:02:24.162Z",
"pullStoppedAt": "2025-05-06T11:02:33.493Z",
"startedAt": "2025-05-06T11:02:34.325Z",
"taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example-cluster/
a1173316d40a45dea9",
"taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
webserver:5",
"updatedAt": "2025-05-06T11:02:34.325Z",
"version": 3
}
}

```

Example

Di seguito è riportato un esempio di evento di modifica dello stato dell'attività per il tipo di EC2 avvio.

```

{
    "version": "0",
    "id": "a65cf262-f104-0dd5-ceda-4b09ba71a441",
    "detail-type": "ECS Task State Change",
    "source": "aws.ecs",
    "account": "123456789012",
    "time": "2025-05-12T13:12:06Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:ecs:us-east-1:123456789012:task/example/c1ffa94f19a540ed8d9f7e1d2a5d"
    ]
}

```

```

],
"detail": {
  "attachments": [
    {
      "id": "52333e3b-b812-41a8-b057-9ed184bbe5e1",
      "type": "eni",
      "status": "ATTACHED",
      "details": [
        {
          "name": "subnetId",
          "value": "subnet-0d0eab1bb38d5ca64"
        },
        {
          "name": "networkInterfaceId",
          "value": "eni-0ea90f746500773a4"
        },
        {
          "name": "macAddress",
          "value": "0e:d5:9b:ce:49:fb"
        },
        {
          "name": "privateDnsName",
          "value": "ip-10-0-1-37.ec2.internal"
        },
        {
          "name": "privateIPv4Address",
          "value": "10.0.1.37"
        }
      ]
    }
  ],
  "attributes": [
    {
      "name": "ecs.cpu-architecture",
      "value": "x86_64"
    }
  ],
  "availabilityZone": "us-east-1b",
  "capacityProviderName": "Infra-ECS-Cluster-example-fa84e0cc-AsgCapacityProvider-0seQJU9pizmp",
  "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/example",
  "connectivity": "CONNECTED",
  "connectivityAt": "2025-05-12T13:11:44.98Z",

```

```

    "containerInstanceArn": "arn:aws:ecs:us-east-1:123456789012:container-instance/
example/d1d84798400f49f3b21cb61610c1e",
    "containers": [
        {
            "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/example/
c1ffa94f19a540ed8d9f7e1d2a5d3626/197d0994-5367-4a6d-9f9a-f075e4a6",
            "lastStatus": "RUNNING",
            "name": "aws-otel-collector",
            "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.32.0",
            "imageDigest":
"sha256:7a1b3560655071bcacd66902c20ebe9a69470d5691fe3bd36baace7c2f3c4640",
            "runtimeId":
"8e926f0ccd8fe2b459926f49584ba6d33a3d9f61398dbabe944ee6a13a8ff3a1",
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example/
c1ffa94f19a540ed8d9f7e1d2a5d",
            "networkInterfaces": [
                {
                    "attachmentId": "52333e3b-b812-41a8-b057-9ed184bbe5e1",
                    "privateIpv4Address": "10.0.1.37"
                }
            ],
            "cpu": "0"
        },
        {
            "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/example/
c1ffa94f19a540ed8d9f7e1d2a5d3626/cab39ef0-9c50-459d-844b-b9d51d73d",
            "lastStatus": "RUNNING",
            "name": "web",
            "image": "nginx",
            "imageDigest":
"sha256:c15da6c91de8d2f436196f3a768483ad32c258ed4e1beb3d367a27ed67253e66",
            "runtimeId":
"9f1c73f0094f051541d9e5c2ab1e172d83c4eb5171bcc857c4504b02770ff3b8",
            "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example/
c1ffa94f19a540ed8d9f7e1d2a5d",
            "networkInterfaces": [
                {
                    "attachmentId": "52333e3b-b812-41a8-b057-9ed184bbe5e1",
                    "privateIpv4Address": "10.0.1.37"
                }
            ],
            "cpu": "99",
            "memory": "100"
        }
    ]
}

```

```

    ],
    "cpu": "256",
    "createdAt": "2025-05-12T13:11:44.98Z",
    "desiredStatus": "RUNNING",
    "enableExecuteCommand": false,
    "group": "family:webserver",
    "launchType": "EC2",
    "lastStatus": "RUNNING",
    "memory": "512",
    "overrides": {
      "containerOverrides": [
        {
          "name": "aws-otel-collector"
        },
        {
          "name": "web"
        }
      ]
    },
    "pullStartedAt": "2025-05-12T13:11:59.491Z",
    "pullStoppedAt": "2025-05-12T13:12:05.896Z",
    "startedAt": "2025-05-12T13:12:06.053Z",
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/example/
c1ffa94f19a540ed8d9f7e1d2a5d",
    "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/
webserver",
    "updatedAt": "2025-05-12T13:12:06.053Z",
    "version": 4
  }
}

```

Eventi di azione del servizio Amazon ECS

Amazon ECS invia eventi di operazioni di servizio con il tipo di dettaglio Operazione di servizio ECS. A differenza degli eventi di modifica dello stato delle istanze di container e delle attività, gli eventi di operazioni di servizio non includono un numero di versione nel campo di risposta `details`. Di seguito è riportato uno schema di eventi utilizzato per creare una EventBridge regola per gli eventi di azione del servizio Amazon ECS. Per ulteriori informazioni, consulta la sezione Guida [introduttiva EventBridge alla](#) Amazon EventBridge User Guide.

```
{
```

```
"source": [
  "aws.ecs"
],
"detail-type": [
  "ECS Service Action"
]
}
```

Amazon ECS invia eventi di tipo INFO, WARN e ERROR. Di seguito sono riportati gli eventi di operazioni di servizio.

Eventi di operazione di servizio con il tipo di eventi **INFO**

SERVICE_STEADY_STATE

Il servizio è integro e con il numero desiderato di attività, raggiungendo così uno stato costante. Il pianificatore del servizio segnala periodicamente lo stato, quindi potresti ricevere questo messaggio più volte.

TASKSET_STEADY_STATE

Il set di attività è integro e con il numero desiderato di attività, raggiungendo così uno stato costante.

CAPACITY_PROVIDER_STEADY_STATE

Un provider di capacità associato a un servizio raggiunge uno stato costante.

SERVICE_DESIRED_COUNT_UPDATED

Quando il pianificatore del servizio aggiorna il conteggio desiderato calcolato per un servizio o un set di attività. Questo evento non viene inviato quando il conteggio desiderato viene aggiornato manualmente da un utente.

TASKS_STOPPED

Il servizio ha interrotto l'operazione in esecuzione.

SERVICE_DEPLOYMENT_IN_PROGRESS

È in corso l'implementazione di un servizio. L'implementazione del servizio può essere un rollback o una nuova revisione del servizio.

SERVICE_DEPLOYMENT_COMPLETED

L'implementazione di un servizio è in stato stazionario ed è completa. L'implementazione del servizio può essere un rollback o l'implementazione di una revisione aggiornata del servizio.

Eventi di operazione di servizio con il tipo di eventi **WARN**

SERVICE_TASK_START_IMPAIRED

Il servizio non è in grado di avviare le attività in modo coerente.

SERVICE_DISCOVERY_INSTANCE_UNHEALTHY

Un servizio che utilizza l'individuazione dei servizi contiene un'attività non integra. Il pianificatore del servizio rileva che un'attività in un registro di servizio non è integra.

VPC_LATTICE_TARGET_UNHEALTHY

Il servizio che utilizza VPC Lattice ha rilevato che uno degli obiettivi del VPC Lattice non è integro.

Eventi di operazione di servizio con il tipo di eventi **ERROR**

SERVICE_DAEMON_PLACEMENT_CONSTRAINT_VIOLATED

Un'attività in un servizio che utilizza la strategia del pianificatore del servizio DAEMON non soddisfa più la strategia di vincolo di posizionamento per il servizio.

ECS_OPERATION_THROTTLED

Il pianificatore del servizio è stato limitato a causa dei limiti dell'API Amazon ECS.

SERVICE_DISCOVERY_OPERATION_THROTTLED

Il service scheduler è stato limitato a causa dei limiti di accelerazione delle API. AWS Cloud Map Può verificarsi nei servizi configurati per utilizzare l'individuazione dei servizi.

SERVICE_TASK_PLACEMENT_FAILURE

Il pianificatore del servizio non è in grado di posizionare un'attività. La causa è descritta nel campo `reason`.

Una causa comune per la generazione di questo evento di servizio è dovuta alla mancanza di risorse nel cluster per collocare l'attività. Ad esempio, non vi è una sufficiente capacità di CPU

o memoria nelle istanze di container disponibili o non è disponibile alcuna istanza di container. Un'altra causa comune si verifica quando l'agente di container di Amazon ECS viene disconnesso nell'istanza di container, impedendo all'utilità di pianificazione di collocare l'attività.

SERVICE_TASK_CONFIGURATION_FAILURE

Il pianificatore del servizio non è in grado di posizionare un'attività a causa di un errore di configurazione. La causa è descritta nel campo `reason`.

Una causa comune della generazione di questo evento di servizio è dovuta al fatto che i tag sono stati applicati al servizio ma l'utente o il ruolo non avevano scelto il nuovo formato del nome della risorsa Amazon (ARN) nella regione. Per ulteriori informazioni, consulta [Nomi di risorse Amazon \(ARNs\) e IDs](#). Un'altra causa comune è che Amazon ECS non è riuscito ad assumere il ruolo IAM del processo fornito.

SERVICE_HEALTH_UNKNOWN

Il servizio non è stato in grado di descrivere i dati sanitari relativi alle attività.

SERVICE_DEPLOYMENT_FAILED

L'implementazione di un servizio non ha raggiunto il livello di stabilità. Ciò accade quando CloudWatch viene attivato un o l'interruttore rileva un errore di implementazione del servizio.

Example Evento dello stato costante del servizio

Gli eventi dello stato costante del servizio vengono forniti nel seguente formato. Per ulteriori informazioni sui EventBridge parametri, consulta la sezione [Eventi EventBridge nella](#) Amazon EventBridge User Guide.

```
{
  "version": "0",
  "id": "af3c496d-f4a8-65d1-70f4-a69d52e9b584",
  "detail-type": "ECS Service Action",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2019-11-19T19:27:22Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
  ],
  "detail": {
    "eventType": "INFO",
```



```
    "eventName": "SERVICE_STEADY_STATE",
    "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
    "createdAt": "2019-11-19T19:27:22.695Z"
  }
}
```

Example Evento dello stato costante del provider di capacità

Gli eventi dello stato costante del provider di capacità vengono forniti nel formato seguente.

```
{
  "version": "0",
  "id": "b9baa007-2f33-0eb1-5760-0d02a572d81f",
  "detail-type": "ECS Service Action",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2019-11-19T19:37:00Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
  ],
  "detail": {
    "eventType": "INFO",
    "eventName": "CAPACITY_PROVIDER_STEADY_STATE",
    "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
    "capacityProviderArns": [
      "arn:aws:ecs:us-west-2:111122223333:capacity-provider/ASG-tutorial-capacity-provider"
    ],
    "createdAt": "2019-11-19T19:37:00.807Z"
  }
}
```

Example Eventi compromessi di avvio dei processi di servizio

Gli eventi compromessi di avvio attività di servizio vengono forniti nel formato seguente.

```
{
  "version": "0",
  "id": "57c9506e-9d21-294c-d2fe-e8738da7e67d",
  "detail-type": "ECS Service Action",
  "source": "aws.ecs",
  "account": "111122223333",
```

```

    "time": "2019-11-19T19:55:38Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
    ],
    "detail": {
        "eventType": "WARN",
        "eventName": "SERVICE_TASK_START_IMPAIRED",
        "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
        "createdAt": "2019-11-19T19:55:38.725Z"
    }
}

```

Example Evento di errore di posizionamento dei processi di servizio

Gli eventi di errore di posizionamento dell'attività di servizio vengono forniti nel formato seguente. Per ulteriori informazioni, consulta la sezione [Eventi EventBridge nella](#) Amazon EventBridge User Guide.

Nell'esempio seguente, l'attività tenta di utilizzare il provider di capacità FARGATE_SPOT, ma il pianificatore del servizio non è stato in grado di acquisire alcuna capacità Fargate Spot.

```

{
    "version": "0",
    "id": "ddca6449-b258-46c0-8653-e0e3a6d0468b",
    "detail-type": "ECS Service Action",
    "source": "aws.ecs",
    "account": "111122223333",
    "time": "2019-11-19T19:55:38Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
    ],
    "detail": {
        "eventType": "ERROR",
        "eventName": "SERVICE_TASK_PLACEMENT_FAILURE",
        "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
        "capacityProviderArns": [
            "arn:aws:ecs:us-west-2:111122223333:capacity-provider/FARGATE_SPOT"
        ],
        "reason": "RESOURCE:FARGATE",
        "createdAt": "2019-11-06T19:09:33.087Z"
    }
}

```

Nell'esempio seguente, per quanto riguarda il tipo di EC2 avvio, è stato tentato di avviare l'operazione sull'istanza Container, 2dd1b186f39845a584488d2ef155c131 ma lo scheduler del servizio non è riuscito a collocarla a causa di una CPU insufficiente.

```
{
  "version": "0",
  "id": "ddca6449-b258-46c0-8653-e0e3a6d0468b",
  "detail-type": "ECS Service Action",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2019-11-19T19:55:38Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
  ],
  "detail": {
    "eventType": "ERROR",
    "eventName": "SERVICE_TASK_PLACEMENT_FAILURE",
    "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
    "containerInstanceArns": [
      "arn:aws:ecs:us-west-2:111122223333:container-instance/default/2dd1b186f39845a584488d2ef155c131"
    ],
    "reason": "RESOURCE:CPU",
    "createdAt": "2019-11-06T19:09:33.087Z"
  }
}
```

Eventi di modifica dello stato di implementazione del servizio Amazon ECS

Amazon ECS invia eventi di modifica dello stato di implementazione del servizio con il tipo di dettaglio Modifica dello stato di implementazione ECS. Di seguito è riportato uno schema di eventi utilizzato per creare una EventBridge regola per gli eventi di modifica dello stato di distribuzione del servizio Amazon ECS. Per ulteriori informazioni sulla creazione di una EventBridge regola, consulta la sezione Guida [introduttiva ad Amazon EventBridge](#) nella Amazon EventBridge User Guide.

```
{
  "source": [
    "aws.ecs"
  ],
  "detail-type": [
    "ECS Deployment State Change"
  ]
}
```

```
]
}
```

Amazon ECS invia eventi di tipo INFO e ERROR. Per ulteriori informazioni, consulta [Eventi di azione del servizio Amazon ECS](#)

Di seguito sono riportati gli eventi di modifica dello stato di un'implementazione di servizi.

SERVICE_DEPLOYMENT_IN_PROGRESS

L'implementazione del servizio è in corso. Questo evento viene inviato sia per le implementazioni iniziali che per le implementazioni dei ripristini dello stato precedente.

SERVICE_DEPLOYMENT_COMPLETED

L'implementazione del servizio è stata completata. Questo evento viene inviato una volta che un servizio raggiunge uno stato costante dopo un'implementazione.

SERVICE_DEPLOYMENT_FAILED

L'implementazione del servizio non è riuscita. Questo evento viene inviato per i servizi con la logica dell'interruttore automatico di implementazione abilitata.

Example Evento di implementazione del servizio in corso

Gli eventi di implementazione del servizio in corso vengono recapitati all'avvio di un'implementazione iniziale e di un'implementazione di un ripristino dello stato precedente. La differenza tra i due è nel campo `reason`. Per ulteriori informazioni sui EventBridge parametri, consulta i [metadati degli eventi di AWS servizio](#) nella Amazon EventBridge User Guide.

Di seguito è illustrato un output di esempio per l'avvio di un'implementazione iniziale.

```
{
  "version": "0",
  "id": "ddca6449-b258-46c0-8653-e0e3a6EXAMPLE",
  "detail-type": "ECS Deployment State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2020-05-23T12:31:14Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
```

```

],
"detail": {
  "eventType": "INFO",
  "eventName": "SERVICE_DEPLOYMENT_IN_PROGRESS",
  "deploymentId": "ecs-svc/123",
  "updatedAt": "2020-05-23T11:11:11Z",
  "reason": "ECS deployment deploymentId in progress."
}
}

```

Di seguito è illustrato un output di esempio per l'avvio di un'implementazione del ripristino dello stato precedente. Il campo `reason` fornisce l'ID dell'implementazione in cui il servizio sta eseguendo il ripristino dello stato precedente.

```

{
  "version": "0",
  "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
  "detail-type": "ECS Deployment State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2020-05-23T12:31:14Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
  ],
  "detail": {
    "eventType": "INFO",
    "eventName": "SERVICE_DEPLOYMENT_IN_PROGRESS",
    "deploymentId": "ecs-svc/123",
    "updatedAt": "2020-05-23T11:11:11Z",
    "reason": "ECS deployment circuit breaker: rolling back to
deploymentId deploymentID."
  }
}

```

Example Evento di implementazione del servizio completata

Gli eventi di implementazione del servizio completata vengono forniti nel seguente formato. Per ulteriori informazioni, consulta [Implementa i servizi Amazon ECS sostituendo le attività](#).

```

{
  "version": "0",

```

```

    "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
    "detail-type": "ECS Deployment State Change",
    "source": "aws.ecs",
    "account": "111122223333",
    "time": "2020-05-23T12:31:14Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
    ],
    "detail": {
        "eventType": "INFO",
        "eventName": "SERVICE_DEPLOYMENT_COMPLETED",
        "deploymentId": "ecs-svc/123",
        "updatedAt": "2020-05-23T11:11:11Z",
        "reason": "ECS deployment deploymentID completed."
    }
}

```

Example Evento di implementazione del servizio non riuscita

Gli eventi di implementazione del servizio non riuscita vengono forniti nel seguente formato. Un evento di implementazione del servizio con stato non riuscito verrà inviato solo per i servizi che dispongono della logica dell'interruttore automatico di implementazione abilitata. Per ulteriori informazioni, consulta [Implementa i servizi Amazon ECS sostituendo le attività](#).

```

{
    "version": "0",
    "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
    "detail-type": "ECS Deployment State Change",
    "source": "aws.ecs",
    "account": "111122223333",
    "time": "2020-05-23T12:31:14Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
    ],
    "detail": {
        "eventType": "ERROR",
        "eventName": "SERVICE_DEPLOYMENT_FAILED",
        "deploymentId": "ecs-svc/123",
        "updatedAt": "2020-05-23T11:11:11Z",
        "reason": "ECS deployment circuit breaker: task failed to start."
    }
}

```

```
}
```

Gestione degli eventi Amazon ECS

Amazon ECS invia gli eventi secondo il criterio almeno una volta. Ciò significa che potresti ricevere più copie di un determinato evento. Inoltre, gli eventi potrebbero non essere inviati al listener di eventi nell'ordine in cui si sono verificati.

Per ordinare correttamente gli eventi, la sezione `detail` di ogni evento contiene una proprietà `version`. Ogni volta che una risorsa cambia stato, avviene un incremento di `version`. Gli eventi duplicati hanno la stessa `version` nell'oggetto `detail`. Se stai replicando l'istanza del container Amazon ECS e lo stato dell'attività con EventBridge, puoi confrontare la versione di una risorsa riportata da Amazon ECS APIs con quella `version` riportata per la risorsa EventBridge per verificare che la versione nel tuo flusso di eventi sia attuale. Gli eventi con un numero di proprietà di versione più alto devono essere considerati come eventi successivi a quelli aventi numeri di versione più bassi.

Esempio: gestione degli eventi in una funzione AWS Lambda

Nell'esempio seguente viene riportata una funzione Lambda scritta in Python 3.9, la quale acquisisce sia eventi di modifica dello stato delle istanze di container che dei processi e li salva in una delle due tabelle Amazon DynamoDB:

- `ECSCtrInstanceState`— Memorizza lo stato più recente di un'istanza di container. L'ID tabella è il valore `containerInstanceArn` dell'istanza di container;
- `ECSTaskStato`: memorizza lo stato più recente di un'attività. L'ID tabella è il valore `taskArn` dell'attività.

```
import json
import boto3

def lambda_handler(event, context):
    id_name = ""
    new_record = {}

    # For debugging so you can see raw event format.
    print('Here is the event:')
    print((json.dumps(event)))

    if event["source"] != "aws.ecs":
```

```

        raise ValueError("Function only supports input from events with a source type
of: aws.ecs")

# Switch on task/container events.
table_name = ""
if event["detail-type"] == "ECS Task State Change":
    table_name = "ECSTaskState"
    id_name = "taskArn"
    event_id = event["detail"]["taskArn"]
elif event["detail-type"] == "ECS Container Instance State Change":
    table_name = "ECSCtrInstanceState"
    id_name = "containerInstanceArn"
    event_id = event["detail"]["containerInstanceArn"]
else:
    raise ValueError("detail-type for event is not a supported type. Exiting
without saving event.")

new_record["cw_version"] = event["version"]
new_record.update(event["detail"])

# "status" is a reserved word in DDB, but it appears in containerPort
# state change messages.
if "status" in event:
    new_record["current_status"] = event["status"]
    new_record.pop("status")

# Look first to see if you have received a newer version of an event ID.
# If the version is OLDER than what you have on file, do not process it.
# Otherwise, update the associated record with this latest information.
print("Looking for recent event with same ID...")
dynamodb = boto3.resource("dynamodb", region_name="us-east-1")
table = dynamodb.Table(table_name)
saved_event = table.get_item(
    Key={
        id_name : event_id
    }
)
if "Item" in saved_event:
    # Compare events and reconcile.
    print(("EXISTING EVENT DETECTED: Id " + event_id + " - reconciling"))
    if saved_event["Item"]["version"] < event["detail"]["version"]:
        print("Received event is a more recent version than the stored event -
updating")

```



```

        table.put_item(
            Item=new_record
        )
    else:
        print("Received event is an older version than the stored event -
ignoring")
    else:
        print(("Saving new event - ID " + event_id))

        table.put_item(
            Item=new_record
        )

```

Il seguente esempio di Fargate mostra una funzione Lambda scritta in Python 3.9 che acquisisce gli eventi di modifica dello stato delle attività e li salva nella seguente tabella Amazon DynamoDB:

```

import json
import boto3

def lambda_handler(event, context):
    id_name = ""
    new_record = {}

    # For debugging so you can see raw event format.
    print('Here is the event:')
    print((json.dumps(event)))

    if event["source"] != "aws.ecs":
        raise ValueError("Function only supports input from events with a source type
of: aws.ecs")

    # Switch on task/container events.
    table_name = ""
    if event["detail-type"] == "ECS Task State Change":
        table_name = "ECSTaskState"
        id_name = "taskArn"
        event_id = event["detail"]["taskArn"]
    else:
        raise ValueError("detail-type for event is not a supported type. Exiting
without saving event.")

    new_record["cw_version"] = event["version"]
    new_record.update(event["detail"])

```

```
# "status" is a reserved word in DDB, but it appears in containerPort
# state change messages.
if "status" in event:
    new_record["current_status"] = event["status"]
    new_record.pop("status")

# Look first to see if you have received a newer version of an event ID.
# If the version is OLDER than what you have on file, do not process it.
# Otherwise, update the associated record with this latest information.
print("Looking for recent event with same ID...")
dynamodb = boto3.resource("dynamodb", region_name="us-east-1")
table = dynamodb.Table(table_name)
saved_event = table.get_item(
    Key={
        id_name : event_id
    }
)
if "Item" in saved_event:
    # Compare events and reconcile.
    print(("EXISTING EVENT DETECTED: Id " + event_id + " - reconciling"))
    if saved_event["Item"]["version"] < event["detail"]["version"]:
        print("Received event is a more recent version than the stored event -
updating")
        table.put_item(
            Item=new_record
        )
    else:
        print("Received event is an older version than the stored event -
ignoring")
else:
    print(("Saving new event - ID " + event_id))

    table.put_item(
        Item=new_record
    )
```

Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata

Container Insights raccoglie, aggrega e riepiloga le metriche e i log delle applicazioni containerizzate e dei microservizi. fornisce tutte le metriche di Container Insights, oltre a metriche aggiuntive su attività e container.

Container Insights rileva tutti i container in esecuzione in un cluster e raccoglie dati sulle prestazioni a ogni livello dello stack di prestazioni. I dati operativi sono raccolti come eventi di log delle prestazioni. Questi sono elementi che usano uno schema JSON strutturato che consente ai dati ad alta cardinalità di essere acquisiti e archiviati su larga scala. Da questi dati, CloudWatch crea metriche aggregate di livello superiore a livello di cluster, servizio e attività come metriche. CloudWatch I parametri includono l'utilizzo di risorse come CPU, memoria, dischi e rete. I parametri sono disponibili in pannelli di controllo automatici CloudWatch.

Le metriche riflettono solo le risorse con attività in esecuzione nell'intervallo di tempo specificato. Ad esempio, se si dispone di un cluster con un servizio al suo interno ma tale servizio non ha attività in uno stato RUNNING, non verranno inviate metriche a CloudWatch. Se si dispone di due servizi e uno di essi ha attività in esecuzione e l'altro no, verranno inviate solo le metriche per il servizio con attività in esecuzione.

Il 2 dicembre 2024, AWS ha rilasciato Container Insights con osservabilità migliorata per Amazon ECS. Questa versione supporta l'osservabilità migliorata per i cluster Amazon ECS utilizzando i tipi di lancio Amazon e EC2 Fargate. Dopo aver configurato Container Insights con un'osservabilità migliorata su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura dal livello del cluster fino al livello del contenitore nel tuo ambiente e visualizza questi dati critici sulle prestazioni in dashboard curate, eliminando il carico di lavoro legato alla configurazione dell'osservabilità. Per informazioni su come configurare Container Insights con una migliore osservabilità, consulta [Container Insights con osservabilità migliorata](#)

Ti consigliamo di utilizzare Container Insights con una migliore osservabilità anziché Container Insights perché offre una visibilità dettagliata nell'ambiente dei container, riducendo il tempo medio di risoluzione. Per ulteriori informazioni, consulta [Amazon ECS Container Insights con metriche di osservabilità avanzate](#) nella Amazon CloudWatch User Guide.

Gli eventi che puoi visualizzare sono quelli che Amazon ECS invia ad Amazon EventBridge. Per ulteriori informazioni, consulta [Eventi di Amazon ECS](#).

⚠ Important

Le metriche raccolte da CloudWatch Container Insights vengono addebitate come metriche personalizzate. [Per ulteriori informazioni sui CloudWatch prezzi, consulta CloudWatch la sezione Prezzi.](#)

Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container

Quando crei una definizione di attività, puoi configurare un controllo dello stato dei tuoi contenitori. I controlli di integrità sono comandi eseguiti localmente su un contenitore e convalidano lo stato e la disponibilità delle applicazioni.

L'agente del container Amazon ECS monitora (e invia i report correlati) solo i controlli dell'integrità specificati nella definizione di attività. Amazon ECS non monitora i controlli dell'integrità di Docker incorporati in un'immagine del container e non specificati nella definizione del container. I parametri di controllo dello stato specificati in una definizione del container sostituiscono qualsiasi controllo dello stato Docker presente nell'immagine del container.

Quando un controllo di integrità è definito in una definizione di attività, il contenitore esegue il processo di controllo dello stato all'interno del contenitore, quindi valuta il codice di uscita per determinare lo stato dell'applicazione.

Il controllo dello stato di salute comprende i seguenti parametri:

- **Comando:** il comando che il contenitore esegue per determinare se è integro. La matrice di stringhe può iniziare con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell predefinita del container.
- **Intervallo:** il periodo di tempo (in secondi) tra ogni controllo sanitario.
- **Timeout:** il periodo di tempo (in secondi) di attesa dell'esito positivo di un controllo sanitario prima che venga considerato un fallimento.
- **Tentativi:** il numero di volte in cui si ripete un controllo di integrità non riuscito prima che il contenitore venga considerato inintegro.
- **Periodo di inizio:** il periodo di prova opzionale per concedere ai container il tempo necessario per riavviarsi prima che i controlli di integrità non riusciti vengano conteggiati ai fini del numero massimo di nuovi tentativi.

Se un controllo dello stato va a buon fine all'interno di `startPeriod`, il container è considerato integro e gli errori successivi vengono conteggiati rispetto al numero massimo di nuovi tentativi.

Per informazioni su come specificare un controllo dello stato di salute nella definizione di un'attività, vedere. [Controllo dello stato](#)

Di seguito vengono descritti i possibili valori dello stato di salute di un contenitore:

- **HEALTHY**: il controllo dell'integrità del container è stato superato correttamente.
- **UNHEALTHY**: il controllo dell'integrità del container non è andato a buon fine.
- **UNKNOWN**: il controllo dell'integrità del container è in fase di valutazione, non è stato definito alcun controllo dell'integrità o Amazon ECS non dispone dello stato di integrità del container.

I comandi di controllo dello stato di salute vengono eseguiti sul contenitore. Pertanto è necessario includere i comandi nell'immagine del contenitore.

Il controllo dello stato si connette all'applicazione tramite l'interfaccia di loopback del contenitore all'indirizzo `localhost` o `127.0.0.1`. Un codice di uscita pari a `0` indica l'esito positivo, mentre un codice di uscita diverso da zero indica un errore.

Quando utilizzi i controlli dello stato dei contenitori, considera quanto segue:

- I controlli dello stato del container richiedono la versione 1.17.0 o successiva dell'agente del container Amazon ECS.
- I controlli dello stato dei container sono supportati per le attività Fargate se si utilizza una versione della piattaforma Linux 1.1.0 o superiore o una versione della piattaforma Windows o superiore 1.1.0

In che modo Amazon ECS determina lo stato delle attività

I contenitori essenziali e con il comando `Health Check` nella definizione dell'attività sono gli unici considerati per determinare lo stato dell'attività.

Le seguenti regole vengono valutate nell'ordine:

1. Se lo stato di un contenitore essenziale è **UNHEALTHY**, lo stato dell'attività è **UNHEALTHY**.
2. Se lo stato di un contenitore essenziale è **UNKNOWN**, lo stato dell'attività è **UNKNOWN**.

3. Se lo stato di tutti i contenitori essenziali è **HEALTHY**, lo stato dell'attività è **HEALTHY**.

Considerate il seguente esempio di integrità di un'attività con 2 contenitori essenziali.

Contenitore 1: salute	Salute del contenitore 2	Stato dell'attività
UNHEALTHY	UNKNOWN	UNHEALTHY
UNHEALTHY	HEALTHY	UNHEALTHY
HEALTHY	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY

Considerate il seguente esempio di integrità di un'attività con 3 contenitori.

Salute del contenitore 1	Salute del contenitore 2	Salute del contenitore 3	Stato dell'attività
UNHEALTHY	UNKNOWN	UNKNOWN	UNHEALTHY
UNHEALTHY	UNKNOWN	HEALTHY	UNHEALTHY
UNHEALTHY	HEALTHY	HEALTHY	UNHEALTHY
HEALTHY	UNKNOWN	HEALTHY	UNKNOWN
HEALTHY	UNKNOWN	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY	HEALTHY

In che modo i controlli sanitari vengono influenzati dalle disconnessioni degli agenti

Se l'agente container Amazon ECS si disconnette dal servizio Amazon ECS, ciò non causerà la transizione del container a uno stato **UNHEALTHY**. Questo è stato progettato per garantire che i contenitori rimangano in funzione durante il riavvio degli agenti o la temporanea indisponibilità.

Lo stato del controllo di integrità è l'ultima risposta ricevuta dall'agente Amazon ECS, quindi se il container è stato considerato HEALTHY prima della disconnessione, tale stato rimarrà invariato fino alla riconnessione dell'agente e all'esecuzione di un altro controllo dello stato. Non ci sono ipotesi sullo stato dei controlli dell'integrità del container.

Visualizzazione dello stato dei container Amazon ECS

Puoi visualizzare lo stato del contenitore nella console e utilizzare l'API nella `DescribeTasks` risposta. Per ulteriori informazioni, consulta il riferimento [DescribeTasks](#) all'API di Amazon Elastic Container Service.

Se utilizzi la registrazione per il tuo contenitore, ad esempio Amazon CloudWatch Logs, puoi configurare il comando health check per inoltrare l'output sullo stato del contenitore ai tuoi log. Assicurati di utilizzare `2&1` to catch sia `stdout` le `stderr` informazioni che.

```
"command": [  
    "CMD-SHELL",  
    "curl -f http://localhost/ >> /proc/1/fd/1 2>&1 || exit 1"  
],
```

Monitora lo stato delle istanze dei container Amazon ECS

Amazon ECS offre il monitoraggio dell'integrità delle istanze di container. Puoi determinare rapidamente se Amazon ECS ha rilevato problemi che potrebbero impedire alle istanze di container di eseguire i container. Amazon ECS esegue controlli automatici su ogni istanza di container in esecuzione con la versione `1.57.0` o successiva dell'agente per identificare i problemi. Per ulteriori informazioni sulla verifica della versione dell'agente su un'istanza di container, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

È necessario utilizzare la AWS CLI versione `1.22.3` o successiva o la AWS CLI versione `2.3.6` o successiva. Per informazioni su come aggiornare il AWS CLI, vedere [Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente versione 2](#).

Per visualizzare lo stato dell'istanza del contenitore, esegui `describe-container-instances` con l'`CONTAINER_INSTANCE_HEALTH` opzione.

Di seguito sono riportati i valori validi per `overallStatus`:

- OK
- IMPAIRED
- INSUFFICIENT_DATA
- INITIALIZING

Di seguito è riportato un esempio di come eseguire `describe-container-instances`.

```
aws ecs describe-container-instances \
  --cluster cluster_name \
  --container-instances 47279cd2cadb41cbaef2dcEXAMPLE \
  --include CONTAINER_INSTANCE_HEALTH
```

Di seguito è riportato un esempio dell'oggetto dello stato di integrità nell'output.

```
"healthStatus": {
  "overallStatus": "OK",
  "details": [{
    "type": "CONTAINER_RUNTIME",
    "status": "OK",
    "lastUpdated": "2021-11-10T03:30:26+00:00",
    "lastStatusChange": "2021-11-10T03:26:41+00:00"
  }]
}
```

Problemi di integrità dell'istanza del contenitore

Se lo stato è `overallStatus` diverso da `OK`, prova quanto segue:

- Attendi e poi corri `describe-container-instances`
- Visualizza lo stato dell'istanza del contenitore nella EC2 console o utilizzando la CLI.
- Rivedi le CloudWatch metriche. Per ulteriori informazioni, consulta [Monitora Amazon ECS utilizzando CloudWatch](#)
- Controlla se ci sono problemi con il servizio. AWS Health Dashboard

Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni

Amazon ECS si integra con AWS Distro per OpenTelemetry raccogliere dati di traccia dalla tua applicazione. Amazon ECS utilizza un contenitore AWS Distro for OpenTelemetry sidecar per raccogliere e indirizzare i dati di traccia. AWS X-Ray Per ulteriori informazioni, consulta [Configurazione di AWS Distro for OpenTelemetry Collector in Amazon ECS](#). Puoi quindi utilizzarlo AWS X-Ray per identificare errori ed eccezioni, analizzare i rallentamenti delle prestazioni e i tempi di risposta.

Affinché AWS Distro for OpenTelemetry Collector possa inviare i dati di traccia a AWS X-Ray, l'applicazione deve essere configurata per creare i dati di traccia. Per ulteriori informazioni, consulta [Strumentazione dell'applicazione per AWS X-Ray](#) nella AWS X-Ray Guida per gli sviluppatori.

Autorizzazioni IAM richieste per AWS Distro per l'integrazione con OpenTelemetry AWS X-Ray

L'integrazione di Amazon ECS con AWS Distro for OpenTelemetry richiede la creazione di un ruolo di attività e la specificazione del ruolo nella definizione dell'attività. Ti consigliamo di configurare AWS Distro for OpenTelemetry sidecar per indirizzare i log dei container verso Logs. CloudWatch

Important

Se raccogli anche le metriche delle applicazioni utilizzando AWS Distro per l'OpenTelemetry integrazione, assicurati che il ruolo Task IAM contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta [Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni](#).

Crea la seguente policy, quindi collegala al ruolo di esecuzione dell'attività.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:PutLogEvents",
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:DescribeLogGroups",
        "logs:PutRetentionPolicy",
        "xray:PutTraceSegments",
        "xray:PutTelemetryRecords",
        "xray:GetSamplingRules",
        "xray:GetSamplingTargets",
        "xray:GetSamplingStatisticSummaries",
        "ssm:GetParameters"
      ],
      "Resource": "*"
    }
  ]
}
```

6. Scegli Next (Successivo).

Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Specificare il AWS Distro for OpenTelemetry sidecar da AWS X-Ray integrare nella definizione dell'attività

La console Amazon ECS semplifica la creazione del contenitore AWS Distro for OpenTelemetry sidecar utilizzando l'opzione Use trace collection. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Se non utilizzi la console Amazon ECS, puoi aggiungere il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente frammento di definizione dell'attività mostra la definizione del contenitore per l'aggiunta del AWS Distro for sidecar for integration. OpenTelemetry AWS X-Ray

```
{
  "family": "otel-using-xray",
  "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryXrayRole",
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
  "containerDefinitions": [{
    "name": "aws-otel-emitter",
    "image": "application-image",
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-create-group": "true",
        "awslogs-group": "/ecs/aws-otel-emitter",
        "awslogs-region": "us-east-1",
        "awslogs-stream-prefix": "ecs"
      }
    },
    "dependsOn": [{
      "containerName": "aws-otel-collector",
      "condition": "START"
    }]
  }],
  {
    "name": "aws-otel-collector",
```

```
"image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
"essential": true,
"command": [
  "--config=/etc/ecs/otel-instance-metrics-config.yaml"
],
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-create-group": "True",
    "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
    "awslogs-region": "us-east-1",
    "awslogs-stream-prefix": "ecs"
  }
}
},
"networkMode": "awsvpc",
"requiresCompatibilities": [
  "FARGATE"
],
"cpu": "1024",
"memory": "3072"
}
```

Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni

Amazon ECS on Fargate supporta la raccolta di metriche dalle applicazioni in esecuzione su Fargate e l'esportazione su Amazon o CloudWatch Amazon Managed Service for Prometheus.

Puoi utilizzare i metadati raccolti per correlare i dati sulle prestazioni delle applicazioni con i dati dell'infrastruttura sottostante, riducendo il tempo medio necessario per risolvere il problema.

Amazon ECS utilizza un contenitore AWS Distro for OpenTelemetry sidecar per raccogliere e indirizzare i parametri dell'applicazione verso la destinazione. L'esperienza della console Amazon ECS semplifica il processo di aggiunta di questa integrazione durante la creazione delle definizioni delle attività.

Argomenti

- [Esportazione dei parametri delle applicazioni su Amazon CloudWatch](#)

- [Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus](#)

Esportazione dei parametri delle applicazioni su Amazon CloudWatch

Amazon ECS on Fargate supporta l'esportazione dei parametri delle applicazioni personalizzate su Amazon CloudWatch come parametri personalizzati. Questo viene fatto aggiungendo il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. La console Amazon ECS semplifica questo processo aggiungendo l'opzione Use metric collection durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

I parametri dell'applicazione vengono esportati in CloudWatch Logs con il nome del gruppo di log `/aws/ecs/application/metrics` e i parametri possono essere visualizzati nel namespace `ECS/AWSOTel/Application`. L'applicazione deve essere dotata dell'SDK. OpenTelemetry Per ulteriori informazioni, consulta [Introduzione a AWS Distro for OpenTelemetry in the AWS Distro per](#) la documentazione. OpenTelemetry

Considerazioni

Quando si utilizza l'integrazione di Amazon ECS on Fargate AWS con Distro OpenTelemetry per inviare i parametri delle applicazioni ad Amazon, è necessario considerare quanto segue. CloudWatch

- Questa integrazione invia solo i parametri dell'applicazione personalizzati a CloudWatch. Se desideri parametri a livello di attività, puoi attivare Container Insights nella configurazione del cluster Amazon ECS. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#).
- La AWS distribuzione per l' OpenTelemetry integrazione è supportata per i carichi di lavoro Amazon ECS ospitati su Fargate e i carichi di lavoro Amazon ECS ospitati su istanze Amazon. EC2. Al momento le istanze esterne non sono supportate.
- CloudWatch supporta un massimo di 30 dimensioni per metrica. Di default, Amazon ECS include le dimensioni `TaskARN`, `ClusterARN`, `LaunchType`, `TaskDefinitionFamily` e `TaskDefinitionRevision` nei parametri. Le restanti 25 dimensioni possono essere definite dalla tua applicazione. Se sono configurate più di 30 dimensioni, non è CloudWatch possibile visualizzarle. Quando ciò si verifica, le metriche dell'applicazione verranno visualizzate nello spazio dei nomi delle `ECS/AWSOTel/Application` CloudWatch metriche ma senza alcuna dimensione. Puoi strumentare la tua applicazione per aggiungere ulteriori dimensioni. Per ulteriori informazioni,

consulta [Utilizzo delle CloudWatch metriche con AWS Distro for in the Distro per la OpenTelemetry documentazione](#). AWS OpenTelemetry

Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon CloudWatch

L'integrazione di Amazon ECS con AWS Distro for OpenTelemetry richiede la creazione di un ruolo IAM dell'attività e la specificazione del ruolo nella definizione dell'attività. Consigliamo di configurare AWS Distro for OpenTelemetry sidecar anche per indirizzare i log dei container verso Logs, il che richiede la creazione e la specificazione di un ruolo IAM per CloudWatch l'esecuzione dell'attività e che venga specificato anche nella definizione dell'attività. La console Amazon ECS si occupa dell'esecuzione delle attività (ruolo IAM) per tuo conto, ma il ruolo IAM dell'attività deve essere creato manualmente e aggiunto alla definizione dell'attività. Per ulteriori informazioni sul ruolo IAM di esecuzione dei processi, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Important

Se stai anche raccogliendo dati di tracciamento delle applicazioni utilizzando AWS Distro per OpenTelemetry l'integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta [Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni](#).

Se l'applicazione richiede ulteriori autorizzazioni, è necessario aggiungerle a questa policy. Ciascuna definizione di attività può specificare un solo ruolo IAM del processo. Ad esempio, se utilizzi un file di configurazione personalizzato memorizzato in Systems Manager, devi aggiungere l'autorizzazione `ssm:GetParameters` a questa policy IAM.

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.

5. Scegli Next (Successivo).
 6. Nella sezione Aggiungi autorizzazioni, cerca `AWSDistroOpenTelemetryPolicyForXray`, quindi seleziona la politica.
 7. (Facoltativo) Impostare un [limite delle autorizzazioni](#). Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.
 - a. Apri la sezione Imposta limite delle autorizzazioni e seleziona Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo.
- IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.
- b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
8. Scegli Next (Successivo).
9. Inserisci il nome del ruolo o il suffisso del nome del ruolo per aiutarti a identificare lo scopo del ruolo.

 Important

Quando assegni un nome a un ruolo, tieni presente quanto segue:

- I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS profilo e non possono essere resi unici per caso.

Ad esempio, non creare ruoli denominati **PRODRole** e **prodrole**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.

- Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.
11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, in Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni seleziona Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consulta [Tags for AWS Identity and Access Management resources](#) nella IAM User Guide.

13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Specificare la AWS Distro for OpenTelemetry sidecar nella definizione dell'attività

La console Amazon ECS semplifica l'esperienza di creazione del contenitore AWS Distro for OpenTelemetry sidecar utilizzando l'opzione Use metric collection. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Se non utilizzi la console Amazon ECS, puoi aggiungere manualmente il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente esempio di definizione delle attività mostra la definizione del contenitore per l'aggiunta di AWS Distro for OpenTelemetry sidecar for Amazon CloudWatch integration.

```
{
  "family": "otel-using-cloudwatch",
  "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryCloudWatchRole",
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "name": "aws-otel-emitter",
      "image": "application-image",
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-create-group": "true",
          "awslogs-group": "/ecs/aws-otel-emitter",
          "awslogs-region": "us-east-1",
          "awslogs-stream-prefix": "ecs"
        }
      },
    },
    {
      "dependsOn": [{
        "containerName": "aws-otel-collector",
        "condition": "START"
      }]
    },
    {
      "name": "aws-otel-collector",
      "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
      "essential": true,
      "command": [
        "--config=/etc/ecs/ecs-cloudwatch.yaml"
      ],
    },
  ],
}
```



```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-create-group": "True",
    "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
    "awslogs-region": "us-east-1",
    "awslogs-stream-prefix": "ecs"
  }
},
"networkMode": "awsvpc",
"requiresCompatibilities": [
  "FARGATE"
],
"cpu": "1024",
"memory": "3072"
}
```

Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus

Amazon ECS supporta l'esportazione di parametri di CPU, memoria, rete e archiviazione a livello di processo e i parametri delle applicazioni personalizzate in Amazon Managed Service for Prometheus. Questo viene fatto aggiungendo il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. La console Amazon ECS semplifica questo processo aggiungendo l'opzione Use metric collection durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

I parametri vengono esportati in Amazon Managed Service for Prometheus e possono essere visualizzati utilizzando il pannello di controllo di Amazon Managed Grafana. L'applicazione deve essere dotata di strumenti con le librerie Prometheus o con l'SDK. OpenTelemetry Per ulteriori informazioni sulla strumentazione dell'applicazione con l' OpenTelemetry SDK, consulta [Introduzione a Distro per AWS](#) la documentazione nella versione Distro. OpenTelemetry AWS OpenTelemetry

Quando si utilizzano le librerie Prometheus, l'applicazione deve esporre un endpoint `/metrics` utilizzato per lo scraping dei dati dei parametri. Per ulteriori informazioni sulla strumentazione della tua applicazione con le librerie Prometheus, consulta [Librerie client Prometheus](#) nella documentazione di Prometheus.

Considerazioni

Quando si utilizza l'integrazione di Amazon ECS on Fargate AWS con Distro OpenTelemetry per inviare i parametri delle applicazioni ad Amazon Managed Service for Prometheus, è necessario considerare quanto segue.

- La AWS distribuzione per l' OpenTelemetry integrazione è supportata per i carichi di lavoro Amazon ECS ospitati su Fargate e i carichi di lavoro Amazon ECS ospitati su istanze Amazon. EC2 Attualmente le istanze esterne non sono supportate.
- Per impostazione predefinita, AWS Distro for OpenTelemetry include tutte le dimensioni a livello di attività disponibili per le metriche dell'applicazione durante l'esportazione in Amazon Managed Service for Prometheus. Puoi anche strumentare la tua applicazione per aggiungere ulteriori dimensioni. Per ulteriori informazioni, consulta la sezione [Guida introduttiva a Prometheus Remote Write Exporter for Amazon Managed Service for Prometheus in the Distro per la documentazione.](#) AWS OpenTelemetry

Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon Managed Service for Prometheus

L'integrazione di Amazon ECS con Amazon Managed Service for Prometheus utilizzando Distro OpenTelemetry for sidecar richiede AWS la creazione di un ruolo IAM dell'attività e la specificazione del ruolo nella definizione dell'attività. Questo ruolo IAM del processo deve essere creato manualmente utilizzando la procedura riportata di seguito prima di registrare la definizione di attività.

Consigliamo di configurare AWS Distro for OpenTelemetry sidecar anche per indirizzare i log dei container verso Logs, il che richiede la creazione di un ruolo IAM per CloudWatch l'esecuzione delle attività e la specificazione anche nella definizione dell'attività. La console Amazon ECS si occupa dell'esecuzione delle attività (ruolo IAM) per tuo conto, ma il ruolo IAM dell'attività deve essere creato manualmente. Per ulteriori informazioni sulla creazione di un ruolo IAM di creazione dei processi, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Important

Se stai anche raccogliendo dati di tracciamento delle applicazioni utilizzando AWS Distro per OpenTelemetry l'integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta [Identifica](#)

le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni.

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Scegli Next (Successivo).
6. Nella sezione Aggiungi autorizzazioni, cerca AmazonPrometheusRemoteWriteAccess, quindi seleziona la politica.
7. (Facoltativo) Impostare un [limite delle autorizzazioni](#). Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.
 - a. Apri la sezione Imposta limite delle autorizzazioni e seleziona Usa un limite delle autorizzazioni per controllare il numero massimo di autorizzazioni del ruolo.

IAM include un elenco delle politiche AWS gestite e gestite dal cliente nel tuo account.
 - b. Selezionare la policy da utilizzare per il limite delle autorizzazioni.
8. Scegli Next (Successivo).
9. Inserisci il nome del ruolo o il suffisso del nome del ruolo per aiutarti a identificare lo scopo del ruolo.

Important

Quando assegni un nome a un ruolo, tieni presente quanto segue:

- I nomi dei ruoli devono essere univoci all'interno del tuo Account AWS profilo e non possono essere resi unici per caso.

Ad esempio, non creare ruoli denominati **PRODRROLE** e **prodrole**. Quando il nome di un ruolo viene utilizzato in una policy o come parte di un ARN, il nome del ruolo fa distinzione tra maiuscole e minuscole, tuttavia quando un nome di ruolo viene

visualizzato ai clienti nella console, ad esempio durante il processo di accesso, il nome del ruolo non fa distinzione tra maiuscole e minuscole.

- Non è possibile modificare il nome del ruolo dopo averlo creato, in quanto altre entità possono fare riferimento al ruolo.

10. (Facoltativo) In Descrizione, inserisci una descrizione per il ruolo.
11. (Facoltativo) Per modificare i casi d'uso e le autorizzazioni per il ruolo, in Fase 1: seleziona le entità attendibili o Fase 2: aggiungi autorizzazioni seleziona Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi i tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo dei tag in IAM, consulta [Tags for AWS Identity and Access Management resources](#) nella IAM User Guide.
13. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

Specificare la AWS Distro for OpenTelemetry sidecar nella definizione dell'attività

La console Amazon ECS semplifica l'esperienza di creazione del contenitore AWS Distro for OpenTelemetry sidecar utilizzando l'opzione Use metric collection. Per ulteriori informazioni, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Se non utilizzi la console Amazon ECS, puoi aggiungere manualmente il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente esempio di definizione di attività mostra la definizione del contenitore per l'aggiunta di AWS Distro for OpenTelemetry sidecar per l'integrazione con Amazon Managed Service for Prometheus.

```
{
  "family": "otel-using-cloudwatch",
  "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryCloudWatchRole",
  "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
  "containerDefinitions": [{
    "name": "aws-otel-emitter",
    "image": "application-image",
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-create-group": "true",
        "awslogs-group": "/ecs/aws-otel-emitter",
        "awslogs-region": "aws-region",
        "awslogs-stream-prefix": "ecs"
      }
    }
  ]
}
```

```

    },
    "dependsOn": [{
      "containerName": "aws-otel-collector",
      "condition": "START"
    }]
  },
  {
    "name": "aws-otel-collector",
    "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
    "essential": true,
    "command": [
      "--config=/etc/ecs/ecs-amp.yaml"
    ],
    "environment": [{
      "name": "AWS_PROMETHEUS_ENDPOINT",
      "value": "https://aps-workspaces.aws-region.amazonaws.com/workspaces/
ws-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
    }],
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-create-group": "True",
        "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
        "awslogs-region": "aws-region",
        "awslogs-stream-prefix": "ecs"
      }
    }
  }
],
"networkMode": "awsvpc",
"requiresCompatibilities": [
  "FARGATE"
],
"cpu": "1024",
"memory": "3072"
}

```

Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail

Amazon Elastic Container Service è integrato con [AWS CloudTrail](#), un servizio che fornisce un registro delle azioni intraprese da un utente, ruolo o un Servizio AWS. CloudTrail acquisisce tutte le

chiamate API per Amazon ECS come eventi. Le chiamate acquisite includono chiamate dalla console Amazon ECS e chiamate in codice alle operazioni dell'API Amazon ECS. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta effettuata ad Amazon ECS, l'indirizzo IP da cui è stata effettuata la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con le credenziali utente root o utente.
- Se la richiesta è stata effettuata per conto di un utente del Centro identità IAM.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro Servizio AWS.

CloudTrail è attivo nel tuo account Account AWS quando crei l'account e hai automaticamente accesso alla cronologia degli CloudTrail eventi. La cronologia CloudTrail degli eventi fornisce un record visualizzabile, ricercabile, scaricabile e immutabile degli ultimi 90 giorni di eventi di gestione registrati in un. Regione AWS Per ulteriori informazioni, consulta [Lavorare con la cronologia degli CloudTrail eventi](#) nella Guida per l'utente AWS CloudTrail. Non sono CloudTrail previsti costi per la visualizzazione della cronologia degli eventi.

Per una registrazione continua degli eventi degli Account AWS ultimi 90 giorni, crea un trail o un data store di eventi [CloudTrailLake](#).

CloudTrail sentieri

Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Tutti i percorsi creati utilizzando il AWS Management Console sono multiregionali. È possibile creare un trail per una singola Regione o per più Regioni tramite AWS CLI. La creazione di un percorso multiregionale è consigliata in quanto consente di registrare l'intera attività del proprio Regioni AWS account. Se si crea un trail per una singola Regione, è possibile visualizzare solo gli eventi registrati nella Regione AWS del trail. Per ulteriori informazioni sui trail, consulta [Creating a trail for your Account AWS](#) e [Creating a trail for an organization](#) nella Guida per l'utente di AWS CloudTrail.

Puoi inviare gratuitamente una copia dei tuoi eventi di gestione in corso al tuo bucket Amazon S3 CloudTrail creando un percorso, tuttavia ci sono costi di storage di Amazon S3. [Per ulteriori informazioni sui CloudTrail prezzi, consulta la pagina Prezzi.AWS CloudTrail](#) Per informazioni sui prezzi di Amazon S3, consulta [Prezzi di Amazon S3](#).

CloudTrail Archivi di dati sugli eventi di Lake

CloudTrail Lake ti consente di eseguire query basate su SQL sui tuoi eventi. CloudTrail [Lake converte gli eventi esistenti in formato JSON basato su righe in formato Apache ORC](#). ORC è un formato di archiviazione a colonne ottimizzato per il recupero rapido dei dati. Gli eventi vengono aggregati in archivi di dati degli eventi, che sono raccolte di eventi immutabili basate sui criteri selezionati applicando i [selettori di eventi avanzati](#). I selettori applicati a un archivio di dati degli eventi controllano quali eventi persistono e sono disponibili per l'esecuzione della query. Per ulteriori informazioni su CloudTrail Lake, consulta [Working with AWS CloudTrail Lake](#) nella Guida per l'utente.AWS CloudTrail

CloudTrail Gli archivi e le richieste di dati sugli eventi di Lake comportano dei costi. Quando crei un datastore di eventi, scegli l'[opzione di prezzo](#) da utilizzare per tale datastore. L'opzione di prezzo determina il costo per l'importazione e l'archiviazione degli eventi, nonché il periodo di conservazione predefinito e quello massimo per il datastore di eventi. [Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione Prezzi.AWS CloudTrail](#)

Eventi di gestione di Amazon ECS in CloudTrail

[Gli eventi](#) di gestione forniscono informazioni sulle operazioni di gestione eseguite sulle risorse del tuo Account AWS. Queste operazioni sono definite anche operazioni del piano di controllo (control-plane). Per impostazione predefinita, CloudTrail registra gli eventi di gestione.

Amazon Elastic Container Service registra tutte le operazioni del piano di controllo di Amazon ECS come eventi di gestione. Ad esempio, le chiamate alle `CreateService` `DeleteCluster` sezioni `RunTask` e generano voci nei file di CloudTrail registro. Per un elenco delle operazioni del piano di controllo di Amazon Elastic Container Service a cui Amazon ECS accede CloudTrail, consulta l'[Amazon Elastic Container Service API](#) Reference.

Esempi di eventi Amazon ECS

Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'operazione API richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia stack ordinata delle chiamate API pubbliche, quindi gli eventi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra un CloudTrail evento che dimostra l'`CreateService`azione.

Note

Questo esempio è stato formattato per migliorare la leggibilità. In un file di CloudTrail registro, tutte le voci e gli eventi sono concatenati in un'unica riga. Inoltre, questo esempio è limitato a una singola voce Amazon ECS. In un vero file di CloudTrail registro, puoi vedere voci ed eventi provenienti da più servizi. AWS

```
{
  "eventVersion": "1.04",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:account_name",
    "arn": "arn:aws:sts::123456789012:user/Mary_Major",
    "accountId": "123456789012",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2018-06-20T18:32:25Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::123456789012:role/Admin",
        "accountId": "123456789012",
        "userName": "Mary_Major"
      }
    }
  },
  "eventTime": "2018-06-20T19:04:36Z",
  "eventSource": "ecs.amazonaws.com",
  "eventName": "CreateCluster",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.12",
  "userAgent": "console.amazonaws.com",
  "requestParameters": {
    "clusterName": "default"
  },
  "responseElements": {
    "cluster": {
      "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/default",
```



```
        "pendingTasksCount": 0,  
        "registeredContainerInstancesCount": 0,  
        "status": "ACTIVE",  
        "runningTasksCount": 0,  
        "statistics": [],  
        "clusterName": "default",  
        "activeServicesCount": 0  
    },  
    },  
    "requestID": "cb8c167e-EXAMPLE",  
    "eventID": "e3c6f4ce-EXAMPLE",  
    "eventType": "AwsApiCall",  
    "recipientAccountId": "123456789012"  
}
```

Per informazioni sul contenuto dei CloudTrail record, consultate il [contenuto dei CloudTrail record](#) nella Guida AWS CloudTrail per l'utente.

Monitora i carichi di lavoro utilizzando i metadati Amazon ECS

È possibile utilizzare i metadati delle attività e dei contenitori per risolvere i problemi dei carichi di lavoro e apportare modifiche alla configurazione in base all'ambiente di runtime.

I metadati includono le seguenti categorie:

- Attributi a livello di attività che forniscono informazioni sulla posizione in cui l'attività è in esecuzione.
- Attributi a livello di contenitore che forniscono l'ID Docker, il nome e i dettagli dell'immagine.

Ciò fornisce visibilità all'interno del contenitore.

- Impostazioni di rete come indirizzi IP, sottoreti e modalità di rete.

Ciò facilita la configurazione e la risoluzione dei problemi della rete.

- Stato e integrità dell'attività

Ciò consente di sapere se le attività sono in esecuzione.

È possibile visualizzare i metadati con uno dei seguenti metodi:

- File di metadati del container

A partire dalla versione 1.15.0 dell'agente di container di Amazon ECS, vari metadati di container sono disponibili all'interno dei contenitori o dell'istanza di container host. Attivando questa caratteristica, è possibile eseguire una query delle informazioni su un'attività, un container e un'istanza di container all'interno del container stesso o dell'istanza di container host. Il file di metadati viene creato sull'istanza host e montato nel container come volume Docker e pertanto non è disponibile quando un processo è ospitato su AWS Fargate.

- Endpoint di metadati delle attività

L'agente del container Amazon ECS inserisce una variabile di ambiente in ogni container denominata endpoint dei metadati dei processi che fornisce vari metadati di processi e [statistiche Docker](#) al container.

- Introspezione del contenitore

L'agente del container di Amazon ECS fornisce un'operazione API per raccogliere i dettagli sull'istanza di container su cui è in esecuzione l'agente e sui processi associati in esecuzione su tale istanza.

Variabili di ambiente Amazon ECS

Le variabili di ambiente controllano il comportamento di un'attività, ad esempio la capacità utilizzata per eseguirla. Amazon ECS imposta le seguenti variabili di ambiente per le tue attività:

- `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`- L'endpoint URL HTTP completo per l'SDK da utilizzare quando si effettua una richiesta di credenziali. Ciò include sia lo schema che l'host. Per ulteriori informazioni, consulta [Container credential provider](#) nella Guida AWS SDKs di riferimento.
- `ECS_CONTAINER_METADATA_URI_V4`- L'indirizzo della versione 4 dei metadati dell'attività. Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 4](#).
- `ECS_CONTAINER_METADATA_URI`- L'indirizzo della versione 3 dei metadati dell'attività. Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 3](#).
- `ECS_AGENT_URI`- L'indirizzo di base per i diversi endpoint supportati da Fargate. Per ulteriori informazioni, consultare:
 - [Endpoint di protezione scalabile per attività Amazon ECS](#)
 - [Endpoint di iniezione di errori Amazon ECS](#)
- `AWS_EXECUTION_ENV`- Le informazioni sul tipo di avvio su cui viene eseguita l'attività.

- Per Fargate, Amazon ECS lo imposta su. `AWS_ECS_FARGATE`
- Infatti EC2, Amazon ECS lo imposta su `AWS_ECS_EC2`.
- `AWS_DEFAULT_REGION`— L'impostazione predefinita Regione AWS per il tuo Account AWS. Questa è la regione predefinita in cui viene eseguita l'attività.
- `AWS_REGION`— La regione in cui viene eseguita l'attività. Se immesso, questo valore sovrascrive `AWS_DEFAULT_REGION`.

È possibile visualizzare le variabili di ambiente nei metadati dell'attività. Per ulteriori informazioni, consulta uno dei seguenti argomenti:

- Fargate - [Metadati delle attività di Amazon ECS disponibili per le attività su Fargate](#)
- EC2 - [Metadati delle attività disponibili per le attività di Amazon ECS su EC2](#)

File di metadati di container di Amazon ECS

A partire dalla versione 1.15.0 dell'agente di container di Amazon ECS, vari metadati di container sono disponibili all'interno dei contenitori o dell'istanza di container host. Attivando questa caratteristica, è possibile eseguire una query delle informazioni su un'attività, un container e un'istanza di container all'interno del container stesso o dell'istanza di container host. Il file di metadati viene creato sull'istanza host e montato nel contenitore come volume Docker e pertanto non è disponibile quando un'attività è ospitata su AWS Fargate.

Il file di metadati di container viene pulito sull'istanza host quando avviene lo stesso per il container. Puoi definire il momento in cui ciò accade tramite la variabile `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION` dell'agente di container. Per ulteriori informazioni, consulta [Pulizia automatica delle attività e delle immagini di Amazon ECS](#).

Argomenti

- [Posizioni dei file di metadati dei container](#)
- [Attivazione dei metadati dei contenitori Amazon ECS](#)
- [Formato di file di metadati del contenitore Amazon ECS](#)

Posizioni dei file di metadati dei container

Di default, il file di metadati di container viene scritto sui percorsi di host e di container seguenti.

- Per le istanze Linux:
 - Percorso host: `/var/lib/ecs/data/metadata/cluster_name/task_id/container_name/ecs-container-metadata.json`

 Note

Il percorso dell'host di Linux presuppone che venga utilizzato il percorso di montaggio della directory dei dati predefinito (`/var/lib/ecs/data`) all'avvio dell'agente. Se non utilizzi l'AMI ottimizzata per Amazon ECS (o il pacchetto `ecs-init` per avviare e mantenere l'agente di container), assicurati di impostare la variabile di configurazione dell'agente `ECS_HOST_DATA_DIR` sul percorso dell'host in cui è ubicato il file di stato dell'agente del container. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

- Percorso container: `/opt/ecs/metadata/random_ID/ecs-container-metadata.json`
- Per le istanze Windows:
 - Percorso host: `C:\ProgramData\Amazon\ECS\data\metadata\ task_id \ container_name \ecs-container-metadata.json`
 - Percorso container: `C:\ProgramData\Amazon\ECS\metadata\ random_ID \ecs-container-metadata.json`

Tuttavia, per semplificare l'accesso, l'ubicazione del file di metadati di container è impostata sulla variabile di ambiente `ECS_CONTAINER_METADATA_FILE` all'interno del container. Puoi leggere il file contenuti dall'interno del container attraverso il comando seguente:

- Per le istanze Linux:

```
cat $ECS_CONTAINER_METADATA_FILE
```

- Per le istanze Windows (): PowerShell

```
Get-Content -path $env:ECS_CONTAINER_METADATA_FILE
```

Attivazione dei metadati dei contenitori Amazon ECS

È possibile attivare i metadati di container a livello di istanza di container impostando la variabile dell'agente di container `ECS_ENABLE_CONTAINER_METADATA` su `true`. Puoi impostare questa variabile nel file di configurazione `/etc/ecs/ecs.config` e riavviare l'agente. Puoi inoltre impostarla come una variabile di ambiente Docker al runtime quando viene avviato l'agente di container. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Se l'opzione `ECS_ENABLE_CONTAINER_METADATA` è impostata su `true` quando viene avviato l'agente, i file di metadati vengono creati per tutti i container creati da quel punto in avanti. L'agente di container di Amazon ECS non è in grado di creare file di metadati per container creati prima dell'impostazione della variabile dell'agente di container `ECS_ENABLE_CONTAINER_METADATA` su `true`. Per assicurarsi che tutti i container ricevano i file di metadati, è necessario impostare questa variabile dell'agente all'avvio dell'istanza di container. Di seguito è riportato uno script di dati utente di esempio che imposterà questa variabile e registrerà l'istanza di container con il cluster.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=your_cluster_name
ECS_ENABLE_CONTAINER_METADATA=true
EOF
```

Formato di file di metadati del contenitore Amazon ECS

Le informazioni seguenti vengono memorizzate nel file JSON di metadati del container.

Cluster

Il nome del cluster su cui viene eseguita l'attività di container.

ContainerInstanceARN

L'Amazon Resource Name (ARN) completo dell'istanza di container dell'host.

TaskARN

L'Amazon Resource Name (ARN) completo dell'attività a cui appartiene il container.

TaskDefinitionFamily

Il nome della famiglia di definizioni di processo utilizzata dal container.

TaskDefinitionRevision

La revisione della definizione di attività utilizzata dal container.

ContainerID

L'ID del container Docker (non si tratta dell'ID del container di Amazon ECS) per il container.

ContainerName

Il nome del container dalla definizione di attività di Amazon ECS per il container.

DockerContainerName

Il nome del container che il daemon Docker utilizza per il container (ad esempio, il nome che viene visualizzato nell'output del comando `docker ps`).

ImageID

Il digest SHA dell'immagine Docker utilizzato per avviare il container.

ImageName

Il nome e il tag dell'immagine Docker utilizzati per avviare il container.

PortMappings

Qualsiasi mappatura delle porte associate al container.

ContainerPort

La porta del container esposto.

HostPort

La porta dell'istanza di container host esposto.

BindIp

L'indirizzo IP di associazione assegnato al container da Docker. Questo indirizzo IP viene applicato solo con la modalità di rete `bridge` ed è accessibile solo dall'istanza di container.

Protocol

Il protocollo di rete utilizzato per la mappatura delle porte.

Networks

Le modalità di rete e l'indirizzo IP del container.

NetworkMode

Le modalità di rete dell'attività a cui appartiene il container.

IPv4Addresses

Gli indirizzi IP associati con il container.

Important

Se l'attività utilizza la modalità di rete `awsvpc`, l'indirizzo IP del container non verrà restituito. In questo caso, puoi recuperare l'indirizzo IP leggendo il file `/etc/hosts` con il comando seguente:

```
tail -1 /etc/hosts | awk '{print $1}'
```

MetadataFileStatus

Lo stato del file di metadati. Quando lo stato è `READY`, il file di metadati è corrente e completo. Se il file non è ancora pronto (ad esempio, nel momento in cui l'attività viene avviata), è disponibile una versione troncata del formato di file. Per evitare una probabile race condition in cui il container viene avviato, ma i metadati non sono ancora stati scritti, puoi analizzare il file di metadati e attendere che questo parametro sia impostato su `READY` prima di dipendere dai metadati. In genere, ciò è disponibile in meno di 1 secondo dall'avvio del container.

AvailabilityZone

Zona di disponibilità in cui risiede l'istanza di container host.

HostPrivateIPv4Address

L'indirizzo IP privato per l'attività a cui appartiene il container.

HostPublicIPv4Address

L'indirizzo IP pubblico per l'attività a cui appartiene il container.

Example File di metadati del container di Amazon ECS (**READY**)

L'esempio seguente mostra un file di metadati di container in stato `READY`.

```
{
```

```

    "Cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
    "TaskARN": "arn:aws:ecs:us-east-1:123456789012:task/MyCluster/
b593651c4d6b44a6b2b583f45c957e15",
    "Family": "curltest-container",
    "Revision": "2",
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits":
    {
        "CPU": 0.25,
        "Memory": 512
    },
    "PullStartedAt": "2025-01-17T20:56:17.394610044Z",
    "PullStoppedAt": "2025-01-17T20:56:25.282708213Z",
    "AvailabilityZone": "us-east-1b",
    "LaunchType": "FARGATE",
    "Containers": [
    {
        "DockerId": "b593651c4d6b44a6b2b583f45c957e15-3356213583",
        "Name": "curltest", "DockerName": "curltest",
        "Image": "public.ecr.aws/amazonlinux/amazonlinux:latest",

"ImageID": "sha256:7f371357694782356b65c7fd60dd1ca124c47bd5ed1b1ffe7c0e17f562898367",
        "Labels":
        {
            "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-
east-1:123456789012:cluster/MyCluster",
            "com.amazonaws.ecs.container-name": "curltest",
            "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-1:123456789012:task/MyCluster/b593651c4d6b44a6b2b583f45c957e15",
            "com.amazonaws.ecs.task-definition-family": "curltest-
container", "com.amazonaws.ecs.task-definition-version": "2"
        },
        "DesiredStatus": "RUNNING",
        "KnownStatus": "RUNNING",
        "Limits":
        {
            "CPU": 2
        },
        "CreatedAt": "2025-01-17T20:56:26.180347056Z",
        "StartedAt": "2025-01-17T20:56:26.180347056Z",
        "Type": "NORMAL",
        "LogDriver": "awslogs",
        "LogOptions":

```



```

        {
            "awslogs-create-group": "true",
            "awslogs-group": "/ecs/curltest-container",
            "awslogs-region": "us-east-1",
            "awslogs-stream": "ecs/curltest/b593651c4d6b44a6b2b583f45c957e15"
        },
        "ContainerARN": "arn:aws:ecs:us-east-1:123456789012:container/MyCluster/
b593651c4d6b44a6b2b583f45c957e15/934575e8-5bdb-478f-b763-2341a85b690e",
        "Networks": [
            {
                "NetworkMode": "awsvpc",
                "IPv4Addresses": ["10.0.1.58"]
            }
        ],
        "Snapshotter": "overlayfs"
    },
    "ClockDrift": {
        "ClockErrorBound": 0.487801, "ReferenceTimestamp": "2025-01-17T20:56:02Z",
        "ClockSynchronizationStatus": "SYNCHRONIZED"
    },
    "FaultInjectionEnabled": false
}

```

Example File di metadati di container Amazon ECS incompleto (non ancora **READY**)

L'esempio seguente mostra un file di metadati di container che non ha ancora raggiunto lo stato **READY**. Le informazioni incluse nel file sono limitate a pochi parametri noti dalla definizione di attività. In genere, il file di metadati di container è disponibile entro 1 secondo dopo l'avvio del container.

```

{
    "Cluster": "default",
    "ContainerInstanceARN": "arn:aws:ecs:us-west-2:012345678910:container-instance/
default/1f73d099-b914-411c-a9ff-81633b7741dd",
    "TaskARN": "arn:aws:ecs:us-west-2:012345678910:task/default/
d90675f8-1a98-444b-805b-3d9cabb6fcd4",
    "ContainerName": "metadata"
}

```

Metadati delle attività disponibili per le attività di Amazon ECS su EC2

L'agente del container di Amazon ECS fornisce un metodo per recuperare vari metadati dei processi e [statistiche Docker](#). Questo metodo è noto come endpoint dei metadati dei processi. Sono disponibili le seguenti versioni:

- Endpoint dei metadati dei processi versione 4: fornisce una varietà di metadati e statistiche Docker ai container. Può anche fornire dati sulla velocità di rete. Disponibile per le attività di Amazon ECS avviate su istanze Amazon EC2 Linux che eseguono almeno una versione 1.39.0 dell'agente container Amazon ECS. Per le istanze Amazon EC2 Windows che utilizzano la modalità di awsvpc rete, l'agente contenitore Amazon ECS deve avere almeno una versione 1.54.0 Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 4](#).
- Endpoint dei metadati dei processi versione 3: fornisce una varietà di metadati e statistiche Docker ai container. Disponibile per le attività di Amazon ECS avviate su istanze Amazon EC2 Linux che eseguono almeno una versione 1.21.0 dell'agente container Amazon ECS. Per le istanze Amazon EC2 Windows che utilizzano la modalità di awsvpc rete, l'agente contenitore Amazon ECS deve avere almeno una versione 1.54.0 Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 3](#).
- Endpoint di metadati delle attività versione 2: disponibile per le attività di Amazon ECS avviate su istanze Amazon EC2 Linux che eseguono almeno una versione 1.17.0 dell'agente container Amazon ECS. Per le istanze Amazon EC2 Windows che utilizzano la modalità di awsvpc rete, l'agente contenitore Amazon ECS deve avere almeno una versione 1.54.0 Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 2](#).

Se la tua attività Amazon ECS è ospitata su Amazon EC2, puoi anche accedere ai metadati dell'host delle attività utilizzando l'endpoint [Instance Metadata Service \(IMDS\)](#). Il comando seguente, se eseguito dall'istanza che ospita l'attività, elenca l'ID dell'istanza host.

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

Le informazioni che è possibile ottenere dall'endpoint sono suddivise in categorie, ad esempio *instance-id*. [Per ulteriori informazioni sulle diverse categorie di metadati delle istanze host che puoi ottenere utilizzando l'endpoint, consulta Categorie di metadati delle istanze.](#)

Endpoint di metadati delle attività Amazon ECS versione 4

L'agente del container Amazon ECS inserisce una variabile di ambiente in ogni container denominata endpoint dei metadati dei processi che fornisce vari metadati di processi e [statistiche Docker](#) al container.

I metadati delle attività e le statistiche sulla velocità di rete vengono inviati a CloudWatch Container Insights e possono essere visualizzati in AWS Management Console. Per ulteriori informazioni, consulta [Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata](#).

Note

Amazon ECS offre versioni precedenti dell'endpoint dei metadati dei processi. Per evitare la necessità di creare nuove versioni degli endpoint dei metadati delle attività in futuro, ulteriori metadati potrebbero essere aggiunti all'output della versione 4. Non rimuoveremo i metadati esistenti né modificheremo i nomi dei campi dei metadati.

La variabile di ambiente viene iniettata per impostazione predefinita nei contenitori delle attività di Amazon ECS avviate su istanze Amazon EC2 Linux che eseguono almeno una versione 1.39.0 dell'agente container Amazon ECS. Per le istanze Amazon EC2 Windows che utilizzano la modalità di awsipc rete, l'agente contenitore Amazon ECS deve avere almeno una versione 1.54.0. Per ulteriori informazioni, consulta [Gestione delle istanze di container Amazon ECS Linux](#).

Note

Puoi aggiungere il supporto per questa funzionalità sulle EC2 istanze Amazon che utilizzano versioni precedenti dell'agente container Amazon ECS aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Percorsi dell'endpoint dei metadati dei processi versione 4

Per i container sono disponibili i seguenti endpoint dei metadati dei processi:

```
${ECS_CONTAINER_METADATA_URI_V4}
```

Questo percorso restituisce il file JSON dei metadati per il container.

`${ECS_CONTAINER_METADATA_URI_V4}/task`

Questo percorso restituisce i metadati per l'attività, incluso un elenco del contenitore IDs e i nomi di tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON V4 dei metadati delle attività di Amazon ECS](#).

`${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags`

Questo percorso restituisce i metadati per il processo incluso nell'endpoint `/task` oltre ai tag dell'istanza di processo e container che possono essere recuperati utilizzando l'API `ListTagsForResource`. Eventuali errori ricevuti durante il recupero dei metadati del tag saranno inclusi nel campo `Errors` della risposta.

 Note

Il `Errors` campo è solo nella risposta per le attività ospitate su istanze Amazon EC2 Linux che eseguono almeno una versione `1.50.0` dell'agente contenitore. Per le istanze Amazon EC2 Windows che utilizzano la modalità di `awsipc` rete, l'agente contenitore Amazon ECS deve avere almeno una versione `1.54.0`. Questo endpoint richiede l'autorizzazione `ecs:ListTagsForResource`.

`${ECS_CONTAINER_METADATA_URI_V4}/stats`

Questo percorso restituisce le statistiche Docker per il container specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

Per le attività di Amazon ECS che utilizzano le modalità di `bridge` rete `awsipc` o ospitate su istanze Amazon EC2 Linux che eseguono almeno una versione `1.43.0` dell'agente container, nella risposta saranno incluse statistiche aggiuntive sulla velocità di rete. Per tutti gli altri processi, la risposta includerà solo le statistiche cumulative di rete.

`${ECS_CONTAINER_METADATA_URI_V4}/task/stats`

Questo percorso restituisce le statistiche Docker per tutti i container associati al processo. Questo può essere utilizzato dai container sidecar per estrarre i parametri di rete. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

Per le attività di Amazon ECS che utilizzano le modalità di `bridge` rete `awsipc` o ospitate su istanze Amazon EC2 Linux che eseguono almeno una versione `1.43.0` dell'agente container,

nella risposta saranno incluse statistiche aggiuntive sulla velocità di rete. Per tutti gli altri processi, la risposta includerà solo le statistiche cumulative di rete.

Risposta JSON V4 dei metadati delle attività di Amazon ECS

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`${ECS_CONTAINER_METADATA_URI_V4}/task`) restituisce le seguenti informazioni. Oltre ai metadati per ogni container all'interno del processo, sono inclusi anche i metadati associati al processo.

Cluster

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

ServiceName

Il nome del servizio a cui appartiene l'attività. `ServiceName` verrà visualizzato per le istanze di container Amazon EC2 e Amazon ECS Anywhere se l'attività è associata a un servizio.

Note

Questi metadati `ServiceName` sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione 1.63.1 o successiva.

VPCID

L'ID VPC dell'istanza del EC2 contenitore Amazon. Questo campo viene visualizzato solo per le EC2 istanze Amazon.

Note

Questi metadati `VPCID` sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione 1.63.1 o successiva.

TaskARN

L'Amazon Resource Name (ARN) dell'attività a cui appartiene il contenitore.

Family

La famiglia della definizione di attività Amazon ECS per il processo.

Revision

La revisione della definizione di attività di Amazon ECS per il processo.

DesiredStatus

Lo stato desiderato per il processo da Amazon ECS.

KnownStatus

Lo stato noto per il processo da Amazon ECS.

Limits

I limiti di risorse specificati a livello di attività, come CPU (espressa in vCPUs) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

PullStartedAt

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

PullStoppedAt

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

AvailabilityZone

La zona di disponibilità in cui si trova l'attività.

Note

I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 (Windows).

LaunchType

Il tipo di avvio utilizzato dall'attività. Quando si utilizzano provider di capacità del cluster, ciò indica se l'attività utilizza Fargate o EC2 l'infrastruttura.

 Note

Questi metadati LaunchType sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva (Linux) o 1.0.0 o successiva (Windows).

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

DockerId

L'ID Docker per il container.

Quando utilizzi Fargate, l'ID è un valore esadecimale a 32 cifre seguito da un numero di 10 cifre.

Name

Il nome del container come specificato nella definizione di attività.

DockerName

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

Image

L'immagine per il container.

ImageID

Il digest SHA-256 per l'immagine.

Ports

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omissso.

Labels

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omissso.

DesiredStatus

Lo stato desiderato per il container da Amazon ECS.

KnownStatus

Lo stato noto per il container da Amazon ECS.

ExitCode

Il codice di uscita per il container. Se il container non si è chiuso, questo parametro viene omissso.

Limits

I limiti per le risorse specificati a livello di container, ad esempio CPU (espressa in unità CPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

CreatedAt

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omissso.

StartedAt

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omissso.

FinishedAt

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omissso.

Type

Il tipo di container. I container specificati nella definizione di attività sono di tipo NORMAL. Puoi ignorare gli altri tipi di container, utilizzati per il provisioning interno di risorse all'attività da parte dell'agente del container di Amazon ECS.

LogDriver

Il driver di log utilizzato dal container.

Note

Questi metadati `LogDriver` sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva.

LogOptions

Le opzioni del driver di log definite per il container.

Note

Questi metadati LogOptions sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva.

ContainerARN

Il nome della risorsa Amazon (ARN) del container.

Note

Questi metadati ContainerARN sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva.

Networks

Le informazioni di rete per il container, ad esempio la modalità di rete e l'indirizzo IP. Se non ci sono informazioni di rete definite, questo parametro viene omissso.

RestartCount

Il numero di volte in cui il contenitore è stato riavviato.

Note

I RestartCount metadati sono inclusi solo se è abilitata una politica di riavvio per il contenitore. Per ulteriori informazioni, consulta [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#).

ExecutionStoppedAt

Il timestamp del momento in cui le attività DesiredStatus sono passate a essere STOPPED. Questo si verifica quando un container fondamentale passa allo stato STOPPED.

Esempi di metadati delle attività di Amazon ECS v4

Gli esempi seguenti mostrano gli output di esempio da ognuno degli endpoint dei metadati dei processi.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output di un'attività eseguita come parte di un servizio (). MyService

```
{
  "DockerId": "ea32192c8553fbff06c9340478a2ff089b2bb5646fb718b4ee206641c9086d66",
  "Name": "curl",
  "DockerName": "ecs-curltest-24-curl-cca48e8dcadd97805600",
  "Image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
  "ImageID":
    "sha256:d691691e9652791a60114e67b365688d20d19940dde7c4736ea30e660d8d3553",
  "Labels": {
    "com.amazonaws.ecs.cluster": "default",
    "com.amazonaws.ecs.container-name": "curl",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/default/8f03e41243824aea923aca126495f665",
    "com.amazonaws.ecs.task-definition-family": "curltest",
    "com.amazonaws.ecs.task-definition-version": "24"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": {
    "CPU": 10,
    "Memory": 128
  },
  "CreatedAt": "2020-10-02T00:15:07.620912337Z",
  "StartedAt": "2020-10-02T00:15:08.062559351Z",
  "Type": "NORMAL",
  "LogDriver": "awslogs",
  "LogOptions": {
    "awslogs-create-group": "true",
    "awslogs-group": "/ecs/metadata",
    "awslogs-region": "us-west-2",
    "awslogs-stream": "ecs/curl/8f03e41243824aea923aca126495f665"
  },
}
```

```

    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/0206b271-
b33f-47ab-86c6-a0ba208a70a9",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.100"
        ],
        "AttachmentIndex": 0,
        "MACAddress": "0e:9e:32:c7:48:85",
        "IPv4SubnetCIDRBlock": "10.0.2.0/24",
        "PrivateDNSName": "ip-10-0-2-100.us-west-2.compute.internal",
        "SubnetGatewayIpv4Address": "10.0.2.1/24"
      }
    ]
  }
}

```

Esempio di risposta dei metadati dei processi

Quando esegui una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/task`, oltre ai metadati per ciascun container all'interno del processo, vengono restituiti i metadati relativi al processo di cui fa parte il container. Di seguito è riportato un esempio di output.

```

{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
  "Family": "curltest",
  "ServiceName": "MyService",
  "Revision": "26",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "PullStartedAt": "2020-10-02T00:43:06.202617438Z",
  "PullStoppedAt": "2020-10-02T00:43:06.31288465Z",
  "AvailabilityZone": "us-west-2d",
  "VPCID": "vpc-1234567890abcdef0",
  "LaunchType": "EC2",
  "Containers": [
    {
      "DockerId":
"598cba581fe3f939459eaba1e071d5c93bb2c49b7d1ba7db6bb19deeb70d8e38",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-curltest-26-internalecspause-e292d586b6f9dade4a00",

```

```

    "Image": "amazon/amazon-ecs-pause:0.1.0",
    "ImageID": "",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/default/158d1c8083dd49d6b527399fd6414f5c",
      "com.amazonaws.ecs.task-definition-family": "curltest",
      "com.amazonaws.ecs.task-definition-version": "26"
    },
    "DesiredStatus": "RESOURCES_PROVISIONED",
    "KnownStatus": "RESOURCES_PROVISIONED",
    "Limits": {
      "CPU": 0,
      "Memory": 0
    },
    "CreatedAt": "2020-10-02T00:43:05.602352471Z",
    "StartedAt": "2020-10-02T00:43:06.076707576Z",
    "Type": "CNI_PAUSE",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.61"
        ],
        "AttachmentIndex": 0,
        "MACAddress": "0e:10:e2:01:bd:91",
        "IPv4SubnetCIDRBlock": "10.0.2.0/24",
        "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
        "SubnetGatewayIpv4Address": "10.0.2.1/24"
      }
    ],
  },
  {
    "DockerId":
"ee08638adaaf009d78c248913f629e38299471d45fe7dc944d1039077e3424ca",
    "Name": "curl",
    "DockerName": "ecs-curltest-26-curl-a0e7dba5aca6d8cb2e00",
    "Image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
    "ImageID":
"sha256:d691691e9652791a60114e67b365688d20d19940dde7c4736ea30e660d8d3553",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "curl",

```

```

        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "26"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
        "CPU": 10,
        "Memory": 128
    },
    "CreatedAt": "2020-10-02T00:43:06.326590752Z",
    "StartedAt": "2020-10-02T00:43:06.767535449Z",
    "Type": "NORMAL",
    "LogDriver": "awslogs",
    "LogOptions": {
        "awslogs-create-group": "true",
        "awslogs-group": "/ecs/metadata",
        "awslogs-region": "us-west-2",
        "awslogs-stream": "ecs/curl/158d1c8083dd49d6b527399fd6414f5c"
    },
    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/
abb51bdd-11b4-467f-8f6c-adcfe1fe059d",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "10.0.2.61"
            ],
            "AttachmentIndex": 0,
            "MACAddress": "0e:10:e2:01:bd:91",
            "IPv4SubnetCIDRBlock": "10.0.2.0/24",
            "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
            "SubnetGatewayIpv4Address": "10.0.2.1/24"
        }
    ]
}

```

Processo di esempio con tag con una risposta ai metadati di tag

Quando si esegue una query sull'endpoint `/${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags` vengono restituiti i metadati relativi al processo, inclusi i tag del processo e dell'istanza di container. Di seguito è riportato un esempio di output.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
  "Family": "curltest",
  "ServiceName": "MyService",
  "Revision": "26",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "PullStartedAt": "2020-10-02T00:43:06.202617438Z",
  "PullStoppedAt": "2020-10-02T00:43:06.31288465Z",
  "AvailabilityZone": "us-west-2d",
  "VPCID": "vpc-1234567890abcdef0",
  "TaskTags": {
    "tag-use": "task-metadata-endpoint-test"
  },
  "ContainerInstanceTags": {
    "tag_key": "tag_value"
  },
  "LaunchType": "EC2",
  "Containers": [
    {
      "DockerId":
"598cba581fe3f939459eaba1e071d5c93bb2c49b7d1ba7db6bb19deeb70d8e38",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-curltest-26-internalecspause-e292d586b6f9dade4a00",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "26"
      },
      "DesiredStatus": "RESOURCES_PROVISIONED",
```

```

    "KnownStatus": "RESOURCES_PROVISIONED",
    "Limits": {
      "CPU": 0,
      "Memory": 0
    },
    "CreatedAt": "2020-10-02T00:43:05.602352471Z",
    "StartedAt": "2020-10-02T00:43:06.076707576Z",
    "Type": "CNI_PAUSE",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.61"
        ],
        "AttachmentIndex": 0,
        "MACAddress": "0e:10:e2:01:bd:91",
        "IPv4SubnetCIDRBlock": "10.0.2.0/24",
        "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
        "SubnetGatewayIpv4Address": "10.0.2.1/24"
      }
    ],
  },
  {
    "DockerId":
"ee08638adaaf009d78c248913f629e38299471d45fe7dc944d1039077e3424ca",
    "Name": "curl",
    "DockerName": "ecs-curltest-26-curl-a0e7dba5aca6d8cb2e00",
    "Image": "11112223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
    "ImageID":
"sha256:d691691e9652791a60114e67b365688d20d19940dde7c4736ea30e660d8d3553",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "curl",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:11112223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
      "com.amazonaws.ecs.task-definition-family": "curltest",
      "com.amazonaws.ecs.task-definition-version": "26"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
      "CPU": 10,
      "Memory": 128
    },
  },

```

```

    "CreatedAt": "2020-10-02T00:43:06.326590752Z",
    "StartedAt": "2020-10-02T00:43:06.767535449Z",
    "Type": "NORMAL",
    "LogDriver": "awslogs",
    "LogOptions": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/metadata",
      "awslogs-region": "us-west-2",
      "awslogs-stream": "ecs/curl/158d1c8083dd49d6b527399fd6414f5c"
    },
    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/
abb51bdd-11b4-467f-8f6c-adcfe1fe059d",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.61"
        ],
        "AttachmentIndex": 0,
        "MACAddress": "0e:10:e2:01:bd:91",
        "IPv4SubnetCIDRBlock": "10.0.2.0/24",
        "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
        "SubnetGatewayIpv4Address": "10.0.2.1/24"
      }
    ]
  }
}

```

Processo di esempio con tag con una risposta ai metadati di errore

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags` vengono restituiti i metadati relativi al processo, inclusi i tag del processo e dell'istanza di container. Se si verifica un errore nel recupero dei dati di tagging, l'errore viene restituito nella risposta. Di seguito è riportato un output di esempio per quando il ruolo IAM associato all'istanza di container non dispone dell'autorizzazione `ecs:ListTagsForResource` consentita.

```

{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
  "Family": "curltest",
  "ServiceName": "MyService",

```



```

"Revision": "26",
"DesiredStatus": "RUNNING",
"KnownStatus": "RUNNING",
"PullStartedAt": "2020-10-02T00:43:06.202617438Z",
"PullStoppedAt": "2020-10-02T00:43:06.31288465Z",
"AvailabilityZone": "us-west-2d",
"VPCID": "vpc-1234567890abcdef0",
"Errors": [
  {
    "ErrorField": "ContainerInstanceTags",
    "ErrorCode": "AccessDeniedException",
    "ErrorMessage": "User: arn:aws:sts::111122223333:assumed-
role/ecsInstanceRole/i-0744a608689EXAMPLE is not authorized to perform:
ecs:ListTagsForResource on resource: arn:aws:ecs:us-west-2:111122223333:container-
instance/default/2dd1b186f39845a584488d2ef155c131",
    "StatusCode": 400,
    "RequestId": "cd597ef0-272b-4643-9bd2-1de469870fa6",
    "ResourceARN": "arn:aws:ecs:us-west-2:111122223333:container-instance/
default/2dd1b186f39845a584488d2ef155c131"
  },
  {
    "ErrorField": "TaskTags",
    "ErrorCode": "AccessDeniedException",
    "ErrorMessage": "User: arn:aws:sts::111122223333:assumed-
role/ecsInstanceRole/i-0744a608689EXAMPLE is not authorized to perform:
ecs:ListTagsForResource on resource: arn:aws:ecs:us-west-2:111122223333:task/
default/9ef30e4b7aa44d0db562749cff4983f3",
    "StatusCode": 400,
    "RequestId": "862c5986-6cd2-4aa6-87cc-70be395531e1",
    "ResourceARN": "arn:aws:ecs:us-west-2:111122223333:task/
default/9ef30e4b7aa44d0db562749cff4983f3"
  }
],
"LaunchType": "EC2",
"Containers": [
  {
    "DockerId":
"598cba581fe3f939459eaba1e071d5c93bb2c49b7d1ba7db6bb19deeb70d8e38",
    "Name": "~internal~ecs~pause",
    "DockerName": "ecs-curltest-26-internalecspause-e292d586b6f9dade4a00",
    "Image": "amazon/amazon-ecs-pause:0.1.0",
    "ImageID": "",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",

```

```

        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "26"
    },
    "DesiredStatus": "RESOURCES_PROVISIONED",
    "KnownStatus": "RESOURCES_PROVISIONED",
    "Limits": {
        "CPU": 0,
        "Memory": 0
    },
    "CreatedAt": "2020-10-02T00:43:05.602352471Z",
    "StartedAt": "2020-10-02T00:43:06.076707576Z",
    "Type": "CNI_PAUSE",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "10.0.2.61"
            ],
            "AttachmentIndex": 0,
            "MACAddress": "0e:10:e2:01:bd:91",
            "IPv4SubnetCIDRBlock": "10.0.2.0/24",
            "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
            "SubnetGatewayIpv4Address": "10.0.2.1/24"
        }
    ],
},
{
    "DockerId":
"ee08638adaaf009d78c248913f629e38299471d45fe7dc944d1039077e3424ca",
    "Name": "curl",
    "DockerName": "ecs-curltest-26-curl-a0e7dba5aca6d8cb2e00",
    "Image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
    "ImageID":
"sha256:d691691e9652791a60114e67b365688d20d19940dde7c4736ea30e660d8d3553",
    "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/
default/158d1c8083dd49d6b527399fd6414f5c",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "26"
    }
}

```

```

    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
        "CPU": 10,
        "Memory": 128
    },
    "CreatedAt": "2020-10-02T00:43:06.326590752Z",
    "StartedAt": "2020-10-02T00:43:06.767535449Z",
    "Type": "NORMAL",
    "LogDriver": "awslogs",
    "LogOptions": {
        "awslogs-create-group": "true",
        "awslogs-group": "/ecs/metadata",
        "awslogs-region": "us-west-2",
        "awslogs-stream": "ecs/curl/158d1c8083dd49d6b527399fd6414f5c"
    },
    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/abb51bdd-11b4-467f-8f6c-adcfe1fe059d",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "10.0.2.61"
            ],
            "AttachmentIndex": 0,
            "MACAddress": "0e:10:e2:01:bd:91",
            "IPv4SubnetCIDRBlock": "10.0.2.0/24",
            "PrivateDNSName": "ip-10-0-2-61.us-west-2.compute.internal",
            "SubnetGatewayIpv4Address": "10.0.2.1/24"
        }
    ]
}

```

Esempio di risposta delle statistiche del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/stats` vengono restituiti i parametri di rete per il container. Per le attività di Amazon ECS che utilizzano le modalità di bridge rete `awsvpc` o ospitate su EC2 istanze Amazon che eseguono almeno una versione `1.43.0` dell'agente container, nella risposta saranno incluse statistiche aggiuntive sulla velocità di rete. Per tutti gli altri processi, la risposta includerà solo le statistiche cumulative di rete.

Di seguito è riportato un esempio di output di un'attività Amazon ECS su Amazon EC2 che utilizza la modalità bridge di rete.

```
{
  "read": "2020-10-02T00:51:13.410254284Z",
  "preread": "2020-10-02T00:51:12.406202398Z",
  "pids_stats": {
    "current": 3
  },
  "blkio_stats": {
    "io_service_bytes_recursive": [

    ],
    "io_serviced_recursive": [

    ],
    "io_queue_recursive": [

    ],
    "io_service_time_recursive": [

    ],
    "io_wait_time_recursive": [

    ],
    "io_merged_recursive": [

    ],
    "io_time_recursive": [

    ],
    "sectors_recursive": [

    ]
  },
  "num_procs": 0,
  "storage_stats": {

  },
  "cpu_stats": {
    "cpu_usage": {
      "total_usage": 360968065,
      "percpu_usage": [
```

```

        182359190,
        178608875
    ],
    "usage_in_kernelmode": 40000000,
    "usage_in_usermode": 290000000
},
"system_cpu_usage": 13939680000000,
"online_cpus": 2,
"throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
}
},
"precpu_stats": {
    "cpu_usage": {
        "total_usage": 360968065,
        "percpu_usage": [
            182359190,
            178608875
        ],
        "usage_in_kernelmode": 40000000,
        "usage_in_usermode": 290000000
    },
    "system_cpu_usage": 13937670000000,
    "online_cpus": 2,
    "throttling_data": {
        "periods": 0,
        "throttled_periods": 0,
        "throttled_time": 0
    }
},
"memory_stats": {
    "usage": 1806336,
    "max_usage": 6299648,
    "stats": {
        "active_anon": 606208,
        "active_file": 0,
        "cache": 0,
        "dirty": 0,
        "hierarchical_memory_limit": 134217728,
        "hierarchical_memsw_limit": 268435456,
        "inactive_anon": 0,
        "inactive_file": 0,

```

```

        "mapped_file": 0,
        "pgfault": 4185,
        "pgmajfault": 0,
        "pgpgin": 2926,
        "pgpgout": 2778,
        "rss": 606208,
        "rss_huge": 0,
        "total_active_anon": 606208,
        "total_active_file": 0,
        "total_cache": 0,
        "total_dirty": 0,
        "total_inactive_anon": 0,
        "total_inactive_file": 0,
        "total_mapped_file": 0,
        "total_pgfault": 4185,
        "total_pgmajfault": 0,
        "total_pgpgin": 2926,
        "total_pgpgout": 2778,
        "total_rss": 606208,
        "total_rss_huge": 0,
        "total_unevictable": 0,
        "total_writeback": 0,
        "unevictable": 0,
        "writeback": 0
    },
    "limit": 134217728
},
"name": "/ecs-curltest-26-curl-c2e5f6e0cf91b0bead01",
"id": "5fc21e5b015f899d22618f8aede80b6d70d71b2a75465ea49d9462c8f3d2d3af",
"networks": {
    "eth0": {
        "rx_bytes": 84,
        "rx_packets": 2,
        "rx_errors": 0,
        "rx_dropped": 0,
        "tx_bytes": 84,
        "tx_packets": 2,
        "tx_errors": 0,
        "tx_dropped": 0
    }
},
"network_rate_stats": {
    "rx_bytes_per_sec": 0,
    "tx_bytes_per_sec": 0
}

```

```
}
}
```

Esempio di risposta delle statistiche del processo

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/task/stats` vengono restituiti i parametri di rete relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

```
{
  "01999f2e5c6cf4df3873f28950e6278813408f281c54778efec860d0caad4854": {
    "read": "2020-10-02T00:51:32.51467703Z",
    "preread": "2020-10-02T00:51:31.50860463Z",
    "pids_stats": {
      "current": 1
    },
    "blkio_stats": {
      "io_service_bytes_recursive": [

      ],
      "io_serviced_recursive": [

      ],
      "io_queue_recursive": [

      ],
      "io_service_time_recursive": [

      ],
      "io_wait_time_recursive": [

      ],
      "io_merged_recursive": [

      ],
      "io_time_recursive": [

      ],
      "sectors_recursive": [

      ]
    },
    "num_procs": 0,
```

```

"storage_stats": {

},
"cpu_stats": {
  "cpu_usage": {
    "total_usage": 177232665,
    "percpu_usage": [
      13376224,
      163856441
    ],
    "usage_in_kernelmode": 0,
    "usage_in_usermode": 160000000
  },
  "system_cpu_usage": 13977820000000,
  "online_cpus": 2,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"precpu_stats": {
  "cpu_usage": {
    "total_usage": 177232665,
    "percpu_usage": [
      13376224,
      163856441
    ],
    "usage_in_kernelmode": 0,
    "usage_in_usermode": 160000000
  },
  "system_cpu_usage": 13975800000000,
  "online_cpus": 2,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"memory_stats": {
  "usage": 532480,
  "max_usage": 6279168,
  "stats": {
    "active_anon": 40960,

```



```

        "active_file": 0,
        "cache": 0,
        "dirty": 0,
        "hierarchical_memory_limit": 9223372036854771712,
        "hierarchical_memsw_limit": 9223372036854771712,
        "inactive_anon": 0,
        "inactive_file": 0,
        "mapped_file": 0,
        "pgfault": 2033,
        "pgmajfault": 0,
        "pgpgin": 1734,
        "pgpgout": 1724,
        "rss": 40960,
        "rss_huge": 0,
        "total_active_anon": 40960,
        "total_active_file": 0,
        "total_cache": 0,
        "total_dirty": 0,
        "total_inactive_anon": 0,
        "total_inactive_file": 0,
        "total_mapped_file": 0,
        "total_pgfault": 2033,
        "total_pgmajfault": 0,
        "total_pgpgin": 1734,
        "total_pgpgout": 1724,
        "total_rss": 40960,
        "total_rss_huge": 0,
        "total_unevictable": 0,
        "total_writeback": 0,
        "unevictable": 0,
        "writeback": 0
    },
    "limit": 4073377792
},
"name": "/ecs-curltest-26-internalecspause-a6bcc3dbadf9acfe85300",
"id": "01999f2e5c6cf4df3873f28950e6278813408f281c54778efec860d0caad4854",
"networks": {
    "eth0": {
        "rx_bytes": 84,
        "rx_packets": 2,
        "rx_errors": 0,
        "rx_dropped": 0,
        "tx_bytes": 84,
        "tx_packets": 2,

```

```

        "tx_errors": 0,
        "tx_dropped": 0
    }
},
"network_rate_stats": {
    "rx_bytes_per_sec": 0,
    "tx_bytes_per_sec": 0
}
},
"5fc21e5b015f899d22618f8aede80b6d70d71b2a75465ea49d9462c8f3d2d3af": {
    "read": "2020-10-02T00:51:32.512771349Z",
    "preread": "2020-10-02T00:51:31.510597736Z",
    "pids_stats": {
        "current": 3
    },
    "blkio_stats": {
        "io_service_bytes_recursive": [

        ],
        "io_serviced_recursive": [

        ],
        "io_queue_recursive": [

        ],
        "io_service_time_recursive": [

        ],
        "io_wait_time_recursive": [

        ],
        "io_merged_recursive": [

        ],
        "io_time_recursive": [

        ],
        "sectors_recursive": [

        ]
    },
    "num_procs": 0,
    "storage_stats": {

```

```

},
"cpu_stats": {
  "cpu_usage": {
    "total_usage": 379075681,
    "percpu_usage": [
      191355275,
      187720406
    ],
    "usage_in_kernelmode": 60000000,
    "usage_in_usermode": 310000000
  },
  "system_cpu_usage": 13977800000000,
  "online_cpus": 2,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"precpu_stats": {
  "cpu_usage": {
    "total_usage": 378825197,
    "percpu_usage": [
      191104791,
      187720406
    ],
    "usage_in_kernelmode": 60000000,
    "usage_in_usermode": 310000000
  },
  "system_cpu_usage": 13975800000000,
  "online_cpus": 2,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"memory_stats": {
  "usage": 1814528,
  "max_usage": 6299648,
  "stats": {
    "active_anon": 606208,
    "active_file": 0,
    "cache": 0,

```

```

        "dirty": 0,
        "hierarchical_memory_limit": 134217728,
        "hierarchical_memsw_limit": 268435456,
        "inactive_anon": 0,
        "inactive_file": 0,
        "mapped_file": 0,
        "pgfault": 5377,
        "pgmajfault": 0,
        "pgpgin": 3613,
        "pgpgout": 3465,
        "rss": 606208,
        "rss_huge": 0,
        "total_active_anon": 606208,
        "total_active_file": 0,
        "total_cache": 0,
        "total_dirty": 0,
        "total_inactive_anon": 0,
        "total_inactive_file": 0,
        "total_mapped_file": 0,
        "total_pgfault": 5377,
        "total_pgmajfault": 0,
        "total_pgpgin": 3613,
        "total_pgpgout": 3465,
        "total_rss": 606208,
        "total_rss_huge": 0,
        "total_unevictable": 0,
        "total_writeback": 0,
        "unevictable": 0,
        "writeback": 0
    },
    "limit": 134217728
},
"name": "/ecs-curltest-26-curl-c2e5f6e0cf91b0bead01",
"id": "5fc21e5b015f899d22618f8aede80b6d70d71b2a75465ea49d9462c8f3d2d3af",
"networks": {
    "eth0": {
        "rx_bytes": 84,
        "rx_packets": 2,
        "rx_errors": 0,
        "rx_dropped": 0,
        "tx_bytes": 84,
        "tx_packets": 2,
        "tx_errors": 0,
        "tx_dropped": 0
    }
}

```

```
    }  
  },  
  "network_rate_stats": {  
    "rx_bytes_per_sec": 0,  
    "tx_bytes_per_sec": 0  
  }  
}  
}
```

Endpoint di metadati delle attività Amazon ECS versione 3

Important

L'endpoint dei metadati delle attività versione 3 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta [the section called “Endpoint metadati delle attività versione 4”](#).

Se utilizzi attività di Amazon ECS ospitate su AWS Fargate, consulta l'[endpoint di metadati delle attività Amazon ECS versione 3 per le attività](#) su Fargate.

A partire dalla versione 1.21.0 dell'agente del container di Amazon ECS, l'agente inserisce una variabile di ambiente denominata ECS_CONTAINER_METADATA_URI in ogni container in un processo. Quando esegui una query sull'endpoint dei metadati delle attività versione 3, per le attività vengono resi disponibili diversi metadati delle attività e [statistiche Docker](#). Per le attività che utilizzano la modalità di rete bridge, sono disponibili parametri di rete durante l'esecuzione di query sugli endpoint /stats.

La funzionalità task metadata endpoint versione 3 è abilitata per impostazione predefinita per le attività che utilizzano il tipo di avvio Fargate nella versione della piattaforma v1.3.0 o successiva e per le attività che utilizzano il tipo di EC2 avvio e vengono avviate sull'infrastruttura Amazon EC2 Linux che esegue almeno la versione 1.21.0 dell'agente container Amazon ECS o sull' EC2 infrastruttura Amazon Windows che esegue almeno la versione dell'agente contenitore Amazon ECS e utilizza 1.54.0 la modalità di rete. aws-vpc Per ulteriori informazioni, consulta [Gestione delle istanze di container Amazon ECS Linux](#).

Questa caratteristica può essere supportata sulle istanze di container precedenti aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

⚠ Important

Per attività che utilizzano il tipo di avvio Fargate e versioni della piattaforma precedenti alla v1.3.0, è supportato l'endpoint dei metadati dei processi versione 2. Per ulteriori informazioni, consulta [Endpoint di metadati delle attività Amazon ECS versione 2](#).

Percorsi dell'endpoint Task Metadata versione 3

Per i container sono disponibili i seguenti endpoint dei metadati delle attività:

`${ECS_CONTAINER_METADATA_URI}`

Questo percorso restituisce il file JSON dei metadati per il container.

`${ECS_CONTAINER_METADATA_URI}/task`

Questo percorso restituisce i metadati JSON per l'attività, incluso un elenco del contenitore IDs e i nomi di tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON v3 dei metadati delle attività di Amazon ECS](#).

`${ECS_CONTAINER_METADATA_URI}/taskWithTags`

Questo percorso restituisce i metadati per il processo incluso nell'endpoint `/task` oltre ai tag dell'istanza di processo e container che possono essere recuperati utilizzando l'API `ListTagsForResource`.

`${ECS_CONTAINER_METADATA_URI}/stats`

Questo percorso restituisce un file JSON con le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

`${ECS_CONTAINER_METADATA_URI}/task/stats`

Questo percorso restituisce un file JSON delle statistiche Docker per tutti i container associati all'attività. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

Risposta JSON v3 dei metadati delle attività di Amazon ECS

La risposta in formato JSON dell'endpoint dei metadati per l'attività

(`${ECS_CONTAINER_METADATA_URI}/task`) restituisce le seguenti informazioni.

Cluster

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

TaskARN

L'Amazon Resource Name (ARN) dell'attività a cui appartiene il contenitore.

Family

La famiglia della definizione di attività Amazon ECS per il processo.

Revision

La revisione della definizione di attività di Amazon ECS per il processo.

DesiredStatus

Lo stato desiderato per il processo da Amazon ECS.

KnownStatus

Lo stato noto per il processo da Amazon ECS.

Limits

I limiti di risorse specificati a livello di attività, come CPU (espressa in vCPUs) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

PullStartedAt

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

PullStoppedAt

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

AvailabilityZone

La zona di disponibilità in cui si trova l'attività.

Note

I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

DockerId

L'ID Docker per il container.

Name

Il nome del container come specificato nella definizione di attività.

DockerName

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

Image

L'immagine per il container.

ImageID

Il digest SHA-256 per l'immagine.

Ports

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omissso.

Labels

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omissso.

DesiredStatus

Lo stato desiderato per il container da Amazon ECS.

KnownStatus

Lo stato noto per il container da Amazon ECS.

ExitCode

Il codice di uscita per il container. Se il container non si è chiuso, questo parametro viene omissso.

Limits

I limiti per le risorse specificati a livello di container, ad esempio CPU (espressa in unità CPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

CreatedAt

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omissso.

StartedAt

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omissso.

FinishedAt

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omissso.

Type

Il tipo di container. I container specificati nella definizione di attività sono di tipo NORMAL. Puoi ignorare gli altri tipi di container, utilizzati per il provisioning interno di risorse all'attività da parte dell'agente del container di Amazon ECS.

Networks

Le informazioni di rete per il container, ad esempio la modalità di rete e l'indirizzo IP. Se non ci sono informazioni di rete definite, questo parametro viene omissso.

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta [Impostare l'ora per l'istanza Linux](#) nella Amazon EC2 User Guide for Linux instances.

ReferenceTime

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio 2021-09-07T16:57:44Z.

ClockErrorBound

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

ClockSynchronizationStatus

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono SYNCHRONIZED e NOT_SYNCHRONIZED.

ExecutionStoppedAt

Il timestamp del momento in cui le attività DesiredStatus sono passate a essere STOPPED. Questo si verifica quando un container fondamentale passa allo stato STOPPED.

Esempi di metadati delle attività di Amazon ECS v3

Gli esempi seguenti mostrano gli output di esempio degli endpoint dei metadati delle attività.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output.

```
{
  "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
  "Name": "nginx-curl",
  "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
  "Image": "nrdlngr/nginx-curl",
  "ImageID":
"sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
  "Labels": {
    "com.amazonaws.ecs.cluster": "default",
    "com.amazonaws.ecs.container-name": "nginx-curl",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
    "com.amazonaws.ecs.task-definition-family": "nginx",
    "com.amazonaws.ecs.task-definition-version": "5"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
```

```

    "Limits": {
      "CPU": 512,
      "Memory": 512
    },
    "CreatedAt": "2018-02-01T20:55:10.554941919Z",
    "StartedAt": "2018-02-01T20:55:11.064236631Z",
    "Type": "NORMAL",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.106"
        ]
      }
    ]
  }
}

```

Esempio di risposta dei metadati del processo

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI}/task` vengono restituiti i metadati relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

La seguente risposta JSON è relativa a un'attività con un unico container.

```

{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",

```

```

    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
    "com.amazonaws.ecs.task-definition-family": "nginx",
    "com.amazonaws.ecs.task-definition-version": "5"
  },
  "DesiredStatus": "RESOURCES_PROVISIONED",
  "KnownStatus": "RESOURCES_PROVISIONED",
  "Limits": {
    "CPU": 0,
    "Memory": 0
  },
  "CreatedAt": "2018-02-01T20:55:08.366329616Z",
  "StartedAt": "2018-02-01T20:55:09.058354915Z",
  "Type": "CNI_PAUSE",
  "Networks": [
    {
      "NetworkMode": "awsvpc",
      "IPv4Addresses": [
        "10.0.2.106"
      ]
    }
  ]
},
{
  "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
  "Name": "nginx-curl",
  "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
  "Image": "nrdlngr/nginx-curl",
  "ImageID":
"sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
  "Labels": {
    "com.amazonaws.ecs.cluster": "default",
    "com.amazonaws.ecs.container-name": "nginx-curl",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
    "com.amazonaws.ecs.task-definition-family": "nginx",
    "com.amazonaws.ecs.task-definition-version": "5"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": {
    "CPU": 512,
    "Memory": 512
  },
},

```

```

    "CreatedAt": "2018-02-01T20:55:10.554941919Z",
    "StartedAt": "2018-02-01T20:55:11.064236631Z",
    "Type": "NORMAL",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.106"
        ]
      }
    ]
  },
  "PullStartedAt": "2018-02-01T20:55:09.372495529Z",
  "PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
  "AvailabilityZone": "us-east-2b"
}

```

Endpoint di metadati delle attività Amazon ECS versione 2

 Important

L'endpoint dei metadati delle attività versione 2 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta [the section called “Endpoint metadati delle attività versione 4”](#).

A partire dalla versione 1.17.0 dell'agente del container Amazon ECS, sono disponibili diversi metadati dei processi e [statistiche Docker](#) per i processi che utilizzano la modalità di rete awsvpc in un endpoint HTTP fornito dall'agente del container Amazon ECS.

Tutti i contenitori appartenenti alle attività avviate con la modalità di `awsipc` rete ricevono un IPv4 indirizzo locale all'interno di un intervallo di indirizzi locale del collegamento predefinito. Quando un container invia una query all'endpoint dei metadati, l'agente del container di Amazon ECS è in grado di determinare a quale processo appartiene il container in base al suo indirizzo IP univoco, per restituire quindi metadati e statistiche per tale processo.

Abilitazione dei metadati dei processi

La caratteristica dei metadati delle attività versione 2 è abilitata di default per:

- Processi che utilizzano il tipo di avvio Fargate e la versione della piattaforma v1.1.0 o versioni successive. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).
- Attività che utilizzano il tipo di EC2 avvio che utilizzano anche la modalità di aws vpc rete e vengono avviate sull'infrastruttura Amazon EC2 Linux che esegue almeno la versione 1.17.0 dell'agente container Amazon ECS o sull'infrastruttura Amazon EC2 Windows che esegue almeno 1.54.0 la versione dell'agente container Amazon ECS. Per ulteriori informazioni, consulta [Gestione delle istanze di container Amazon ECS Linux](#).

Questa caratteristica può essere supportata sulle istanze di container precedenti aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).

Percorsi per gli endpoint dei metadati dei processi

Per i container sono disponibili i seguenti endpoint API:

169.254.170.2/v2/metadata

Questo endpoint restituisce i metadati JSON per l'attività, incluso un elenco del contenitore IDs e i nomi di tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON per i metadati dei processi](#).

169.254.170.2/v2/metadata/<container-id>

Questo endpoint restituisce un file JSON con i metadati per l'ID container Docker specificato.

169.254.170.2/v2/metadata/taskWithTags

Questo percorso restituisce i metadati per il processo incluso nell'endpoint /task oltre ai tag dell'istanza di processo e container che possono essere recuperati utilizzando l'API `ListTagsForResource`.

169.254.170.2/v2/stats

Questo endpoint restituisce un file JSON delle statistiche Docker per tutti i container associati all'attività. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta la documentazione dell'[ContainerStats](#) API Docker.

169.254.170.2/v2/stats/<container-id>

Questo endpoint restituisce un file JSON con le statistiche Docker per l'ID container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

Risposta JSON per i metadati dei processi

La risposta in formato JSON dell'endpoint dei metadati per l'attività (169.254.170.2/v2/metadata) restituisce le seguenti informazioni.

Cluster

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

TaskARN

L'Amazon Resource Name (ARN) dell'attività a cui appartiene il contenitore.

Family

La famiglia della definizione di attività Amazon ECS per il processo.

Revision

La revisione della definizione di attività di Amazon ECS per il processo.

DesiredStatus

Lo stato desiderato per il processo da Amazon ECS.

KnownStatus

Lo stato noto per il processo da Amazon ECS.

Limits

I limiti di risorse specificati a livello di attività, come CPU (espressa in vCPUs) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

PullStartedAt

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

PullStoppedAt

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

AvailabilityZone

La zona di disponibilità in cui si trova l'attività.

Note

I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

DockerId

L'ID Docker per il container.

Name

Il nome del container come specificato nella definizione di attività.

DockerName

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

Image

L'immagine per il container.

ImageID

Il digest SHA-256 per l'immagine.

Ports

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omissso.

Labels

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omissso.

DesiredStatus

Lo stato desiderato per il container da Amazon ECS.

KnownStatus

Lo stato noto per il container da Amazon ECS.

ExitCode

Il codice di uscita per il container. Se il container non si è chiuso, questo parametro viene omissso.

Limits

I limiti per le risorse specificati a livello di container, ad esempio CPU (espressa in unità CPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

CreatedAt

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omissso.

StartedAt

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omissso.

FinishedAt

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omissso.

Type

Il tipo di container. I container specificati nella definizione di attività sono di tipo NORMAL. Puoi ignorare gli altri tipi di container, utilizzati per il provisioning interno di risorse all'attività da parte dell'agente del container di Amazon ECS.

Networks

Le informazioni di rete per il container, ad esempio la modalità di rete e l'indirizzo IP. Se non ci sono informazioni di rete definite, questo parametro viene omissso.

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare

la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta [Impostare l'ora per l'istanza Linux](#) nella Amazon EC2 User Guide for Linux instances.

ReferenceTime

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio 2021-09-07T16:57:44Z.

ClockErrorBound

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

ClockSynchronizationStatus

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono SYNCHRONIZED e NOT_SYNCHRONIZED.

ExecutionStoppedAt

Il timestamp del momento in cui le attività DesiredStatus sono passate a essere STOPPED. Questo si verifica quando un container fondamentale passa allo stato STOPPED.

Esempio di risposta dei metadati del processo

La seguente risposta JSON è relativa a un'attività con un unico container.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",

```

```

    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
      "com.amazonaws.ecs.task-definition-family": "nginx",
      "com.amazonaws.ecs.task-definition-version": "5"
    },
    "DesiredStatus": "RESOURCES_PROVISIONED",
    "KnownStatus": "RESOURCES_PROVISIONED",
    "Limits": {
      "CPU": 0,
      "Memory": 0
    },
    "CreatedAt": "2018-02-01T20:55:08.366329616Z",
    "StartedAt": "2018-02-01T20:55:09.058354915Z",
    "Type": "CNI_PAUSE",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.106"
        ]
      }
    ]
  },
  {
    "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
    "Name": "nginx-curl",
    "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
    "Image": "nrdlngr/nginx-curl",
    "ImageID":
"sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "nginx-curl",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
      "com.amazonaws.ecs.task-definition-family": "nginx",
      "com.amazonaws.ecs.task-definition-version": "5"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {

```

```
    "CPU": 512,
    "Memory": 512
  },
  "CreatedAt": "2018-02-01T20:55:10.554941919Z",
  "StartedAt": "2018-02-01T20:55:11.064236631Z",
  "Type": "NORMAL",
  "Networks": [
    {
      "NetworkMode": "awsvpc",
      "IPv4Addresses": [
        "10.0.2.106"
      ]
    }
  ]
},
"PullStartedAt": "2018-02-01T20:55:09.372495529Z",
"PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
"AvailabilityZone": "us-east-2b"
}
```

Metadati delle attività di Amazon ECS disponibili per le attività su Fargate

Amazon ECS su Fargate fornisce un metodo per recuperare vari metadati, parametri di rete e [statistiche Docker](#) relativi ai container e i processi in essi contenuti. Questo metodo è noto come endpoint dei metadati dei processi. Le seguenti versioni degli endpoint dei metadati dei processi sono disponibili per i processi di Amazon ECS su Fargate:

- Endpoint dei metadati dei processi versione 4: disponibile per i processi che utilizzano la versione della piattaforma 1.4.0 o successiva.
- Endpoint dei metadati dei processi versione 3: disponibile per i processi che utilizzano la versione della piattaforma 1.1.0 o successiva.

Tutti i contenitori appartenenti alle attività avviate con la modalità di `awsvpc` rete ricevono un IPv4 indirizzo locale all'interno di un intervallo di indirizzi locale del collegamento predefinito. Quando un container invia una query all'endpoint dei metadati, l'agente del container è in grado di determinare a quale attività appartiene il container in base al suo indirizzo IP univoco, per restituire quindi metadati e statistiche per tale attività.

Argomenti

- [Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate](#)
- [Endpoint di metadati delle attività Amazon ECS versione 3 per attività su Fargate](#)

Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate

Important

Se utilizzi attività Amazon ECS ospitate su EC2 istanze Amazon, consulta l'endpoint di metadati delle [attività Amazon ECS](#).

A partire dalla piattaforma Fargate versione 1.4.0, una variabile di ambiente denominata `ECS_CONTAINER_METADATA_URI_V4` viene inserita in ogni container di un processo. Quando esegui una query sull'endpoint dei metadati dei processi versione 4, per i processi vengono resi disponibili diversi metadati delle attività e le [statistiche Docker](#).

L'endpoint dei metadati dei processi versione 4 funziona come l'endpoint della versione 3, ma fornisce metadati di rete aggiuntivi per i container e i processi. Ulteriori parametri di rete sono disponibili anche quando si eseguono query sugli endpoint `/stats`.

L'endpoint dei metadati delle attività è attivo per impostazione predefinita per tutte le attività di Amazon ECS eseguite su AWS Fargate quella versione 1.4.0 della piattaforma di utilizzo o successiva.

Note

Per evitare la necessità di creare nuove versioni degli endpoint dei metadati delle attività in futuro, ulteriori metadati potrebbero essere aggiunti all'output della versione 4. Non rimuoveremo i metadati esistenti né modificheremo i nomi dei campi dei metadati.

Percorsi per gli endpoint metadati delle attività Fargate versione 4

Per i container sono disponibili i seguenti endpoint dei metadati delle attività:

`${ECS_CONTAINER_METADATA_URI_V4}`

Questo percorso restituisce i metadati per il container.

`${ECS_CONTAINER_METADATA_URI_V4}/task`

Questo percorso restituisce i metadati per l'attività, incluso un elenco del contenitore IDs e i nomi di tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON dei metadati delle attività di Amazon ECS v4 per le attività su Fargate](#).

`${ECS_CONTAINER_METADATA_URI_V4}/stats`

Questo percorso le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

 Note

Le attività di Amazon ECS AWS Fargate richiedono che il contenitore funzioni per circa 1 secondo prima di restituire le statistiche del contenitore.

`${ECS_CONTAINER_METADATA_URI_V4}/task/stats`

Questo percorso restituisce le statistiche Docker per tutti i container associati al processo. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

 Note

Le attività di Amazon ECS AWS Fargate richiedono che il contenitore funzioni per circa 1 secondo prima di restituire le statistiche del contenitore.

Risposta JSON dei metadati delle attività di Amazon ECS v4 per le attività su Fargate

I seguenti metadati vengono restituiti nella risposta in formato JSON dell'endpoint dei metadati dei processi (`${ECS_CONTAINER_METADATA_URI_V4}/task`).

Cluster

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

VPCID

L'ID VPC dell'istanza del EC2 contenitore Amazon. Questo campo viene visualizzato solo per le EC2 istanze Amazon.

Note

Questi metadati VPCID sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione 1.63.1 o successiva.

TaskARN

L'Amazon Resource Name (ARN) dell'attività a cui appartiene il contenitore.

Family

La famiglia della definizione di attività Amazon ECS per il processo.

Revision

La revisione della definizione di attività di Amazon ECS per il processo.

DesiredStatus

Lo stato desiderato per il processo da Amazon ECS.

KnownStatus

Lo stato noto per il processo da Amazon ECS.

Limits

I limiti di risorse specificati a livello di attività, ad esempio CPU (espressi in vCPUs) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

PullStartedAt

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

PullStoppedAt

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

AvailabilityZone

La zona di disponibilità in cui si trova l'attività.

Note

I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 (Windows).

LaunchType

Il tipo di avvio utilizzato dall'attività. Quando si utilizzano provider di capacità del cluster, ciò indica se l'attività utilizza Fargate o EC2 l'infrastruttura.

Note

Questi metadati LaunchType sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva (Linux) o 1.0.0 o successiva (Windows).

EphemeralStorageMetrics

La dimensione riservata e l'uso corrente dello spazio di archiviazione temporanea di questa attività.

Note

Fargate riserva spazio su disco destinato unicamente a questo motore di calcolo. Non ti viene addebitato alcun costo. Sebbene non sia mostrato in queste metriche, puoi visualizzare questo spazio di archiviazione aggiuntivo in altri strumenti, come `df`.

Utilized

L'utilizzo dello spazio di archiviazione temporanea (in MiB) di questa attività.

Reserved

L'utilizzo dello spazio di archiviazione riservato (in MiB) di questa attività. La dimensione dello spazio di archiviazione temporanea non può essere modificata in un'attività in esecuzione. Per modificare la quantità di spazio di archiviazione temporanea, è possibile specificare

l'oggetto `ephemeralStorage` nella definizione di attività. Lo `ephemeralStorage` è specificato in GiB, non in MiB. Lo `ephemeralStorage` e i parametri `EphemeralStorageMetrics` sono disponibili solo per i processi che utilizzano la piattaforma Fargate Linux versione 1.4.0 o successiva.

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

Quando utilizzi Fargate, l'ID è un valore esadecimale a 32 cifre seguito da un numero di 10 cifre.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omissso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omissso.

`DesiredStatus`

Lo stato desiderato per il container da Amazon ECS.

KnownStatus

Lo stato noto per il container da Amazon ECS.

ExitCode

Il codice di uscita per il container. Se il container non si è chiuso, questo parametro viene omissso.

Limits

I limiti per le risorse specificati a livelli di container, ad esempio CPU (espressa in unità CPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

CreatedAt

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omissso.

StartedAt

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omissso.

FinishedAt

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omissso.

Type

Il tipo di container. I container specificati nella definizione di attività sono di tipo NORMAL. Puoi ignorare gli altri tipi di container, utilizzati per il provisioning interno di risorse all'attività da parte dell'agente del container di Amazon ECS.

LogDriver

Il driver di log utilizzato dal container.

Note

Questi metadati `LogDriver` sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione `1.45.0` o successiva.

LogOptions

Le opzioni del driver di log definite per il container.

Note

Questi metadati `LogOptions` sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva.

ContainerARN

Il nome della risorsa Amazon (ARN) del container.

Note

Questi metadati `ContainerARN` sono inclusi solo quando si utilizza la versione dell'agente del container Linux di Amazon ECS versione 1.45.0 o successiva.

Networks

Le informazioni di rete per il container, ad esempio la modalità di rete e l'indirizzo IP. Se non ci sono informazioni di rete definite, questo parametro viene omissso.

Snapshotter

Il snapshotter che è stato usato da containerd per scaricare l'immagine del contenitore. I valori validi sono `overlayfs`, che è l'impostazione predefinita, e `soci`, ovvero il valore utilizzato durante il caricamento lento con un indice SOCI. Questo parametro è disponibile solo per le attività in esecuzione sulla versione della piattaforma 1.4.0 di Linux.

RestartCount

Il numero di volte in cui il contenitore è stato riavviato.

Note

I `RestartCount` metadati sono inclusi solo se è abilitata una politica di riavvio per il contenitore. Per ulteriori informazioni, consulta [Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori](#).

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta [Impostare l'ora per l'istanza Linux](#) nella Amazon EC2 User Guide for Linux instances.

ReferenceTime

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio 2021-09-07T16:57:44Z.

ClockErrorBound

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

ClockSynchronizationStatus

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono SYNCHRONIZED e NOT_SYNCHRONIZED.

ExecutionStoppedAt

Il timestamp del momento in cui le attività DesiredStatus sono passate a essere STOPPED. Questo si verifica quando un container fondamentale passa allo stato STOPPED.

Esempi di metadati delle attività di Amazon ECS v4 per attività su Fargate

Gli esempi seguenti mostrano gli output di esempio degli endpoint dei metadati dei processi per i processi di Amazon ECS in esecuzione su AWS Fargate.

Dal container è possibile utilizzare il comando curl seguito dall'endpoint dei metadati dell'attività per eseguire una query sull'endpoint, ad esempio `curl ${ECS_CONTAINER_METADATA_URI_V4}/task`.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output.

```
{
```

```

    "DockerId": "cd189a933e5849daa93386466019ab50-2495160603",
    "Name": "curl",
    "DockerName": "curl",
    "Image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
    "ImageID":
"sha256:25f3695bedfb454a50f12d127839a68ad3caf91e451c1da073db34c542c4d2cb",
    "Labels": {
        "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-west-2:111122223333:cluster/
default",
        "com.amazonaws.ecs.container-name": "curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/default/
cd189a933e5849daa93386466019ab50",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "2"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
        "CPU": 10,
        "Memory": 128
    },
    "CreatedAt": "2020-10-08T20:09:11.44527186Z",
    "StartedAt": "2020-10-08T20:09:11.44527186Z",
    "Type": "NORMAL",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "192.0.2.3"
            ],
            "AttachmentIndex": 0,
            "MACAddress": "0a:de:f6:10:51:e5",
            "IPv4SubnetCIDRBlock": "192.0.2.0/24",
            "DomainNameServers": [
                "192.0.2.2"
            ],
            "DomainNameSearchList": [
                "us-west-2.compute.internal"
            ],
            "PrivateDNSName": "ip-10-0-0-222.us-west-2.compute.internal",
            "SubnetGatewayIpv4Address": "192.0.2.0/24"
        }
    ],

```

```

"ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/05966557-
f16c-49cb-9352-24b3a0dcd0e1",
"LogOptions": {
  "awslogs-create-group": "true",
  "awslogs-group": "/ecs/containerlogs",
  "awslogs-region": "us-west-2",
  "awslogs-stream": "ecs/curl/cd189a933e5849daa93386466019ab50"
},
"LogDriver": "awslogs",
"Snapshotter": "overlayfs"
}

```

Esempi di metadati delle attività di Amazon ECS v4 per attività su Fargate

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/task` vengono restituiti i metadati relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

```

{
  "Cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/MyEmptyCluster",
  "TaskARN": "arn:aws:ecs:us-east-1:123456789012:task/MyEmptyCluster/
bfa2636268144d039771334145e490c5",
  "Family": "sample-fargate",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": {
    "CPU": 0.25,
    "Memory": 512
  },
  "PullStartedAt": "2023-07-21T15:45:33.532811081Z",
  "PullStoppedAt": "2023-07-21T15:45:38.541068435Z",
  "AvailabilityZone": "us-east-1d",
  "Containers": [
    {
      "DockerId": "bfa2636268144d039771334145e490c5-1117626119",
      "Name": "curl-image",
      "DockerName": "curl-image",
      "Image": "curlimages/curl",
      "ImageID":
"sha256:daf3f46a2639c1613b25e85c9ee4193af8a1d538f92483d67f9a3d7f21721827",
      "Labels": {

```

```

    "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/
MyEmptyCluster",
    "com.amazonaws.ecs.container-name": "curl-image",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-1:123456789012:task/
MyEmptyCluster/bfa2636268144d039771334145e490c5",
    "com.amazonaws.ecs.task-definition-family": "sample-fargate",
    "com.amazonaws.ecs.task-definition-version": "5"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": { "CPU": 128 },
  "CreatedAt": "2023-07-21T15:45:44.91368314Z",
  "StartedAt": "2023-07-21T15:45:44.91368314Z",
  "Type": "NORMAL",
  "Networks": [
    {
      "NetworkMode": "awsvpc",
      "IPv4Addresses": ["172.31.42.189"],
      "AttachmentIndex": 0,
      "MACAddress": "0e:98:9f:33:76:d3",
      "IPv4SubnetCIDRBlock": "172.31.32.0/20",
      "DomainNameServers": ["172.31.0.2"],
      "DomainNameSearchList": ["ec2.internal"],
      "PrivateDNSName": "ip-172-31-42-189.ec2.internal",
      "SubnetGatewayIpv4Address": "172.31.32.1/20"
    }
  ],
  "ContainerARN": "arn:aws:ecs:us-east-1:123456789012:container/MyEmptyCluster/
bfa2636268144d039771334145e490c5/da6cccf7-1178-400c-afdf-7536173ee209",
  "Snapshotter": "overlayfs"
},
{
  "DockerId": "bfa2636268144d039771334145e490c5-3681984407",
  "Name": "fargate-app",
  "DockerName": "fargate-app",
  "Image": "public.ecr.aws/docker/library/httpd:latest",
  "ImageID":
    "sha256:8059bdd0058510c03ae4c808de8c4fd2c1f3c1b6d9ea75487f1e5caa5ececa02",
  "Labels": {
    "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/
MyEmptyCluster",
    "com.amazonaws.ecs.container-name": "fargate-app",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-1:123456789012:task/
MyEmptyCluster/bfa2636268144d039771334145e490c5",

```

```

    "com.amazonaws.ecs.task-definition-family": "sample-fargate",
    "com.amazonaws.ecs.task-definition-version": "5"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": { "CPU": 2 },
  "CreatedAt": "2023-07-21T15:45:44.954460255Z",
  "StartedAt": "2023-07-21T15:45:44.954460255Z",
  "Type": "NORMAL",
  "Networks": [
    {
      "NetworkMode": "awsvpc",
      "IPv4Addresses": ["172.31.42.189"],
      "AttachmentIndex": 0,
      "MACAddress": "0e:98:9f:33:76:d3",
      "IPv4SubnetCIDRBlock": "172.31.32.0/20",
      "DomainNameServers": ["172.31.0.2"],
      "DomainNameSearchList": ["ec2.internal"],
      "PrivateDNSName": "ip-172-31-42-189.ec2.internal",
      "SubnetGatewayIPv4Address": "172.31.32.1/20"
    }
  ],
  "ContainerARN": "arn:aws:ecs:us-east-1:123456789012:container/MyEmptyCluster/
bfa2636268144d039771334145e490c5/f65b461d-aa09-4acb-a579-9785c0530cbc",
  "Snapshotter": "overlayfs"
}
],
"LaunchType": "FARGATE",
"ClockDrift": {
  "ClockErrorBound": 0.446931,
  "ReferenceTimestamp": "2023-07-21T16:09:17Z",
  "ClockSynchronizationStatus": "SYNCHRONIZED"
},
"EphemeralStorageMetrics": {
  "Utilized": 261,
  "Reserved": 20496
}
}

```


Esempio di risposta delle statistiche del processo

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/task/stats` vengono restituiti i parametri di rete relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

```
{
  "3d1f891cded94dc795608466cce8ddcf-464223573": {
    "read": "2020-10-08T21:24:44.938937019Z",
    "preread": "2020-10-08T21:24:34.938633969Z",
    "pids_stats": {},
    "blkio_stats": {
      "io_service_bytes_recursive": [
        {
          "major": 202,
          "minor": 26368,
          "op": "Read",
          "value": 638976
        },
        {
          "major": 202,
          "minor": 26368,
          "op": "Write",
          "value": 0
        },
        {
          "major": 202,
          "minor": 26368,
          "op": "Sync",
          "value": 638976
        },
        {
          "major": 202,
          "minor": 26368,
          "op": "Async",
          "value": 0
        },
        {
          "major": 202,
          "minor": 26368,
          "op": "Total",
          "value": 638976
        }
      ]
    }
  }
}
```

```

],
"io_serviced_recursive": [
  {
    "major": 202,
    "minor": 26368,
    "op": "Read",
    "value": 12
  },
  {
    "major": 202,
    "minor": 26368,
    "op": "Write",
    "value": 0
  },
  {
    "major": 202,
    "minor": 26368,
    "op": "Sync",
    "value": 12
  },
  {
    "major": 202,
    "minor": 26368,
    "op": "Async",
    "value": 0
  },
  {
    "major": 202,
    "minor": 26368,
    "op": "Total",
    "value": 12
  }
],
"io_queue_recursive": [],
"io_service_time_recursive": [],
"io_wait_time_recursive": [],
"io_merged_recursive": [],
"io_time_recursive": [],
"sectors_recursive": []
},
"num_procs": 0,
"storage_stats": {},
"cpu_stats": {
  "cpu_usage": {

```

```

    "total_usage": 1137691504,
    "percpu_usage": [
      696479228,
      441212276,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ],
    "usage_in_kernelmode": 80000000,
    "usage_in_usermode": 810000000
  },
  "system_cpu_usage": 9393210000000,
  "online_cpus": 2,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"precpu_stats": {
  "cpu_usage": {
    "total_usage": 1136624601,
    "percpu_usage": [
      695639662,
      440984939,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ]
  }
}

```

```

    0,
    0,
    0,
    0
  ],
  "usage_in_kernelmode": 80000000,
  "usage_in_usermode": 810000000
},
"system_cpu_usage": 9373330000000,
"online_cpus": 2,
"throttling_data": {
  "periods": 0,
  "throttled_periods": 0,
  "throttled_time": 0
}
},
"memory_stats": {
  "usage": 6504448,
  "max_usage": 8458240,
  "stats": {
    "active_anon": 1675264,
    "active_file": 557056,
    "cache": 651264,
    "dirty": 0,
    "hierarchical_memory_limit": 536870912,
    "hierarchical_memsw_limit": 9223372036854772000,
    "inactive_anon": 0,
    "inactive_file": 3088384,
    "mapped_file": 430080,
    "pgfault": 11034,
    "pgmajfault": 5,
    "pgpgin": 8436,
    "pgpgout": 7137,
    "rss": 4669440,
    "rss_huge": 0,
    "total_active_anon": 1675264,
    "total_active_file": 557056,
    "total_cache": 651264,
    "total_dirty": 0,
    "total_inactive_anon": 0,
    "total_inactive_file": 3088384,
    "total_mapped_file": 430080,
    "total_pgfault": 11034,
    "total_pgmajfault": 5,

```

```

    "total_pgpgin": 8436,
    "total_pgpgout": 7137,
    "total_rss": 4669440,
    "total_rss_huge": 0,
    "total_unevictable": 0,
    "total_writeback": 0,
    "unevictable": 0,
    "writeback": 0
  },
  "limit": 9223372036854772000
},
"name": "curltest",
"id": "3d1f891cded94dc795608466cce8ddcf-464223573",
"networks": {
  "eth1": {
    "rx_bytes": 2398415937,
    "rx_packets": 1898631,
    "rx_errors": 0,
    "rx_dropped": 0,
    "tx_bytes": 1259037719,
    "tx_packets": 428002,
    "tx_errors": 0,
    "tx_dropped": 0
  }
},
"network_rate_stats": {
  "rx_bytes_per_sec": 43.298687872232854,
  "tx_bytes_per_sec": 215.39347269466413
}
}
}

```

Endpoint di metadati delle attività Amazon ECS versione 3 per attività su Fargate

Important

L'endpoint dei metadati delle attività versione 3 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta [the section called “Endpoint metadati delle attività versione 4 per le attività su Fargate”](#).

A partire dalla piattaforma Fargate versione 1.1.0, una variabile di ambiente denominata `ECS_CONTAINER_METADATA_URI` viene inserita in ogni container di un processo. Quando esegui una query sull'endpoint dei metadati delle attività versione 3, per le attività vengono resi disponibili diversi metadati delle attività e [statistiche Docker](#).

L'endpoint dei metadati dei processi è abilitato di default per tutti i processi di Amazon ECS ospitati su Fargate che utilizzano la versione della piattaforma 1.1.0 o versione successiva. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).

Percorsi per gli endpoint metadati delle attività per le attività su Fargate

Per i container sono disponibili i seguenti endpoint API:

`${ECS_CONTAINER_METADATA_URI}`

Questo percorso restituisce il file JSON dei metadati per il container.

`${ECS_CONTAINER_METADATA_URI}/task`

Questo percorso restituisce i metadati JSON per l'attività, incluso un elenco del contenitore IDs e i nomi di tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON dei metadati delle attività di Amazon ECS v3 per le attività su Fargate](#).

`${ECS_CONTAINER_METADATA_URI}/stats`

Questo percorso restituisce un file JSON con le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

`${ECS_CONTAINER_METADATA_URI}/task/stats`

Questo percorso restituisce un file JSON delle statistiche Docker per tutti i container associati all'attività. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta [ContainerStats](#) la documentazione dell'API Docker.

Risposta JSON dei metadati delle attività di Amazon ECS v3 per le attività su Fargate

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`${ECS_CONTAINER_METADATA_URI}/task`) restituisce le seguenti informazioni.

Cluster

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

TaskARN

L'Amazon Resource Name (ARN) dell'attività a cui appartiene il contenitore.

Family

La famiglia della definizione di attività Amazon ECS per il processo.

Revision

La revisione della definizione di attività di Amazon ECS per il processo.

DesiredStatus

Lo stato desiderato per il processo da Amazon ECS.

KnownStatus

Lo stato noto per il processo da Amazon ECS.

Limits

I limiti di risorse specificati a livello di attività, come CPU (espressa in vCPUs) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

PullStartedAt

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

PullStoppedAt

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

AvailabilityZone

La zona di disponibilità in cui si trova l'attività.

Note

I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

DockerId

L'ID Docker per il container.

Name

Il nome del container come specificato nella definizione di attività.

DockerName

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

Image

L'immagine per il container.

ImageID

Il digest SHA-256 per l'immagine.

Ports

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omissso.

Labels

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omissso.

DesiredStatus

Lo stato desiderato per il container da Amazon ECS.

KnownStatus

Lo stato noto per il container da Amazon ECS.

ExitCode

Il codice di uscita per il container. Se il container non si è chiuso, questo parametro viene omissso.

Limits

I limiti per le risorse specificati a livello di container, ad esempio CPU (espressa in unità CPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omissso.

CreatedAt

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omissso.

StartedAt

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omissso.

FinishedAt

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omissso.

Type

Il tipo di container. I container specificati nella definizione di attività sono di tipo NORMAL. Puoi ignorare gli altri tipi di container, utilizzati per il provisioning interno di risorse all'attività da parte dell'agente del container di Amazon ECS.

Networks

Le informazioni di rete per il container, ad esempio la modalità di rete e l'indirizzo IP. Se non ci sono informazioni di rete definite, questo parametro viene omissso.

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta [Impostare l'ora per l'istanza Linux](#) nella Amazon EC2 User Guide for Linux instances.

ReferenceTime

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio 2021-09-07T16:57:44Z.

ClockErrorBound

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

ClockSynchronizationStatus

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono SYNCHRONIZED e NOT_SYNCHRONIZED.

ExecutionStoppedAt

Il timestamp del momento in cui le attività DesiredStatus sono passate a essere STOPPED. Questo si verifica quando un container fondamentale passa allo stato STOPPED.

Esempi di metadati delle attività di Amazon ECS v3 per attività su Fargate

La seguente risposta JSON è relativa a un'attività con un unico container.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      }
    },
  ],
}
```

```

    "DesiredStatus": "RESOURCES_PROVISIONED",
    "KnownStatus": "RESOURCES_PROVISIONED",
    "Limits": {
      "CPU": 0,
      "Memory": 0
    },
    "CreatedAt": "2018-02-01T20:55:08.366329616Z",
    "StartedAt": "2018-02-01T20:55:09.058354915Z",
    "Type": "CNI_PAUSE",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
          "10.0.2.106"
        ]
      }
    ]
  },
  {
    "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
    "Name": "nginx-curl",
    "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
    "Image": "nrdlngr/nginx-curl",
    "ImageID":
"sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
    "Labels": {
      "com.amazonaws.ecs.cluster": "default",
      "com.amazonaws.ecs.container-name": "nginx-curl",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
      "com.amazonaws.ecs.task-definition-family": "nginx",
      "com.amazonaws.ecs.task-definition-version": "5"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
      "CPU": 512,
      "Memory": 512
    },
    "CreatedAt": "2018-02-01T20:55:10.554941919Z",
    "StartedAt": "2018-02-01T20:55:11.064236631Z",
    "Type": "NORMAL",
    "Networks": [
      {

```

```
        "NetworkMode": "awsvpc",
        "IPv4Addresses": [
            "10.0.2.106"
        ]
    }
],
"PullStartedAt": "2018-02-01T20:55:09.372495529Z",
"PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
"AvailabilityZone": "us-east-2b"
}
```

Introspezione dei container Amazon ECS

L'agente del container di Amazon ECS fornisce un'operazione API per raccogliere i dettagli sull'istanza di container su cui è in esecuzione l'agente e sui processi associati in esecuzione su tale istanza. Puoi utilizzare il comando curl dall'interno dell'istanza di container per interrogare l'agente del container di Amazon ECS (porta 51678) e restituire metadati dell'istanza di container o informazioni sulle attività.

Important

L'istanza di container deve disporre di un ruolo IAM che consenta l'accesso ad Amazon ECS per recuperare i metadati. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

Per visualizzare i metadati dell'istanza di container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente. I metadati includono l'ID dell'istanza di container, il cluster Amazon ECS in cui è registrata l'istanza di container e le informazioni sulla versione dell'agente del container di Amazon ECS.

```
curl -s http://localhost:51678/v1/metadata | python3 -mjson.tool
```

Output:

```
{
  "Cluster": "cluster_name",
```

```

    "ContainerInstanceArn": "arn:aws:ecs:region:aws_account_id:container-
instance/cluster_name/container_instance_id",
    "Version": "Amazon ECS Agent - v1.30.0 (02ff320c)"
}

```

Per visualizzare le informazioni su tutte le attività in esecuzione in un'istanza di container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente:

```
curl http://localhost:51678/v1/tasks
```

Output:

```

{
  "Tasks": [
    {
      "Arn": "arn:aws:ecs:us-west-2:012345678910:task/default/example5-58ff-46c9-
ae05-543f8example",
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Family": "hello_world",
      "Version": "8",
      "Containers": [
        {
          "DockerId":
"9581a69a761a557fbfce1d0f6745e4af5b9dbfb86b6b2c5c4df156f1a5932ff1",
          "DockerName": "ecs-hello_world-8-mysql-fcae8ac8f9f1d89d8301",
          "Name": "mysql",
          "CreatedAt": "2023-10-08T20:09:11.44527186Z",
          "StartedAt": "2023-10-08T20:09:11.44527186Z",
          "ImageID":
"sha256:2ae34abc2ed0a22e280d17e13f9c01aaf725688b09b7a1525d1a2750e2c0d1de"
        },
        {
          "DockerId":
"bf25c5c5b2d4dba68846c7236e75b6915e1e778d31611e3c6a06831e39814a15",
          "DockerName": "ecs-hello_world-8-wordpress-e8bfddf9b488dff36c00",
          "Name": "wordpress"
        }
      ]
    }
  ]
}

```

```
}
```

Puoi visualizzare le informazioni per una determinata attività in esecuzione in un'istanza di container. Per specificare una determinata attività o uno specifico container, aggiungi uno dei seguenti elementi alla richiesta:

- L'ARN dell'attività (?taskarn=*task_arn*)
- L'ID Docker per un container (?dockerid=*docker_id*)

Per ottenere informazioni sulle attività con un ID Docker del container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente.

Note

Gli agenti container Amazon ECS precedenti alla versione 1.14.2 richiedono un contenitore Docker completo IDs per l'API di introspezione, non la versione breve mostrata con. `docker ps` Puoi ottenere l'ID Docker completo per un container eseguendo il comando `docker ps --no-trunc` nell'istanza di container.

```
curl http://localhost:51678/v1/tasks?dockerid=79c796ed2a7f
```

Output:

```
{
  "Arn": "arn:aws:ecs:us-west-2:012345678910:task/default/e01d58a8-151b-40e8-
bc01-22647b9ecfec",
  "Containers": [
    {
      "DockerId":
"79c796ed2a7f864f485c76f83f3165488097279d296a7c05bd5201a1c69b2920",
      "DockerName": "ecs-nginx-efs-2-nginx-9ac0808dd0afa495f001",
      "Name": "nginx",
      "CreatedAt": "2023-10-08T20:09:11.44527186Z",
      "StartedAt": "2023-10-08T20:09:11.44527186Z",
      "ImageID":
"sha256:2ae34abc2ed0a22e280d17e13f9c01aaf725688b09b7a1525d1a2750e2c0d1de"
    }
  ]
}
```

```
],  
  "DesiredStatus": "RUNNING",  
  "Family": "nginx-efs",  
  "KnownStatus": "RUNNING",  
  "Version": "2"  
}
```

Identifica i comportamenti non autorizzati utilizzando Runtime Monitoring

Amazon GuardDuty è un servizio di rilevamento delle minacce che aiuta a proteggere account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente. Utilizzando modelli di machine learning (ML) e funzionalità di rilevamento di anomalie e minacce, monitora GuardDuty continuamente diverse fonti di log e attività di runtime per identificare e dare priorità ai potenziali rischi per la sicurezza e alle attività dannose nel tuo ambiente.

Runtime Monitoring in GuardDuty protegge i carichi di lavoro in esecuzione su Fargate EC2 e sulle istanze di container AWS monitorando continuamente i log e l'attività di rete per identificare comportamenti dannosi o non autorizzati. Runtime Monitoring utilizza un agente di GuardDuty sicurezza leggero e completamente gestito che analizza il comportamento sull'host, come l'accesso ai file, l'esecuzione dei processi e le connessioni di rete. Ciò riguarda questioni quali l'aumento dei privilegi, l'uso di credenziali esposte o la comunicazione con indirizzi IP e domini dannosi e la presenza di malware sulle istanze Amazon EC2 e sui carichi di lavoro dei container. [Per ulteriori informazioni, consulta GuardDuty Runtime Monitoring nella Guida per l'utente. GuardDuty](#)

L'amministratore della sicurezza abilita il Runtime Monitoring per uno o più account in AWS Organizations for GuardDuty. Inoltre, selezionano se distribuire GuardDuty automaticamente il GuardDuty Security Agent quando si utilizza Fargate. Tutti i tuoi cluster sono protetti automaticamente e GuardDuty gestiscono il security agent per tuo conto.

Puoi anche configurare manualmente il GuardDuty Security Agent nei seguenti casi:

- Si utilizzano istanze di EC2 container
- È necessario un controllo granulare per abilitare il monitoraggio del runtime a livello di cluster

Per utilizzare Runtime Monitoring, è necessario configurare i cluster protetti e installare e gestire il GuardDuty security agent sulle istanze del EC2 contenitore.

Come funziona il monitoraggio del runtime con Amazon ECS

Runtime Monitoring utilizza un agente GuardDuty di sicurezza leggero che monitora l'attività del carico di lavoro di Amazon ECS in base al modo in cui le applicazioni richiedono, ottengono l'accesso e consumano le risorse di sistema sottostanti.

Per le attività di Fargate, il GuardDuty security agent funge da contenitore secondario per ogni attività.

Per le istanze di EC2 container, il GuardDuty security agent viene eseguito come processo sull'istanza.

Il GuardDuty security agent raccoglie i dati dalle seguenti risorse e quindi li invia GuardDuty all'elaborazione. È possibile visualizzare i risultati nella GuardDuty console. Puoi anche inviarli ad altri, ad Servizi AWS esempio AWS Security Hub, o a un fornitore di sicurezza di terze parti per l'aggregazione e la correzione. Per informazioni su come visualizzare e gestire i risultati, consulta [Managing Amazon GuardDuty findings](#) nella Amazon GuardDuty User Guide.

- Risposte dalle seguenti chiamate API Amazon ECS:

- [DescribeClusters](#)

I parametri di risposta includono il tag Runtime Monitoring (quando il tag è impostato) quando si utilizza l'--include TAGSopzione.

- [DescribeTasks](#)

Per il tipo di lancio Fargate, i parametri di risposta includono il contenitore GuardDuty sidecar.

- [ListAccountSettings](#)

I parametri di risposta includono l'impostazione dell'account Runtime Monitoring, impostata dall'amministratore della sicurezza.

- I dati di introspezione dell'agente contenitore. Per ulteriori informazioni, consulta [Introspezione dei container Amazon ECS](#).
- L'endpoint dei metadati dell'attività per il tipo di avvio:
 - [Endpoint di metadati delle attività Amazon ECS versione 4](#)
 - [Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate](#)

Considerazioni

Considerate quanto segue quando utilizzate Runtime Monitoring:

- Il Runtime Monitoring ha un costo associato. Per ulteriori informazioni, consulta la pagina [GuardDuty dei prezzi di Amazon](#).
- Il monitoraggio del runtime non è supportato su Amazon ECS Anywhere.
- Il monitoraggio del runtime non è supportato per il sistema operativo Windows.
- Quando usi Amazon ECS Exec su Fargate, devi specificare il nome del contenitore perché il GuardDuty security agent funziona come contenitore secondario.
- Non puoi usare Amazon ECS Exec nel contenitore sidecar del GuardDuty Security Agent.
- L'utente IAM che controlla il Runtime Monitoring a livello di cluster deve disporre delle autorizzazioni IAM appropriate per l'etichettatura. Per ulteriori informazioni, consulta il [tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag](#) nella IAM User Guide.
- Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire l'agente di GuardDuty sicurezza, che è archiviato in un repository privato Amazon ECR, per tuo conto.

Utilizzo delle risorse

Il tag che aggiungi al cluster viene conteggiato ai fini della quota di tag del cluster.

Il contenitore GuardDuty Agent Sidecar non viene conteggiato ai fini della quota di definizione dei contenitori per attività.

Come con la maggior parte dei software di sicurezza, c'è un leggero sovraccarico per GuardDuty. Per informazioni sui limiti di memoria di Fargate, consultate Limiti di [CPU e memoria nella Guida](#) per l'GuardDuty utente. Per informazioni sui limiti di EC2 memoria di Amazon, consulta [Limite di CPU e memoria per GuardDuty l'agente](#).

Monitoraggio del runtime per carichi di lavoro Amazon ECS Fargate

Se si utilizzano istanze di EC2 container, è necessario configurare manualmente il Runtime Monitoring. Per ulteriori informazioni, consulta [Monitoraggio del runtime per EC2 carichi di lavoro su Amazon ECS](#).

È possibile GuardDuty gestire il security agent sulle istanze del contenitore. Questa opzione è disponibile solo per Fargate. Questa opzione (gestione degli GuardDuty agenti) è disponibile in GuardDuty

Quando si utilizza la gestione degli GuardDuty agenti, GuardDuty esegue le seguenti operazioni:

- Crea endpoint VPC per GuardDuty ogni VPC che ospita un cluster.
- Recupera e installa il GuardDuty security agent più recente come contenitore secondario per tutte le nuove attività autonome di Fargate e le nuove implementazioni di servizi.

La distribuzione di un nuovo servizio avviene la prima volta che si avvia un servizio o quando si aggiorna un servizio esistente con l'opzione Force New Deployment.

Attivazione del monitoraggio del runtime per Amazon ECS

È possibile GuardDuty configurare la gestione automatica del security agent per tutti i cluster Fargate.

Prerequisiti

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere 1.4.0 o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
 - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
 - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il [tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag](#) nella IAM User Guide.
- Connessione al repository Amazon ECR:

Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:

- Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per

ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia Amazon ECR \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.

- Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

[Per ulteriori informazioni, consulta Subnet privata e gateway NAT.](#)

- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per GuardDuty. Per ulteriori informazioni, consulta la pagina relativa alle [autorizzazioni dei ruoli collegati ai servizi GuardDuty](#) nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo 755 su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze dei EC2 container:

- È necessario utilizzare una versione 20230929 o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione 1.77 o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione 5.10 del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta [Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring](#).
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, vedere [Configurazione di Systems Manager per EC2 le istanze](#) nella Guida per l'AWS Systems Manager Session Manager utente.

Procedura

È possibile abilitare il monitoraggio del runtime in GuardDuty. Per informazioni su come abilitare la funzionalità, consulta [Enabling Runtime Monitoring](#) nella Amazon GuardDuty User Guide.

Aggiungere il monitoraggio del runtime alle attività esistenti di Amazon ECS Fargate

Quando attivi il Runtime Monitoring, tutte le nuove attività autonome e le nuove implementazioni di servizi nel cluster vengono protette automaticamente. Al fine di preservare il vincolo di immutabilità, le attività esistenti non vengono influenzate.

Prerequisiti

1. Attiva il monitoraggio del runtime. Per ulteriori informazioni, consulta [Attivazione del monitoraggio del runtime per Amazon ECS](#).
2. Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Procedura

- Per proteggere immediatamente un'attività, devi eseguire una delle seguenti azioni:
 - Per le attività autonome, interrompi le attività e quindi avviale. Per ulteriori informazioni, consulta [Interruzione di un'attività Amazon ECS](#) e [Esecuzione di un'applicazione come attività Amazon ECS](#).
 - Per le attività che fanno parte di un servizio, aggiorna il servizio con l'opzione «forza la nuova distribuzione». Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Rimuovere il monitoraggio del runtime da un cluster Amazon ECS

Potresti voler escludere determinati cluster dalla protezione, ad esempio i cluster che usi per i test. Ciò comporta GuardDuty l'esecuzione delle seguenti operazioni sulle risorse del cluster:

- Non utilizzate più il GuardDuty Security Agent per nuove attività autonome di Fargate o per nuove implementazioni di servizi.

Al fine di preservare il vincolo di immutabilità, le attività e le distribuzioni esistenti con Runtime Monitoring abilitato non ne risentono.

- Interrompi la fatturazione e non accetta più eventi di runtime per le attività.

Procedura

Esegui i passaggi seguenti per rimuovere Runtime Monitoring da un cluster.

1. Usa la console Amazon ECS o imposta AWS CLI la chiave del GuardDutyManaged tag sul cluster su. false Per ulteriori informazioni, consulta [Aggiornamento di un cluster](#) o [Utilizzo dei tag utilizzando la CLI o l'API](#). Usa i seguenti valori per il tag.

 Note

La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

Chiave =GuardDutyManaged, Valore = false

2. Elimina l'endpoint GuardDuty VPC per il cluster. Per ulteriori informazioni su come eliminare gli endpoint VPC, consulta [Eliminare un endpoint di interfaccia](#) nella Guida per l'utente.AWS PrivateLink

Rimuovere Runtime Monitoring for Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disattiva la funzionalità in GuardDuty. Per informazioni su come disabilitare la funzionalità, consulta [Enabling Runtime Monitoring](#) nella Amazon GuardDuty User Guide.

GuardDuty esegue le seguenti operazioni:

- Elimina gli endpoint VPC GuardDuty per ogni VPC che ospita un cluster.
- Non distribuisce più il GuardDuty Security Agent a nuove attività autonome di Fargate o a nuove implementazioni di servizi.

Al fine di preservare il vincolo di immutabilità, le attività e le implementazioni esistenti non vengono influenzate finché non vengono interrotte, replicate o ridimensionate.

- Interrompe la fatturazione e non accetta più eventi di esecuzione per le attività.

Monitoraggio del runtime per EC2 carichi di lavoro su Amazon ECS

Utilizza questa opzione quando utilizzi EC2 le istanze per la tua capacità o quando hai bisogno di un controllo granulare del monitoraggio del runtime a livello di cluster su Fargate.

È possibile effettuare il provisioning dei cluster per il monitoraggio del runtime aggiungendo un tag predefinito.

Per le istanze dei EC2 container, si scarica, si installa e si gestisce il GuardDuty security agent.

Per Fargate, GuardDuty gestisce l'agente di sicurezza per tuo conto.

Attivazione del monitoraggio del runtime per Amazon ECS

Puoi attivare il Runtime Monitoring per i cluster con EC2 istanze o quando hai bisogno di un controllo granulare del Runtime Monitoring a livello di cluster su Fargate.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere 1.4.0 o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
 - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
 - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il [tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag](#) nella IAM User Guide.
- Connessione al repository Amazon ECR:

Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:

- Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia Amazon ECR \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.
- Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

[Per ulteriori informazioni, consulta Subnet privata e gateway NAT.](#)

- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per GuardDuty. Per ulteriori informazioni, consulta la pagina relativa alle [autorizzazioni dei ruoli collegati ai servizi GuardDuty](#) nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo su 755.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze dei EC2 container:

- È necessario utilizzare una versione 20230929 o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione 1.77 o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione 5.10 del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta [Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring](#).
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, vedere [Configurazione di Systems Manager per EC2 le istanze](#) nella Guida per l'AWS Systems Manager Session Manager utente.

Si attiva Runtime Monitoring in GuardDuty. Per informazioni su come abilitare la funzionalità, consulta [Enabling Runtime Monitoring](#) nella Amazon GuardDuty User Guide.

Aggiungere il monitoraggio del runtime di un cluster Amazon ECS

Configura il monitoraggio del runtime per il cluster, quindi installa il GuardDuty security agent sulle istanze del EC2 contenitore.

Prerequisiti

1. Attiva il monitoraggio del runtime. Per ulteriori informazioni, consulta [Attivazione del monitoraggio del runtime per Amazon ECS](#).
2. Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il [tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag](#) nella IAM User Guide.

Procedura

Esegui le seguenti operazioni per aggiungere Runtime Monitoring a un cluster.

1. Crea un endpoint VPC GuardDuty per ogni VPC del cluster. Per ulteriori informazioni, consulta [Creazione manuale di un endpoint Amazon VPC](#) nella Guida per l'GuardDuty utente.
2. Configura le istanze del EC2 contenitore.

- a. Aggiorna l'agente Amazon ECS alla versione 1.77 o successiva sulle istanze di EC2 container nel cluster. Per ulteriori informazioni, consulta [Aggiornamento dell'agente del container Amazon ECS](#).
- b. Installa il GuardDuty security agent sulle istanze del EC2 contenitore nel cluster. Per ulteriori informazioni, consulta [Gestire manualmente il security agent su un' EC2 istanza Amazon](#) nella Guida per l'GuardDuty utente.

Tutte le attività e le distribuzioni nuove ed esistenti sono immediatamente protette perché il GuardDuty security agent viene eseguito come processo sull'istanza del EC2 contenitore.

3. Usa la console Amazon ECS o imposta AWS CLI la chiave del GuardDutyManaged tag sul cluster su `true`. Per ulteriori informazioni, consulta [Aggiornamento di un cluster](#) o [Utilizzo dei tag utilizzando la CLI](#) o l'API. Usa i seguenti valori per il tag.

Note

La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

Chiave = GuardDutyManaged, Valore = `true`

Aggiungere il monitoraggio del runtime alle attività esistenti di Amazon ECS

Quando attivi il Runtime Monitoring, tutte le nuove attività autonome e le nuove implementazioni di servizi nel cluster vengono protette automaticamente. Al fine di preservare il vincolo di immutabilità, le attività esistenti non vengono influenzate.

Prerequisiti

- Attiva il monitoraggio del runtime. Per ulteriori informazioni, consulta [Attivazione del monitoraggio del runtime per Amazon ECS](#).

Procedura

- Per proteggere immediatamente un'attività, è necessario eseguire una delle seguenti azioni:

- Per le attività autonome, interrompi le attività e quindi avviale. Per ulteriori informazioni, consultare [Interruzione di un'attività Amazon ECS](#) e [Esecuzione di un'applicazione come attività Amazon ECS](#).
- Per le attività che fanno parte di un servizio, aggiorna il servizio con l'opzione «forza la nuova distribuzione». Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS tramite la console](#).

Rimuovere il monitoraggio del runtime da un cluster Amazon ECS

Puoi rimuovere Runtime Monitoring da un cluster. Ciò GuardDuty causa l'interruzione del monitoraggio di tutte le risorse nel cluster.

Per rimuovere Runtime Monitoring da un cluster

1. Usa la console Amazon ECS o imposta AWS CLI la chiave del GuardDutyManaged tag sul cluster su. `false` Per ulteriori informazioni, consulta [Aggiornamento di un cluster](#) o [Utilizzo dei tag utilizzando la CLI o l'API](#).

Note

La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

Chiave =GuardDutyManaged, Valore = `false`

2. Disinstalla il GuardDuty security agent sulle istanze del EC2 contenitore nel cluster.

Per ulteriori informazioni, consulta [Disinstallazione manuale del security agent nella Guida](#) per l'GuardDuty utente.

3. Elimina l'endpoint GuardDuty VPC per ogni VPC del cluster. Per ulteriori informazioni su come eliminare gli endpoint VPC, consulta [Eliminare un endpoint di interfaccia](#) nella Guida per l'utente.AWS PrivateLink

Aggiornamento del GuardDuty security agent sulle istanze di container Amazon ECS

Per informazioni su come aggiornare il GuardDuty security agent sulle istanze del EC2 container, consulta [Updating GuardDuty security agent](#) nella Amazon GuardDuty User Guide.

Rimuovere Runtime Monitoring for Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disattiva la funzionalità in GuardDuty. Per informazioni su come disabilitare la funzionalità, consulta [Enabling Runtime Monitoring](#) nella Amazon GuardDuty User Guide.

Rimuovi Runtime Monitoring da tutti i cluster. Per ulteriori informazioni, consulta [Rimuovere il monitoraggio del runtime da un cluster Amazon ECS](#).

Risoluzione dei problemi di monitoraggio del runtime

Potrebbe essere necessario risolvere i problemi o verificare che il Runtime Monitoring sia abilitato e in esecuzione sulle attività e sui contenitori.

Argomenti

- [Come posso sapere se Runtime Monitoring è attivo sul mio account?](#)
- [Come posso sapere se il Runtime Monitoring è attivo su un cluster?](#)
- [Come posso sapere se il GuardDuty security agent sta eseguendo un'attività di Fargate?](#)
- [Come posso sapere se il GuardDuty security agent è in esecuzione su un'istanza del EC2 contenitore?](#)
- [Cosa succede quando non esiste un ruolo di esecuzione dell'attività per un'attività in esecuzione nel cluster?](#)
- [Come posso sapere se dispongo delle autorizzazioni corrette per etichettare i cluster per il Runtime Monitoring?](#)
- [Cosa succede in assenza di connessione Amazon ECR?](#)
- [Come posso risolvere gli errori di esaurimento della memoria nelle mie attività di Fargate dopo aver abilitato il Runtime Monitoring?](#)

Come posso sapere se Runtime Monitoring è attivo sul mio account?

Nella console Amazon ECS, le informazioni si trovano nella pagina Impostazioni account.

Puoi anche correre `list-account-settings` con l'`effective-settings` opzione.

```
aws ecs list-account-settings --effective-settings
```

Output

L'impostazione con nome impostato `guardDutyActivate` e valore impostato su `on` indica che l'account è configurato. È necessario verificare con GuardDuty l'amministratore se la gestione è automatica o manuale.

```
{
  "setting": {
    "name": "guardDutyActivate",
    "value": "enabled",
    "principalArn": "arn:aws:iam::123456789012:root",
    "type": "aws-managed"
  }
}
```

Come posso sapere se il Runtime Monitoring è attivo su un cluster?

Puoi rivedere le statistiche di copertura nella GuardDuty console. Ciò include le informazioni sulle risorse Amazon ECS associate al tuo account o i tuoi account membro sono la percentuale di cluster integri rispetto a tutti i cluster selezionati. Regione AWS Ciò include la copertura per i cluster che utilizzano Fargate EC2 e i tipi di lancio. Per ulteriori informazioni, consulta la sezione [Revisione delle statistiche di copertura](#) nella Amazon GuardDuty User Guide.

Come posso sapere se il GuardDuty security agent sta eseguendo un'attività di Fargate?

L'agente GuardDuty di sicurezza funge da contenitore secondario per le attività di Fargate.

Nella console Amazon ECS, il sidecar viene visualizzato in Contenitori nella pagina dei dettagli dell'attività.

Puoi eseguire `describe-tasks` e cercare il contenitore con un nome impostato su `aws-gd-agent` e `lastStatus` impostato su `RUNNING`

L'esempio seguente mostra l'output per il cluster predefinito per l'attività `aws:ecs:us-east-1:123456789012:task/0b69d5c0-d655-4695-98cd-5d2d5EXAMPLE`.

```
aws ecs describe-tasks --cluster default --tasks aws:ecs:us-east-1:123456789012:task/0b69d5c0-d655-4695-98cd-5d2d5EXAMPLE
```

Output

Il contenitore denominato `gd-agent` si trova nello `RUNNING` stato.

```
"containers": [  
  {  
    "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/4df26bb4-f057-467b-a079-96167EXAMPLE",  
    "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/0b69d5c0-d655-4695-98cd-5d2d5EXAMPLE",  
    "lastStatus": "RUNNING",  
    "healthStatus": "UNKNOWN",  
    "memory": "string",  
    "name": "aws-gd-agent"  
  }  
]
```

Come posso sapere se il GuardDuty security agent è in esecuzione su un'istanza del EC2 contenitore?

Esegui il comando seguente per visualizzare lo stato:

```
sudo systemctl status amazon-guardduty-agent
```

Il file di registro si trova nella seguente posizione:

```
/var/log/amzn-guardduty-agent
```

Cosa succede quando non esiste un ruolo di esecuzione dell'attività per un'attività in esecuzione nel cluster?

Per le attività di Fargate, l'attività inizia senza il contenitore sidecar del GuardDuty Security Agent. La GuardDuty dashboard mostrerà che all'attività manca la protezione nella dashboard delle statistiche di copertura.

Come posso sapere se dispongo delle autorizzazioni corrette per etichettare i cluster per il Runtime Monitoring?

Per etichettare un cluster, è necessario disporre dell'`ecs:TagResource` azione per entrambi `CreateCluster` e `UpdateCluster`.

Di seguito è riportato un frammento di una policy di esempio.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:TagResource"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecs:CreateAction" : "CreateCluster",
          "ecs:CreateAction" : "UpdateCluster",
        }
      }
    }
  ]
}
```

Cosa succede in assenza di connessione Amazon ECR?

Per le attività di Fargate, l'attività inizia senza il contenitore sidecar del GuardDuty Security Agent. La GuardDuty dashboard mostrerà che all'attività manca la protezione nella dashboard delle statistiche di copertura.

Come posso risolvere gli errori di esaurimento della memoria nelle mie attività di Fargate dopo aver abilitato il Runtime Monitoring?

Il GuardDuty security agent è un processo leggero. Tuttavia, il processo consuma ancora risorse in base alle dimensioni del carico di lavoro. Ti consigliamo di utilizzare strumenti di tracciamento delle risorse dei container, come Amazon CloudWatch Container Insights, per organizzare le GuardDuty distribuzioni nel cluster. Questi strumenti ti aiutano a scoprire il profilo di consumo del GuardDuty security agent per le tue applicazioni. È quindi possibile modificare le dimensioni dell'attività Fargate, se necessario, per evitare potenziali condizioni di esaurimento della memoria.

Monitora i contenitori Amazon ECS con ECS Exec

Con Amazon ECS Exec, puoi interagire direttamente con i container senza dover prima interagire con il sistema operativo del container host, aprire le porte in entrata o gestire le chiavi SSH. Puoi usare ECS Exec per eseguire comandi o inserire una shell in un contenitore in esecuzione su un'istanza Amazon EC2 o su AWS Fargate. In questo modo è più semplice raccogliere informazioni diagnostiche e risolvere rapidamente gli errori. Ad esempio, in un contesto di sviluppo, è possibile utilizzare ECS Exec per interagire facilmente con vari processi nei container e risolvere i problemi delle applicazioni. Inoltre, negli scenari di produzione, puoi utilizzarlo per ottenere un accesso ininterrotto ai contenitori per risolvere i problemi.

Puoi eseguire comandi in un contenitore Linux o Windows in esecuzione utilizzando ECS Exec dall'API Amazon ECS, AWS Command Line Interface (AWS CLI) o dalla CLI AWS SDKs AWS Copilot. Per informazioni dettagliate sull'utilizzo di ECS Exec e per una guida video sull'utilizzo della CLI di Copilot, consulta Copilot AWS [GitHub documentazione](#).

È inoltre possibile utilizzare ECS Exec per mantenere politiche di controllo degli accessi più rigorose. Attivando selettivamente questa funzionalità, è possibile controllare chi può eseguire comandi e quali processi possono eseguire tali comandi. Con un registro di ogni comando e del relativo output, puoi usare ECS Exec per visualizzare quali attività sono state eseguite e puoi usarlo per controllare chi ha avuto accesso CloudTrail a un contenitore.

Considerazioni

Per questo argomento, è necessario acquisire familiarità con i seguenti aspetti relativi all'utilizzo di ECS Exec:

- ECS Exec non è attualmente supportato utilizzando AWS Management Console

La pagina dei dettagli dei cluster mostra la registrazione e la chiave gestita AWS KMS dal cliente utilizzate per crittografare i dati tra il client locale e il contenitore.

- ECS Exec potrebbe non funzionare come previsto quando viene eseguito su sistemi operativi non supportati da Systems Manager. Per informazioni sui sistemi operativi supportati, consulta [Tipi di sistema operativo nella Guida](#) per l'AWS Systems Manager utente.
- ECS Exec è supportato per le attività eseguite sulla seguente infrastruttura:
 - Contenitori Linux su Amazon EC2 su qualsiasi AMI ottimizzata per Amazon ECS, incluso Bottlerocket

- Contenitori Linux e Windows su istanze esterne (Amazon ECS Anywhere)
- Contenitori Linux e Windows su AWS Fargate
- Contenitori Windows su Amazon EC2 sui seguenti sistemi ottimizzati per Windows Amazon ECS AMIs (con la versione Container Agent 1.56 o successiva):
 - AMI Windows Server 2022 Full ottimizzata per Amazon ECS
 - AMI Windows Server 2022 Core ottimizzata per Amazon ECS
 - AMI Windows Server 2019 Full ottimizzata per Amazon ECS
 - AMI Windows Server 2019 Core ottimizzata per Amazon ECS
 - AMI Windows Server 20H2 Core ottimizzata per Amazon ECS
- Se hai configurato un proxy HTTP per la tua attività, imposta la variabile di `NO_PROXY` ambiente su `"NO_PROXY=169.254.169.254,169.254.170.2"` aggirare il proxy, ad EC2 esempio i metadati e il traffico dei ruoli IAM. Se non configuri la variabile di `NO_PROXY` ambiente, possono verificarsi errori durante il recupero dei metadati dell'istanza o delle credenziali del ruolo IAM dall'endpoint di metadati all'interno del contenitore. L'impostazione della variabile di `NO_PROXY` ambiente come consigliato filtra i metadati e il traffico IAM in modo che le richieste non passino attraverso il proxy. `169.254.169.254` and `169.254.170.2` HTTP
- ECS Exec e Amazon VPC
 - Se utilizzi gli endpoint dell'interfaccia Amazon VPC con Amazon ECS, devi creare tali endpoint per Session Manager di Systems Manager (`ssmmessages`). Per ulteriori informazioni sugli endpoint VPC di Systems Manager, vedere [Utilizzare AWS PrivateLink per configurare un endpoint VPC per Session Manager nella Guida per l'utente AWS Systems Manager](#)
 - Se utilizzi gli endpoint dell'interfaccia Amazon VPC con Amazon ECS e AWS KMS key per la crittografia, devi creare tali endpoint per AWS KMS key. Per ulteriori informazioni, consulta [Connessione a AWS KMS key mediante un endpoint VPC](#) nella Guida per gli sviluppatori di AWS Key Management Service .
 - Se hai attività che vengono eseguite su EC2 istanze Amazon, utilizza la modalità `aws_vpc` di rete. Se non disponi di accesso a Internet (ad esempio se non sei configurato per utilizzare un gateway NAT), devi creare l'interfaccia Amazon VPC (endpoint) per Systems Manager Session Manager (`ssmmessages`). Per ulteriori informazioni sulle considerazioni relative alla modalità di rete `aws_vpc`, consulta [Considerazioni](#). Per ulteriori informazioni sugli endpoint VPC di Systems Manager, vedere [Utilizzare AWS PrivateLink per configurare un endpoint VPC per Session Manager nella Guida per l'utente AWS Systems Manager](#)
- ECS Exec e SSM

- Quando un utente esegue comandi su un container utilizzando ECS Exec, questi comandi vengono eseguiti come utente `root`. SSM Agent e i relativi processi figlio vengono eseguiti come `root` anche quando si specifica un ID utente per il container.
- L'agente SSM richiede la possibilità di scrivere sul file system del contenitore per creare le directory e i file richiesti. Pertanto, la specifica del file system `root` di sola lettura utilizzando il parametro di definizione di attività `readonlyRootFilesystem`, o qualsiasi altro metodo, non è supportata.
- Sebbene l'avvio di sessioni di SSM al di fuori dell'operazione `execute-command` sia possibile, le sessioni non saranno registrate e saranno conteggiate rispetto al limite di sessione. Si consiglia di limitare questo accesso negando l'operazione `ssm:start-session` con una policy IAM. Per ulteriori informazioni, consulta [Limitazione dell'accesso all'operazione Avvia sessione](#).
- Le seguenti funzionalità funzionano come contenitore `sidecar`. Pertanto, è necessario specificare il nome del contenitore su cui eseguire il comando.
 - Monitoraggio del runtime
 - Service Connect
- Gli utenti possono eseguire tutti i comandi disponibili nel contesto del container. Le seguenti operazioni potrebbero causare processi orfani e zombie: terminare il processo principale del container, terminare l'agente dei comandi ed eliminare le dipendenze. Per ripulire i processi zombie, ti consigliamo di aggiungere il flag `initProcessEnabled` alla definizione di attività.
- ECS Exec utilizza parte della CPU e della memoria. Sarà necessario adattare questi valori quando si specificano le allocazioni di risorse CPU e memoria nella definizione di attività.
- È necessario utilizzare la AWS CLI versione 1.22.3 o successiva o la AWS CLI versione 2.3.6 o successiva. Per informazioni su come aggiornare il AWS CLI, vedere [Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente versione 2](#).
- Puoi avere solo una sessione ECS Exec per ogni spazio dei nomi dell'ID processo (PID). Se [condividi uno spazio dei nomi PID in un'attività](#), puoi avviare le sessioni ECS Exec solo in un container.
- La sessione di ECS Exec ha un valore di timeout di inattività pari a 20 minuti. Questo valore non può essere modificato.
- Non è possibile attivare ECS Exec per le attività esistenti. Il servizio può essere attivato solo per le nuove attività.
- Non è possibile utilizzare ECS Exec quando si avvia un'attività su un cluster che utilizza la scalabilità gestita con posizionamento asincrono (avvia un'attività senza istanza). `run-task`

- Non è possibile eseguire ECS Exec su contenitori Microsoft Nano Server.

Prerequisiti

Prima di iniziare a utilizzare ECS Exec, assicurati di aver completato queste azioni:

- Installa e configura la AWS CLI. Per ulteriori informazioni, consulta [Get started with the AWS CLI](#).
- Installa il plug-in Session Manager per AWS CLI. Per ulteriori informazioni, consulta [Installazione del plug-in Session Manager per AWS CLI](#).
- Devi utilizzare un ruolo attività con le autorizzazioni appropriate per ECS Exec. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività](#).
- ECS Exec ha requisiti di versione a seconda che le attività siano ospitate su Amazon EC2 o: AWS Fargate
 - Se utilizzi Amazon EC2, devi utilizzare un'AMI ottimizzata per Amazon ECS rilasciata dopo il 20 gennaio 2021, con una versione agente 1.50.2 o successiva. Per ulteriori informazioni, consulta [Amazon ECS optimized AMIs](#).
 - Se utilizzi AWS Fargate, devi utilizzare una versione della piattaforma 1.4.0 o superiore (Linux) o 1.0.0 (Windows). Per ulteriori informazioni, consulta [Versioni della piattaforma AWS Fargate](#).

Architettura

ECS Exec utilizza AWS Systems Manager (SSM) Session Manager per stabilire una connessione con il contenitore in esecuzione e utilizza le policy AWS Identity and Access Management (IAM) per controllare l'accesso ai comandi in esecuzione in un contenitore in esecuzione. Ciò è reso possibile collegando i file binari di SSM Agent necessari nel container. L'Amazon ECS o AWS Fargate l'agente è responsabile dell'avvio dell'agente principale SSM all'interno del contenitore insieme al codice dell'applicazione. Per ulteriori informazioni, consulta [Systems Manager Session Manager](#).

Puoi controllare quale utente ha effettuato l'accesso al contenitore utilizzando l'ExecuteCommand evento AWS CloudTrail e registrare ogni comando (e il relativo output) su Amazon S3 o Amazon CloudWatch Logs. Per crittografare i dati tra il client locale e il contenitore con la tua chiave di crittografia, devi fornire la chiave AWS Key Management Service (AWS KMS).

Utilizzo di ECS Exec

Modifiche facoltative alla definizione di attività

Se si imposta il parametro di definizione dell'attività `initProcessEnabled` su `true`, viene avviato il processo di inizializzazione all'interno del contenitore. Ciò rimuove tutti i processi secondari rilevati dall'agente SSM zombie. Il seguente comando fornisce un esempio.

```
{
  "taskRoleArn": "ecsTaskRole",
  "networkMode": "awsvpc",
  "requiresCompatibilities": [
    "EC2",
    "FARGATE"
  ],
  "executionRoleArn": "ecsTaskExecutionRole",
  "memory": ".5 gb",
  "cpu": ".25 vcpu",
  "containerDefinitions": [
    {
      "name": "amazon-linux",
      "image": "amazonlinux:latest",
      "essential": true,
      "command": ["sleep", "3600"],
      "linuxParameters": {
        "initProcessEnabled": true
      }
    }
  ],
  "family": "ecs-exec-task"
}
```

Attivazione di ECS Exec per attività e servizi

Puoi attivare la funzionalità ECS Exec per i tuoi servizi e le tue attività autonome specificando il `--enable-execute-command` flag quando usi uno dei seguenti AWS CLI comandi:, o. [create-serviceupdate-servicestart-taskrun-task](#)

Ad esempio, se si esegue il comando seguente, la funzionalità ECS Exec viene attivata per un servizio appena creato che viene eseguito su Fargate. Per ulteriori informazioni sulla creazione dei servizi, consulta [create-service](#).

```
aws ecs create-service \
  --cluster cluster-name \
  --task-definition task-definition-name \
  --enable-execute-command \
  --service-name service-name \
  --launch-type FARGATE \
  --network-configuration
  "awsVpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=ENI"
  \
  --desired-count 1
```

Dopo aver abilitato ECS Exec per un'attività, è possibile emettere il comando seguente per confermare che l'attività è pronta per l'uso. Se la proprietà `lastStatus` di `ExecuteCommandAgent` è riportata come `RUNNING` e la proprietà `enableExecuteCommand` è impostata su `true`, allora il processo è pronto.

```
aws ecs describe-tasks \
  --cluster cluster-name \
  --tasks task-id
```

Il seguente frammento di output è un esempio di cosa potresti vedere.

```
{
  "tasks": [
    {
      ...
      "containers": [
        {
          ...
          "managedAgents": [
            {
              "lastStartedAt": "2021-03-01T14:49:44.574000-06:00",
              "name": "ExecuteCommandAgent",
              "lastStatus": "RUNNING"
            }
          ]
        }
      ],
      ...
      "enableExecuteCommand": true,
      ...
    }
  ]
}
```

```
]
}
```

Esecuzione di comandi tramite ECS Exec

Dopo aver confermato che `ExecuteCommandAgent` è in esecuzione, è possibile aprire una shell interattiva sul container utilizzando il seguente comando. Se il processo contiene più container, è necessario specificare il nome del container utilizzando il flag `--container`. Amazon ECS supporta solo l'avvio di sessioni interattive, pertanto è necessario utilizzare il flag `--interactive`.

Il comando seguente eseguirà un `/bin/sh` comando interattivo su un contenitore denominato *container-name* per un'attività con un ID di *task-id*

task-id è l'Amazon Resource Name (ARN) dell'attività.

```
aws ecs execute-command --cluster cluster-name \
  --task task-id \
  --container container-name \
  --interactive \
  --command "/bin/sh"
```

Registrazione tramite ECS Exec

Attivazione della registrazione delle attività e dei servizi

Important

[Per ulteriori informazioni sui CloudWatch prezzi, consulta la sezione Prezzi. CloudWatch](#)

Amazon ECS offre anche parametri di monitoraggio che vengono forniti senza costi aggiuntivi. Per ulteriori informazioni, consulta [Monitora Amazon ECS utilizzando CloudWatch](#).

Amazon ECS fornisce una configurazione predefinita per i comandi di registrazione eseguiti con ECS Exec. L'impostazione predefinita prevede l'invio di log a CloudWatch Logs utilizzando il driver di `awslogs` log configurato nella definizione dell'attività. Se si desidera fornire una configurazione personalizzata, AWS CLI supporta un `--configuration` flag sia per i comandi che per `create-cluster` `update-cluster`. L'immagine del contenitore richiede `script` e `cat` deve essere installata per poter caricare correttamente i log dei comandi su Amazon S3 CloudWatch o Logs. Per ulteriori informazioni sulla creazione dei cluster, consulta [create-cluster](#).

 Note

Questa configurazione gestisce solo la registrazione della sessione `execute-command`. Non influisce sulla registrazione della tua applicazione.

L'esempio seguente crea un cluster e quindi registra l'output nei tuoi CloudWatch Logs LogGroup named `cloudwatch-log-group-name` e nel tuo bucket Amazon S3 denominato `s3-bucket-name`

È necessario utilizzare una chiave gestita AWS KMS dal cliente per crittografare il gruppo di log quando si imposta l'opzione `CloudWatchEncryptionEnabled true`. Per informazioni su come crittografare il gruppo di log, consulta [Crittografare i dati di registro in CloudWatch Logs using AWS Key Management Service](#), nella Guida per l'utente Amazon CloudWatch Logs

```
aws ecs create-cluster \
  --cluster-name cluster-name \
  --configuration executeCommandConfiguration="{ \
    kmsKeyId=string, \
    logging= OVERRIDE, \
    logConfiguration={ \
      cloudWatchLogGroupName=cloudwatch-log-group-name, \
      cloudWatchEncryptionEnabled=true, \
      s3BucketName=s3-bucket-name, \
      s3EncryptionEnabled=true, \
      s3KeyPrefix=demo \
    } \
  }"
```

La proprietà `logging` determina il comportamento della funzionalità di registrazione di ECS Exec:

- `NONE`: la registrazione è disattivata.
- `DEFAULT`: i log vengono inviati al driver `awslogs` configurato. Se il driver non è configurato, non viene salvato alcun registro.
- `OVERRIDE`: i log vengono inviati al bucket Amazon CloudWatch Logs fornito LogGroup, Amazon S3 o a entrambi.

Autorizzazioni IAM richieste per Amazon CloudWatch Logs o Amazon S3 Logging

Per abilitare la registrazione, il ruolo del processo Amazon ECS a cui si fa riferimento nella definizione di attività deve disporre di autorizzazioni aggiuntive. Queste autorizzazioni aggiuntive possono essere aggiunte al ruolo del processo sotto forma di policy. Sono diversi a seconda che tu indirizzi i log ad Amazon CloudWatch Logs o Amazon S3.

Amazon CloudWatch Logs

La seguente policy di esempio aggiunge le autorizzazioni Amazon CloudWatch Logs richieste.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogStream",
        "logs:DescribeLogStreams",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:region:account-id:log-group:/aws/ecs/cloudwatch-log-group-name:"
    }
  ]
}
```

Amazon S3

La policy dell'esempio seguente aggiunge le autorizzazioni Amazon S3 necessarie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```

        "Action": [
            "s3:GetBucketLocation"
        ],
        "Resource": "*"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:GetEncryptionConfiguration"
        ],
        "Resource": "arn:aws:s3:::s3-bucket-name"
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:PutObject"
        ],
        "Resource": "arn:aws:s3:::s3-bucket-name/*"
    }
]
}

```

Autorizzazioni IAM necessarie per la crittografia utilizzando le proprie AWS KMS key (chiave KMS)

Per impostazione predefinita, i dati trasferiti tra il client locale e il contenitore utilizzano la crittografia TLS 1.2 che fornisce. AWS Per crittografare ulteriormente i dati utilizzando la propria chiave KMS, è necessario creare una chiave KMS e aggiungere l'autorizzazione `kms:Decrypt` al ruolo IAM del processo. Questa autorizzazione sarà utilizzata dal tuo container per decrittare i dati. Per ulteriori informazioni sulla creazione di una chiave KMS, consulta [Creazione di chiavi](#).

Aggiungi la seguente policy in linea al tuo ruolo Task IAM che richiede le AWS KMS autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni ECS Exec](#).

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ]
    }
  ]
}

```

```
    ],  
    "Resource": "kms-key-arn"  
  }  
]  
}
```

Per crittografare i dati utilizzando la propria chiave KMS, all'utente o al gruppo che utilizza l'operazione `execute-command` deve essere concessa l'autorizzazione `kms:GenerateDataKey`.

La seguente policy di esempio per l'utente o il gruppo contiene l'autorizzazione necessaria per utilizzare la propria chiave KMS. È necessario specificare l'Amazon Resource Name (ARN) della chiave KMS.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "kms:GenerateDataKey"  
      ],  
      "Resource": "kms-key-arn"  
    }  
  ]  
}
```

Utilizzo delle policy IAM per limitare l'accesso a ECS Exec

Limiti l'accesso degli utenti all'azione dell'API `execute-command` utilizzando una o più delle seguenti chiavi di condizione della policy IAM:

- `aws:ResourceTag/clusterTagKey`
- `ecs:ResourceTag/clusterTagKey`
- `aws:ResourceTag/taskTagKey`
- `ecs:ResourceTag/taskTagKey`
- `ecs:container-name`
- `ecs:cluster`
- `ecs:task`
- `ecs:enable-execute-command`

Con la seguente policy IAM di esempio, gli utenti possono eseguire comandi in container in esecuzione all'interno di processi con un tag che dispone di una chiave `environment` e un valore `development` in un cluster denominato `cluster-name`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:ExecuteCommand",
        "ecs:DescribeTasks"
      ],
      "Resource": [
        "arn:aws:ecs:region:aws-account-id:task/cluster-name/*",
        "arn:aws:ecs:region:aws-account-id:cluster/cluster-name"
      ],
      "Condition": {
        "StringEquals": {
          "ecs:ResourceTag/environment": "development"
        }
      }
    }
  ]
}
```

Con l'esempio di policy IAM riportato di seguito, gli utenti non possono utilizzare l'API `execute-command` quando il nome del container è `production-app`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ecs:ExecuteCommand"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecs:container-name": "production-app"
        }
      }
    }
  ]
}
```

```

    }
  }
]
}

```

Con le seguenti policy IAM, gli utenti possono avviare i processi solo quando ECS Exec è disattivato.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:RunTask",
        "ecs:StartTask",
        "ecs:CreateService",
        "ecs:UpdateService"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecs:enable-execute-command": "false"
        }
      }
    }
  ]
}

```

Note

Perché l'operazione API `execute-command` contiene solo risorse di processo e cluster in una richiesta, vengono valutati solo i tag di cluster e processo.

Per ulteriori informazioni sulle chiavi di condizione della policy IAM, consulta [Operazioni, risorse e chiavi di condizione per Amazon Elastic Container Service](#) in Service Authorization Reference.

Limitazione dell'accesso all'operazione Avvia sessione

Mentre è possibile avviare sessioni SSM sul container al di fuori di ECS Exec, ciò potrebbe causare potenzialmente la mancata registrazione delle sessioni. Anche le sessioni avviate al di fuori di ECS

Exec vengono conteggiate per la quota di sessione. Si consiglia di limitare questo accesso negando l'operazione `ssm:start-session` direttamente per i processi Amazon ECS con una policy IAM. Puoi negare l'accesso a tutti i processi Amazon ECS o a processi specifici in base ai tag utilizzati.

Di seguito è riportato un esempio di policy IAM che nega l'accesso all'operazione `ssm:start-session` per i processi in tutte le regioni con un nome cluster specificato. Facoltativamente puoi includere un carattere jolly con *cluster-name*.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ssm:StartSession",
      "Resource": [
        "arn:aws:ecs:region:aws-account-id:task/cluster-name/*"
      ]
    }
  ]
}
```

Di seguito è riportato un esempio di policy IAM che nega l'accesso all'operazione `ssm:start-session` sulle risorse in tutte le regioni con chiave di tag `Task-Tag-Key` e valore di tag `Exec-Task`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ssm:StartSession",
      "Resource": "arn:aws:ecs:*:*:task/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Task-Tag-Key": "Exec-Task"
        }
      }
    }
  ]
}
```

Per assistenza su eventuali problemi che potresti riscontrare durante l'utilizzo di Amazon ECS Exec, consulta [Risoluzione dei problemi con](#) Exec.

AWS Compute Optimizer consigli per Amazon ECS

AWS Compute Optimizer genera consigli per le dimensioni delle attività e dei container di Amazon ECS. Per ulteriori informazioni, consulta [Che cos'è AWS Compute Optimizer?](#) nella Guida per l'utente di AWS Compute Optimizer .

Suggerimenti sulle dimensioni delle attività e dei contenitori per i servizi Amazon ECS su AWS Fargate

AWS Compute Optimizer genera consigli per i servizi Amazon ECS su AWS Fargate. AWS Compute Optimizer consiglia le dimensioni della CPU e della memoria dell'attività e le dimensioni della CPU del contenitore, della memoria del contenitore e della memoria del contenitore di prenotazione. Questi suggerimenti vengono visualizzati nelle pagine seguenti della console Compute Optimizer.

- Pagina Suggerimenti per i servizi Amazon ECS su Fargate
- Pagina Dettagli dei servizi Amazon ECS su Fargate

Per ulteriori informazioni, consulta [Visualizzazione dei suggerimenti per i servizi Amazon ECS su Fargate](#) nella Guida per l'utente di AWS Compute Optimizer .

Risoluzione dei problemi legati ad Amazon ECS

Potrebbe essere necessario risolvere i problemi relativi ai sistemi di bilanciamento del carico, alle attività, ai servizi o alle istanze dei container. Questo capitolo ti aiuta a trovare informazioni di diagnostica nell'agent del container di Amazon ECS, nel daemon Docker sull'istanza di container e nel log di eventi del servizio nella console Amazon ECS.

Per informazioni sulle attività interrotte, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolvi gli errori delle attività interrotte.	Visualizzazione degli errori delle attività interrotte da Amazon ECS	
Visualizza gli errori delle attività interrotte.	Risolvi gli errori relativi alle attività interrotte da Amazon ECS	
Rivedi i codici di errore delle attività interrotte.	Messaggi di errore delle attività interrotte da Amazon ECS	
Rivedi gli errori delle CannotPullContainer attività.	CannotPullContainer errori di attività in Amazon ECS	
Visualizza le richieste di ruoli IAM per le attività.	Visualizzazione delle richieste di ruolo IAM per le attività di Amazon ECS	

Per informazioni sugli errori di servizio, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Visualizza i messaggi relativi agli eventi di servizio.	Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS	

Azione	Ulteriori informazioni	
Rivedi i messaggi relativi agli eventi di servizio.	Messaggi sugli eventi del servizio Amazon ECS	
Esamina i problemi relativi al bilanciamento del carico.	Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS	
Esamina i problemi di scalabilità automatica del servizio.	Risoluzione dei problemi relativi alla scalabilità automatica del servizio in Amazon ECS	

Per informazioni sugli errori di definizione delle attività, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolve l'errore di memoria di definizione dell'attività.	Risolvi gli errori di CPU o memoria non validi relativi alla definizione delle attività di Amazon ECS	

Per informazioni sugli errori dell'agente Amazon ECS, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Visualizza i log degli agenti container Amazon ECS.	Visualizzazione dei log degli agenti container Amazon ECS	
Scopri come raccogliere i log di Amazon ECS.	Raccolta dei log dei container con Amazon ECS logs collector	

Azione	Ulteriori informazioni	
Recupera i dettagli diagnostici con l'agente Amazon ECS.	Recupera i dettagli diagnostici di Amazon ECS con l'introspezione degli agenti	

Per informazioni sugli errori Docker, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Usa la diagnostica Docker.	Diagnostica Docker in Amazon ECS	
Attiva la modalità di debug Docker.	Configurazione dell'output verboso dal demone Docker in Amazon ECS	
Risolvi l'errore 500 dell'API Docker.	Risolvi i problemi relativi al Docker in Amazon API error (500): devmapper ECS	

Per informazioni sugli errori ECS Exec e Amazon ECS Anywhere, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolvi i problemi relativi a ECS Exec.	Risolvi i problemi di Amazon ECS Exec	
Risolvi i problemi relativi ad Amazon ECS Anywhere.	Risolvi i problemi relativi ad Amazon ECS Anywhere	

Per informazioni sui problemi di limitazione, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Scopri di più sulle quote di limitazione di Fargate.	AWS Fargate limitazione delle quote	
Scopri le best practice per la limitazione di Amazon ECS.	Gestisci i problemi di limitazione di Amazon ECS	

Per informazioni sugli errori delle API, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolvi gli errori delle API.	Motivi di errore dell'API Amazon ECS	

Risolvi gli errori relativi alle attività interrotte da Amazon ECS

Quando l'attività non viene avviata, viene visualizzato un messaggio di errore nella console e nei parametri di `describe-tasks` output (`stoppedReason`). `stoppedCode`

È possibile visualizzare le attività interrotte nella console per un'ora. Per visualizzare le attività interrotte, è necessario modificare l'opzione di filtro. Per ulteriori informazioni, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

Le pagine seguenti forniscono informazioni sulle attività interrotte.

- Scopri le modifiche ai messaggi di errore relativi alle attività interrotte.

[Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore delle attività](#)

- Visualizza le attività interrotte in modo da ottenere informazioni sulla causa.

[Visualizzazione degli errori delle attività interrotte da Amazon ECS](#)

- Scopri i messaggi di errore relativi alle attività interrotte e i possibili motivi degli errori.

[Messaggi di errore delle attività interrotte da Amazon ECS](#)

- Scopri come verificare la connettività delle attività interrotte e correggere gli errori.

Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS

Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore delle attività

A partire dal 14 giugno 2024, il team di Amazon ECS modificherà i messaggi di errore dell'attività interrotta come descritto nelle tabelle seguenti. Non stopCode cambieranno. Se le applicazioni dipendono da stringhe di messaggi di errore esatte, è necessario aggiornare le applicazioni con le nuove stringhe. Per assistenza in caso di domande o problemi, contatta. Supporto AWS

Note

Ti consigliamo di non fare affidamento sui messaggi di errore per l'automazione, poiché i messaggi di errore sono soggetti a modifiche.

CannotPullContainerError

Vecchio messaggio di errore	Nuovo messaggio di errore	
CannotPullContainerError: Risposta di errore dal demone: accesso pull negato <i>repository</i> , il repository non esiste o potrebbe richiedere 'accesso docker': negato: Utente: <i>roleARN</i>	CannotPullContainerError: L'operazione non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre immagini dal registro. Risposta di errore dal demone: pull access negato per <i>repositor</i> <i>y</i> , il repository non esiste o potrebbe richiedere 'docker login': denied: User: <i>roleARN</i> is not authorized to perform: ecr: BatchGetI mage on resource: <i>image</i> perché nessuna policy	

Vecchio messaggio di errore	Nuovo messaggio di errore	
	basata sull'identità consente l'azione ecr:. BatchGetImage • CannotPullContainerError: L'attività non può estrarre l'immagine. Controlla se l'immagine esiste. Risposta di errore dal demone: accesso pull negato per <i>repository</i> , il repository non esiste o potrebbe richiedere 'docker login': negato: l'accesso richiesto alla risorsa è negato.	
CannotPullContainerError: Risposta di errore dal demone: Get<i>imageURI</i>: net/http: richiesta annullata in attesa della connessione	CannotPullContainerError: L'operazione non può estrarre l'immagine. Controlla la configurazione della tua rete. Risposta di errore dal demone: Get: net/http<i>image</i>: richiesta annullata in attesa della connessione (Client.Timeout superato in attesa delle intestazioni)	

Vecchio messaggio di errore	Nuovo messaggio di errore	
CannotPullContainerError<ip><port>: ref pull è stato riprovato 5 volte: failed to copy:: failed open: failed to do request: Get: dial tcp httpReadSeeker: i/o timeout <i>registry-uri</i>	CannotPullContainerError: L'operazione non può essere estratta dal registro <i>image-uri . registry-uri</i> C'è un problema di connessione tra l'attività e il registro. Controlla la configurazione della rete dell'attività.: failed to copy: httpReadSeeker: failed open: failed to do request: Get <i>registry-uri</i> : dial tcp<ip>: <port>i/o timeout	

ResourceNotFoundException

Vecchio messaggio di errore	Nuovo messaggio di errore	
Il recupero di dati segreti dalla AWS Secrets Manager <i>region</i> regione: secret: <i>secretARN</i> ResourceNotFoundException: Secrets Manager non riesce a trovare il segreto specificato.	ResourceNotFoundException: l'operazione non può recuperare il segreto con ARN <i>secretARN</i> " da. AWS Secrets Manager Controlla se il segreto esiste nella regione specificata. ResourceNotFoundException: Recupero di dati segreti dalla AWS Secrets Manager regione <i>region</i> : secret: <i>secretARN</i> ResourceNotFoundException: Secrets Manager non riesce a trovare il segreto specificato.	

ResourceInitializationError

Vecchio messaggio di errore	Nuovo messaggio di errore	
ResourceInitializationError <ip><port>: impossibile recuperare i segreti o l'autenticazione del registro: recupero delle risorse di esecuzione e non riuscito: impossibile recuperare l'autenticazione del registro ecr: la chiamata di servizio è stata ritentata 3 volte:: invio della richiesta non riuscito a causa di RequestError: Post "https://api.ecr.us-east-1.amazonaws.com/»: dial tcp:: i/o timeout. Controlla la configurazione della tua rete di attività.	ResourceInitializationError : impossibile recuperare i segreti o l'autenticazione del registro: l'attività non può recuperare l'autenticazione del registro da Amazon ECR: c'è un problema di connessione tra l'attività e Amazon ECR. Controlla la configurazione della rete delle tue attività. RequestError: richiesta di invio non riuscita a causa di: Post "https://api.ecr.us-east-1.amazonaws.com«: dial tcp<ip>:<port>: i/o timeout	
ResourceInitializationError : impossibile recuperare segreti o autenticazione del registro: recupero delle risorse di esecuzione non riuscito: impossibile recuperare e segreti da ssm: la chiamata di servizio è stata ritentata 5 volte:: contesto della richiesta annullato a causa di RequestCanceled: scadenza del contesto superata	ResourceInitializationError: impossibile recuperare segreti o autenticazione del registro: impossibile recuperare segreti da ssm: L'attività non può recuperare segreti da. AWS Systems Manager C'è un problema di connessione tra l'attività e AWS Systems Manager Parameter Store. Controlla la configurazione della rete della tua attività. RequestCanceled: contesto della richiesta annullato a	

Vecchio messaggio di errore	Nuovo messaggio di errore	
ResourceInitializationError : impossibile scaricare i file env: comando di download del file: flusso di errore non vuoto:: contesto della richiesta annullato a causa di RequestCanceled: scadenza del contesto superata	causa del superamento della scadenza del contesto ResourceInitializationError: impossibile scaricare i file env: l'attività non può scaricare i file delle variabili di ambiente da Amazon S3. C'è un problema di connessione tra l'attività e Amazon S3. Controlla la configurazione della rete dell'attività. La chiamata di servizio è stata ritentata 5 volte: RequestCanceled: il contesto della richiesta è stato annullato a causa del superamento della scadenza del contesto	
ResourceInitializationError : impossibile convalidare il logger args: :signal:killed	ResourceInitializationError : failed to validate logger args: L'attività non riesce a trovare il gruppo di CloudWatch log Amazon definito nella definizione dell'attività. C'è un problema di connessione tra l'attività e Amazon CloudWatch. Controlla la configurazione della tua rete.: signal: killed	

Vecchio messaggio di errore	Nuovo messaggio di errore	
ResourceInitializationError : impossibile recuperare i segreti o l'autenticazione del registro: comando pull fallito:: signal: killed	ResourceInitializationError : impossibile recuperare i segreti o l'autenticazione del registro: controlla la configura zione della rete delle attività.: signal: killed	

Visualizzazione degli errori delle attività interrotte da Amazon ECS

Se si verificano dei problemi con l'avvio di un'attività, questa potrebbe essere interrotta a causa di errori dell'applicazione o della configurazione. Ad esempio, quando esegui l'attività, questa visualizza lo stato PENDING e poi scompare.

Se l'attività è stata creata da un servizio Amazon ECS, le azioni intraprese da Amazon ECS per gestire il servizio sono pubblicate negli eventi del servizio. Puoi visualizzare gli eventi nell' AWS Management Console,, AWS CLI AWS SDKs, API Amazon ECS o negli strumenti che utilizzano l'API SDKs and. Questi eventi includono l'arresto e la sostituzione di un'attività da parte di Amazon ECS a causa dell'interruzione dell'esecuzione dei container dell'attività o del superamento di troppi controlli dell'integrità effettuati da Elastic Load Balancing.

Se la tua attività è stata eseguita su un'istanza di container su Amazon EC2 o su computer esterni, puoi anche esaminare i log del runtime del contenitore e dell'agente Amazon ECS. Questi log si trovano sull' EC2 istanza Amazon host o sul computer esterno. Per ulteriori informazioni, consulta [Visualizzazione dei log degli agenti container Amazon ECS](#).

Procedura

Console

AWS Management Console

I seguenti passaggi possono essere utilizzati per verificare la presenza di errori nelle attività interrotte utilizzando la console. Per visualizzare le attività interrotte, è necessario modificare l'opzione di filtro.

Le attività interrotte vengono visualizzate nella console solo per 1 ora.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella *name* pagina Cluster:, scegli la scheda Attività.
5. Configura il filtro per visualizzare le attività interrotte. Per Filtra lo stato desiderato, scegli Interrotto.

L'opzione Interrotto mostra le attività interrotte, mentre Qualsiasi stato desiderato mostra tutte le attività.

6. Scegli il processo interrotto da ispezionare.
7. Nella riga relativa all'attività interrotta, nella colonna Stato più recente, scegli Interrotto.

Una finestra pop-up mostra il motivo dell'interruzione.

AWS CLI

1. Elenca i processi arrestati in un cluster. L'output contiene l'Amazon Resource Name (ARN) del processo, necessario per descrivere il processo.

```
aws ecs list-tasks \
  --cluster cluster_name \
  --desired-status STOPPED \
  --region region
```

2. Descrivi l'attività interrotta per recuperare le informazioni. Per ulteriori informazioni, consulta [describe-tasks](#) nel riferimento. AWS Command Line Interface

```
aws ecs describe-tasks \
  --cluster cluster_name \
  --tasks arn:aws:ecs:region:account_id:task/cluster_name/task_ID \
  --region region
```

Utilizzate i seguenti parametri di output.

- **stopCode**- Il codice di arresto indica, ad esempio, il motivo per cui un'attività è stata interrotta `ResourceInitializationError`
- **StoppedReason**- Il motivo per cui l'attività è stata interrotta.

- `reason`(nella `containers` struttura) - Il motivo fornisce ulteriori dettagli sul contenitore fermo.

Passaggi successivi

Visualizza le attività interrotte in modo da ottenere informazioni sulla causa. Per ulteriori informazioni, consulta [Messaggi di errore delle attività interrotte da Amazon ECS](#).

Messaggi di errore delle attività interrotte da Amazon ECS

Di seguito sono riportati i possibili messaggi di errore che potresti ricevere quando l'attività si interrompe in modo imprevisto.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, vedere. [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#)

Ai codici di errore delle attività interrotte è associata una categoria, ad esempio "ResourceInitializationError". Per ulteriori informazioni su ciascuna categoria, consulta quanto segue:

Categoria	Ulteriori informazioni	
TaskFailedToStart	Risoluzione degli errori di Amazon ECS TaskFailedToStart	
ResourceInitializationError	Risoluzione degli errori di Amazon ECS ResourceInitializationError	
ResourceNotFoundException	Risoluzione degli errori di Amazon ECS ResourceNotFoundException	
SpotInterruptionError	Risoluzione degli errori di Amazon ECS SpotInterruptionError	
InternalError	Risoluzione degli errori di Amazon ECS InternalError	

Categoria	Ulteriori informazioni	
OutOfMemoryError	Risoluzione degli errori di Amazon ECS OutOfMemoryError	
ContainerRuntimeError	Risoluzione degli errori di Amazon ECS ContainerRuntimeError	
ContainerRuntimeTimeoutError	Risoluzione degli errori di Amazon ECS ContainerRuntimeTimeoutError	
CannotStartContainerError	Risoluzione degli errori di Amazon ECS CannotStartContainerError	
CannotStopContainerError	Risoluzione degli errori di Amazon ECS CannotStopContainerError	
CannotInspectContainerError	Risoluzione degli errori di Amazon ECS CannotInspectContainerError	
CannotCreateVolumeError	Risoluzione degli errori di Amazon ECS CannotCreateVolumeError	
CannotPullContainer	CannotPullContainer errori di attività in Amazon ECS	

Risoluzione degli errori di Amazon ECS TaskFailedToStart

Di seguito sono riportati alcuni messaggi TaskFailedToStart di errore e azioni che puoi intraprendere per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

EC2 Errore imprevisto durante il tentativo di creare un'interfaccia di rete con l'assegnazione dell'IP pubblico abilitata nella sottorete '**subnet-id**'

Ciò accade quando un'operazione Fargate che utilizza la modalità di awsipc rete ed è eseguita in una sottorete con un indirizzo IP pubblico e la sottorete non dispone di indirizzi IP sufficienti.

Il numero di indirizzi IP disponibili è disponibile nella pagina dei dettagli della sottorete nella EC2 console Amazon o utilizzando [describe-subnets](#). Per ulteriori informazioni, consulta [Visualizza la sottorete](#) nella Amazon VPC User Guide.

Per risolvere questo problema, puoi creare una nuova sottorete in cui eseguire l'attività.

InternalError: **<reason>**

Questo errore si verifica quando viene richiesto un allegato ENI. Amazon gestisce in modo EC2 asincrono l'approvvigionamento dell'ENI. Il processo di provisioning richiede del tempo. Amazon ECS raggiunge un valore di timeout in caso di lunghi tempi di attesa o errori non segnalati. Ci sono momenti in cui viene eseguito il provisioning dell'ENI, ma il report arriva ad Amazon ECS dopo il timeout dell'errore. In questo caso, Amazon ECS rileva l'errore segnalato dell'attività con un'ENI in uso.

La definizione dell'attività selezionata non è compatibile con la strategia di elaborazione selezionata

Questo errore si verifica quando si sceglie una definizione di attività con un tipo di avvio che non corrisponde al tipo di capacità del cluster. Per ulteriori informazioni, consulta [Tipi di avvio di Amazon ECS](#). È necessario selezionare una definizione di attività che corrisponda al provider di capacità assegnato al cluster.

Impossibile collegare l'interfaccia di rete all'indice dei dispositivi non utilizzati

Questo errore si verifica quando si utilizza un tipo awsipc di rete e non c'è abbastanza CPU/ memoria per l'operazione. Innanzitutto, controlla la CPU per le istanze. Per ulteriori informazioni, consulta le [specifiche dei tipi di EC2 istanze Amazon](#) nei tipi di EC2 istanze Amazon. Prendi il valore della CPU per l'istanza e moltiplicalo ENIs per il numero dell'istanza. Usa quel valore e nella definizione dell'attività.

AGENTE

L'istanza di container su cui hai provato ad avviare un processo ha un agente attualmente disconnesso. Per prevenire lunghi tempi di attesa per il posizionamento delle attività, la richiesta è stata respinta.

Per informazioni su come risolvere i problemi relativi a un agente disconnesso, consulta [How do I troubleshoot a disconnected Amazon ECS agent?](#) (Come posso risolvere i problemi di un agente Amazon ECS disconnesso).

Risoluzione degli errori di Amazon ECS ResourceInitializationError

Di seguito sono riportati alcuni messaggi ResourceInitialization di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

Errori

- [L'attività non può richiamare l'autenticazione del registro da Amazon ECR. C'è un problema di connessione tra l'attività e Amazon ECR. Controlla la configurazione della rete delle tue attività.](#)
- [L'attività non può scaricare i file delle variabili di ambiente da Amazon S3. C'è un problema di connessione tra l'attività e Amazon S3. Controlla la configurazione della rete della tua attività.](#)
- [L'attività non può recuperare segreti da AWS Systems Manager Parameter Store. Controlla la connessione di rete tra l'attività e AWS Systems Manager.](#)
- [L'attività non può trarre segreti da AWS Secrets Manager. C'è un problema di connessione tra l'attività e Secrets Manager. Controlla la configurazione della rete delle tue attività.](#)
- [L'operazione non può recuperare il segreto da Secrets Manager. L'operazione non può recuperare il segreto con ARN secretARN " da Secrets Manager. Controlla se il segreto esiste nella regione specificata.](#)
- [comando pull non riuscito: impossibile recuperare i segreti o l'autenticazione del registro. Controlla la configurazione della rete dell'attività.](#)
- [L'attività non riesce a trovare il gruppo di CloudWatch log Amazon definito nella definizione dell'attività. C'è un problema di connessione tra l'attività e Amazon CloudWatch. Controlla la configurazione della tua rete.](#)
- [impossibile inizializzare il driver di registrazione](#)
- [non è riuscito a richiamare i comandi EFS utils per configurare i volumi EFS](#)

L'attività non può richiamare l'autenticazione del registro da Amazon ECR. C'è un problema di connessione tra l'attività e Amazon ECR. Controlla la configurazione della rete delle tue attività.

Questo errore indica che l'attività non può connettersi ad Amazon ECR.

Verifica la connessione tra l'attività e Amazon ECR. Per informazioni, consultare [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

L'attività non può scaricare i file delle variabili di ambiente da Amazon S3. C'è un problema di connessione tra l'attività e Amazon S3. Controlla la configurazione della rete della tua attività.

Questo errore si verifica quando l'attività non può scaricare il file di ambiente da Amazon S3.

Verifica la connessione tra l'attività e l'endpoint VPC Amazon S3. Per informazioni, consultare [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

L'attività non può recuperare segreti da AWS Systems Manager Parameter Store. Controlla la connessione di rete tra l'attività e AWS Systems Manager.

Questo errore si verifica quando l'attività non è in grado di estrarre l'immagine definita nella definizione dell'attività utilizzando le credenziali in Systems Manager.

Verificare la connessione tra l'attività e l'endpoint VPC di Systems Manager. Per informazioni, consultare [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

L'attività non può trarre segreti da AWS Secrets Manager. C'è un problema di connessione tra l'attività e Secrets Manager. Controlla la configurazione della rete delle tue attività.

Questo errore si verifica quando l'attività non è in grado di estrarre l'immagine definita nella definizione dell'attività utilizzando le credenziali in Secrets Manager.

L'errore indica che esiste un problema di connettività di rete tra l'endpoint VPC di Systems Manager e l'attività.

Per informazioni su come verificare la connettività tra l'attività e l'endpoint, vedere. [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#)

L'operazione non può recuperare il segreto da Secrets Manager. L'operazione non può recuperare il segreto con ARN **secretARN** " da Secrets Manager. Controlla se il segreto esiste nella regione specificata.

Questo errore si verifica quando l'attività non è in grado di estrarre l'immagine definita nella definizione dell'attività utilizzando le credenziali in Secrets Manager.

Questo problema è causato da uno dei seguenti motivi:

Causa dell'errore..	Esegui questa operazione...	
Problema di connettività di rete tra l'endpoint VPC di Secrets Manager e l'attività. Il problema è un problema di rete quando viene visualizzata una delle seguenti stringhe nel messaggio di errore: • digita tcp • componi udp • <ip>:<port>: timeout di I/O • net/http: timeout dell'handshake TLS • read: timeout della connessione • Client.Timeout superato in attesa degli header • net/http: richiesta annullata in attesa della connessione • segnale: ucciso • scadenza del contesto superata	Verifica la connettività tra l'attività e l'endpoint Secrets Manager. Per ulteriori informazioni, consulta Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS.	
Il ruolo di esecuzione dell'attività definito nella definizione dell'attività non dispone delle autorizzazioni per Secrets Manager.	Aggiungi le autorizzazioni richieste per Secrets Manager al ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta Autorizzazioni Secrets Manager o Systems Manager.	

Causa dell'errore..	Esegui questa operazione...	
L'ARN segreto non esiste	Verifica che l'ARN esista in Secrets Manager. Per informazioni sulla visualizzazione delle immagini, consulta Find secrets in Secrets Manager nella Secrets Manager Developer Guide.	

comando pull non riuscito: impossibile recuperare i segreti o l'autenticazione del registro Controlla la configurazione della rete dell'attività.

Questo errore si verifica quando l'attività non riesce a connettersi ad Amazon ECR, Systems Manager o Secrets Manager. Ciò è dovuto a un'errata configurazione della rete.

Per risolvere questo problema, verifica la connettività tra l'attività e Amazon ECR. È inoltre necessario verificare la connettività tra l'attività e il servizio che memorizza il segreto (Systems Manager o Secrets Manager). Per ulteriori informazioni, consulta [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

L'attività non riesce a trovare il gruppo di CloudWatch log Amazon definito nella definizione dell'attività. C'è un problema di connessione tra l'attività e Amazon CloudWatch. Controlla la configurazione della tua rete.

Questo errore si verifica quando l'attività non riesce a trovare il gruppo di CloudWatch log definito nella definizione dell'attività.

L'errore indica che esiste un problema di connettività di rete tra l'endpoint CloudWatch VPC e l'attività.

Per informazioni su come verificare la connettività tra l'attività e l'endpoint, consulta. [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#)

impossibile inizializzare il driver di registrazione

Questo errore si verifica quando l'attività non riesce a trovare il gruppo di CloudWatch log definito nella definizione dell'attività.

L'errore indica che il CloudWatch gruppo nella definizione dell'attività non esiste.

Usa i seguenti passaggi per trovare i mancanti CloudWatch.

1. Esegui il comando seguente per ottenere le informazioni sulla definizione dell'attività.

```
aws ecs describe-task-definition \
  --task-definition task-def-name
```

Guardate l'output di ogni contenitore e annotate il `awslogs-group` valore.

```
"logConfiguration": {
  "logDriver": "awslogs",
  "options": {
    "awslogs-group": "/ecs/example-group",
    "awslogs-create-group": "true",
    "awslogs-region": "us-east-1",
    "awslogs-stream-prefix": "ecs"
  },

```

2. Verifica che il gruppo esista. CloudWatch Per ulteriori informazioni, consulta [Working with log groups and log stream](#) nella Amazon CloudWatch Logs User Guide.

Il problema è che il gruppo specificato nella definizione dell'attività non è corretto o il gruppo di log non esiste.

3. Risolvi il problema.

Il problema è...	Esegui questa operazione...	
Il gruppo di log errato è specificato nella definizione dell'attività.	Aggiorna la definizione del task per includere la configurazione del gruppo di log nella definizione del container. Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition consulta il riferimento	

Il problema è...	Esegui questa operazione...	
	all'API di Amazon Elastic Container Service.	
Il gruppo di log non esiste in CloudWatch	Crea il gruppo di log. Per ulteriori informazioni, consulta Creare un gruppo di log in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.	

non è riuscito a richiamare i comandi EFS utils per configurare i volumi EFS

I seguenti problemi potrebbero impedirti di montare i volumi Amazon EFS sulle tue richieste:

- Il file system Amazon EFS non è configurato correttamente.
- L'attività non dispone delle autorizzazioni richieste.
- Esistono problemi relativi alle configurazioni di rete e VPC.

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Perché non posso montare i miei volumi Amazon EFS sulle mie AWS Fargate attività su AWS re:POST](#).

Risoluzione degli errori di Amazon ECS ResourceNotFoundException

Di seguito sono riportati alcuni messaggi `ResourceNotFoundException` di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

L'operazione non può recuperare il segreto con ARN ***secretARN*** " da. AWS Secrets Manager
Controlla se il segreto esiste nella regione specificata.

Questo errore si verifica quando l'attività non è in grado di recuperare il segreto da Secrets Manager. Ciò significa che il segreto specificato nella definizione dell'attività (e contenuto nel messaggio di errore) non esiste in Secrets Manager.

La regione è presente nel messaggio di errore.

Il recupero di dati segreti dalla AWS Secrets Manager regione *region*: secret: *secretARN*
 ResourceNotFoundException: Secrets Manager non riesce a trovare il segreto specificato.

Per informazioni sulla ricerca di un segreto, consulta [Trova segreti AWS Secrets Manager nella Guida per l'AWS Secrets Manager utente](#).

Utilizzate la tabella seguente per determinare e risolvere l'errore.

Problema	Azioni	
Il segreto si trova in una regione diversa dalla definizione dell'attività.	a. Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni, consulta Creare un AWS Secrets Manager segreto. b. Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	
La definizione dell'attività ha un ARN segreto errato. Il segreto corretto esiste in Secrets Manager.	Aggiorna la definizione dell'attività con il segreto corretto. Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	

Problema	Azioni	
Il segreto non esiste più.	a. Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni, consulta Creare un AWS Secrets Manager segreto. b. Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	

Risoluzione degli errori di Amazon ECS SpotInterruption

L'SpotInterruptionerrore ha diverse ragioni a seconda del tipo Fargate e del tipo di EC2 lancio.

Per verificare la presenza di un messaggio di errore relativo alle attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

Tipo di avvio di Fargate

L'SpotInterruptionerrore si verifica quando non c'è capacità Fargate Spot o quando Fargate riprende la capacità Spot.

È possibile eseguire le attività in più zone di disponibilità per consentire una maggiore capacità.

EC2 tipo di avvio

Questo errore si verifica quando non ci sono istanze Spot disponibili o EC2 ripristina la capacità dell'istanza Spot.

Puoi far funzionare le tue istanze in più zone di disponibilità per aumentare la capacità.

Risoluzione degli errori di Amazon ECS InternalError

Si applica a: Tipo di lancio Fargate

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, vedere [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

L'InternalError errore quando l'agente rileva un errore interno imprevisto, non correlato al runtime.

Questo errore si verifica solo se si utilizza la piattaforma versione 1.4 o una versione successiva.

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Come posso risolvere un task Amazon ECS che non è stato avviato in un cluster ECS](#) su re:post. AWS

Risoluzione degli errori di Amazon ECS OutOfMemoryError

Di seguito sono riportati alcuni messaggi OutOfMemoryError di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

contenitore interrotto a causa dell'utilizzo della memoria

Questo errore si verifica quando un contenitore esce a causa di processi al suo interno che consumano più memoria di quella allocata nella definizione dell'attività o a causa di vincoli relativi all'host o al sistema operativo.

Risoluzione degli errori di Amazon ECS ContainerRuntimeError

Di seguito sono riportati alcuni messaggi ContainerRuntimeError di errore e azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

ContainerRuntimeError

Questo errore si verifica quando l'agente riceve un errore non previsto da containerd per un'operazione specifica del runtime. Questo errore in genere è causato da un errore interno nell'agente o nel runtime di containerd.

Questo errore si verifica solo se si utilizza la versione 1.4.0 della piattaforma o versioni successive (Linux) o 1.0.0 o versioni successive (Windows).

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Why is my Amazon ECS task Stopped](#) on AWS re:post.

Risoluzione degli errori di Amazon ECS ContainerRuntimeTimeoutError

Di seguito sono riportati alcuni messaggi ContainerRuntimeTimeoutError di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

Impossibile passare alla fase di esecuzione; timeout scaduto dopo un'attesa di 1 minuto o errore di timeout Docker

Questo errore si verifica quando un container non è in grado di passare a uno stato RUNNING o STOPPED entro il periodo di timeout. Il motivo e il valore di timeout vengono forniti nel messaggio di errore.

Risoluzione degli errori di Amazon ECS CannotStartContainerError

Di seguito sono riportati alcuni messaggi CannotStartContainerError di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

impossibile ottenere lo stato del contenitore: **<reason>**

Questo errore si verifica quando un container non può essere avviato.

Se il contenitore tenta di superare la memoria specificata qui, il contenitore viene interrotto. Aumenta la memoria fornita al contenitore. Questo è il memory parametro nella definizione dell'attività. Per ulteriori informazioni, consulta [the section called "Memoria"](#).

Risoluzione degli errori di Amazon ECS CannotStopContainerError

Di seguito sono riportati alcuni messaggi CannotStopContainerError di errore e azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

CannotStopContainerError

Questo errore si verifica quando un container non può essere arrestato.

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Why is my Amazon ECS task Stopped](#) on AWS re:post.

Risoluzione degli errori di Amazon ECS CannotInspectContainerError

Di seguito sono riportati alcuni messaggi CannotInspectContainerError di errore e azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

CannotInspectContainerError

Questo errore si verifica quando l'agente del container non può descrivere il container tramite il runtime del container.

Quando si utilizza una versione della piattaforma 1.3 o precedente, l'agente Amazon ECS restituisce il motivo da Docker.

Quando si utilizza la versione della piattaforma 1.4.0 o successiva (Linux) 1.0.0 o successiva (Windows), l'agente Fargate restituisce il motivo da containerd.

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Why is my Amazon ECS task Stopped](#) on AWS re:post.

Risoluzione degli errori di Amazon ECS CannotCreateVolumeError

Di seguito sono riportati alcuni messaggi CannotCreateVolumeError di errore e le azioni che è possibile eseguire per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#).

CannotCreateVolumeError

Questo errore si verifica quando l'agente non è in grado di creare il montaggio del volume specificato nella definizione di attività.

Questo errore si verifica solo se si utilizza la versione 1.4.0 della piattaforma o versioni successive (Linux) o 1.0.0 o versioni successive (Windows).

Per informazioni su come eseguire il debug e risolvere questo problema, consulta [Why is my Amazon ECS task Stopped](#) on AWS re:post.

CannotPullContainer errori di attività in Amazon ECS

I seguenti errori indicano che l'attività non è stata avviata perché Amazon ECS non è in grado di recuperare l'immagine del contenitore specificata.

Note

La piattaforma Fargate versione 1.4 tronca i messaggi di errore troppo lunghi.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, consulta. [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#)

Errori

- [L'operazione non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre immagini dal registro.](#)
- [L'attività non può estrarre image-name " dal repository Amazon ECR ". repository URI C'è un problema di connessione tra l'attività e Amazon ECR. Controlla la configurazione della rete delle tue attività.](#)
- [L'operazione non può estrarre l'immagine. Controlla la configurazione della tua rete](#)
- [CannotPullContainerError: pull image manifest è stato riprovato 5 volte: errore di risoluzione del riferimento](#)
- [Errore API \(500\): Get https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: richiesta annullata in attesa della connessione](#)
- [Errori API](#)
- [write /var/lib/docker/tmp/GetImageBlob111111111: non è rimasto spazio sul dispositivo](#)
- [ERRORE: toomanyrequests: Troppe richieste o hai raggiunto il limite di pull rate.](#)
- [Risposta di errore dal demone: Get: net/http: richiesta annullata in attesa della url connessione](#)
- [ref pull è stato riprovato 1 volta/e: copia non riuscita: apertura fallita httpReaderSeeker: codice di stato imprevisto](#)

- [pull access negato](#)
- [comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido o dereferenziazione del puntatore nullo](#)
- [errore durante l'estrazione dell'immagine conf/errore durante l'estrazione della configurazione dell'immagine](#)
- [Contesto annullato](#)

L'operazione non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre immagini dal registro.

Questo errore indica che l'attività non può recuperare l'immagine specificata nella definizione dell'attività a causa di problemi di autorizzazione.

Per risolvere il problema:

1. Verifica che l'immagine esista in **repository**. Per informazioni sulla visualizzazione delle immagini, consulta [Visualizzazione dei dettagli delle immagini in Amazon ECR nella Amazon Elastic Container Registry User Guide](#).
2. Verifica di disporre delle **role-arn** autorizzazioni corrette per estrarre l'immagine.

Per informazioni su come aggiornare i ruoli, consulta [Aggiornare le autorizzazioni per un ruolo nella Guida](#) per l'AWS Identity and Access Management uso.

L'attività utilizza uno dei seguenti ruoli:

- Per le attività con il tipo di avvio Fargate, questo è il ruolo di esecuzione dell'attività. Per informazioni sulle autorizzazioni aggiuntive per Amazon ECR, [Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia](#)
- Per le attività con il tipo di EC2 avvio, questo è il ruolo dell'istanza del contenitore. Per informazioni sulle autorizzazioni aggiuntive per Amazon ECR, [Autorizzazioni Amazon ECR](#)

L'attività non può estrarre **image-name** " dal repository Amazon ECR ". **repository URI** C'è un problema di connessione tra l'attività e Amazon ECR. Controlla la configurazione della rete delle tue attività.

Questo errore indica che l'attività non può connettersi ad Amazon ECR. Verifica la connessione al **repository URI** repository.

Per informazioni su come verificare e risolvere il problema, consulta [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

L'operazione non può estrarre l'immagine. Controlla la configurazione della tua rete

Questo errore indica che l'attività non può connettersi ad Amazon ECR.

Per informazioni su come verificare e risolvere il problema, consulta [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#).

CannotPullContainerError: pull image manifest è stato riprovato 5 volte: errore di risoluzione del riferimento

Questo errore indica che l'attività non può estrarre l'immagine.

Per risolvere questo, puoi:

- Verifica che l'immagine specificata nella definizione dell'attività corrisponda all'immagine nel repository.
- Amazon ECS impone la stabilità della versione dell'immagine. Se l'immagine originale non è più disponibile, viene visualizzato questo errore. Il tag image fa parte dell'applicazione di questo comportamento. Cambia l'immagine nella definizione dell'attività da usare:latest come tag a una versione specifica. Per ulteriori informazioni, consulta [Risoluzione dell'immagine del contenitore](#).

Per informazioni su come verificare e risolvere il problema, consulta. [Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS](#)

Errore API (500): Get https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: richiesta annullata in attesa della connessione

Questo errore indica che la connessione è scaduta perché non esiste un percorso verso Internet.

Per risolvere questo problema, puoi procedere in questi modi:

- Per le attività in sottoreti pubbliche, puoi specificare ENABLED (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).
- Per le attività in sottoreti private, puoi specificare DISABLED (DISABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo e quindi configurare un gateway NAT nel VPC per instradare le richieste a Internet. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC.

Errori API

Questo errore indica che c'è un problema di connessione con l'endpoint Amazon ECR.

Per informazioni su come risolvere questo problema, consulta [How can I resolve the Amazon ECR error "CannotPullContainerError: API error» in Amazon ECS sul Supporto sito Web.](#)

write /var/lib/docker/tmp/**GetImageBlob111111111**: non è rimasto spazio sul dispositivo

Questo errore indica che lo spazio su disco è insufficiente.

Per risolvere questo problema, libera spazio sul disco.

Se utilizzi l'AMI ottimizzata per Amazon ECS, puoi utilizzare il seguente comando per recuperare i 20 file più grandi sul tuo file system:

```
du -Sh / | sort -rh | head -20
```

Output di esempio:

```
5.7G    /var/lib/docker/
containers/50501b5f4cbf90b406e0ca60bf4e6d4ec8f773a6c1d2b451ed8e0195418ad0d2
1.2G    /var/log/ecs
594M    /var/lib/docker/devicemapper/mnt/
c8e3010e36ce4c089bf286a623699f5233097ca126ebd5a700af023a5127633d/rootfs/data/logs
...
```

In alcuni casi, il volume root potrebbe essere compilato da un contenitore in esecuzione. Se il container sta utilizzando il driver di log `json-file` predefinito senza un limite `max-size`, è probabile che il file di log sia responsabile della maggior parte dello spazio occupato. Puoi utilizzare il comando `docker ps` per verificare quale container sta utilizzando lo spazio mappando il nome della directory dall'output sopra l'ID del container. Ad esempio:

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
50501b5f4cbf	amazon/amazon-ecs-agent:latest	"/agent"	4 days ago
Up 4 days		ecs-agent	

Di default, quando utilizzi il driver di log `json-file`, Docker acquisisce l'output standard (e l'errore standard) di tutti i container e li scrive in file utilizzando il formato JSON. Puoi impostare `max-`

size come opzione del driver di log per evitare che il file di log occupi troppo spazio. Per ulteriori informazioni, consulta [JSON File logging driver](#) nella documentazione di Docker.

La seguente è uno snippet di definizione di container che mostra come usare questa opzione:

```
{
  "log-driver": "json-file",
  "log-opts": {
    "max-size": "256m"
  }
}
```

Un'alternativa, se i log del contenitore occupano troppo spazio su disco, consiste nell'utilizzare il driver di registro. awslogs Il driver di registro invia i awslogs log a CloudWatch, il che libera lo spazio su disco che verrebbe altrimenti utilizzato per i registri del contenitore sull'istanza del contenitore. Per ulteriori informazioni, consulta [Invia i log di Amazon ECS a CloudWatch](#).

ERRORE: toomanyrequests: Troppe richieste o hai raggiunto il limite di pull rate.

Questo errore indica che esiste una limitazione della velocità di Docker Hub.

Se ricevi uno degli errori seguenti, è probabile che si stiano per raggiungere i limiti della frequenza di Docker Hub:

Per ulteriori informazioni sui limiti di frequenza di Docker Hub, consulta [Informazioni sulla limitazione della frequenza di Docker Hub](#).

[Se hai aumentato il limite di velocità di Docker Hub e devi autenticare i file Docker pull per le istanze di container, vedi Autenticazione del registro privato per le istanze di container.](#)

Risposta di errore dal demone: Get: net/http: richiesta annullata in attesa della **url** connessione

Questo errore indica che la connessione è scaduta perché non esiste un percorso verso Internet.

Per risolvere questo problema, puoi procedere in questi modi:

- Per le attività in sottoreti pubbliche, puoi specificare ENABLED (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).
- Per le attività in sottoreti private, puoi specificare DISABLED (DISABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo e quindi configurare un gateway NAT nel VPC

per instradare le richieste a Internet. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC.

ref pull è stato riprovato 1 volta/e: copia non riuscita: apertura fallita httpReaderSeeker: codice di stato imprevisto

Questo errore indica che si è verificato un errore durante la copia di un'immagine.

Per risolvere questo problema, consulta uno dei seguenti articoli:

- Per i processi Fargate, consulta [Come posso risolvere l'errore "cannotpullcontainererror" per i miei processi Amazon ECS su Fargate.](#)
- Per altri processi, consulta [Come posso risolvere l'errore "cannotpullcontainererror" per i miei processi Amazon ECS.](#)

pull access negato

Questo errore indica che non è possibile accedere all'immagine.

Per risolvere questo problema, potrebbe essere necessario autenticare il client Docker con Amazon ECR. Per ulteriori informazioni, consulta [Autenticazione del registro privato](#) nella Amazon ECR User Guide.

comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido o dereferenziazione del puntatore nullo

Questo errore indica che non è possibile accedere all'immagine a causa di un indirizzo di memoria non valido o di una dereferenziazione del puntatore pari a zero.

Per risolvere il problema:

- Verifica di disporre delle regole del gruppo di sicurezza per accedere ad Amazon S3.
- Quando si utilizzano gli endpoint del gateway, è necessario aggiungere una route nella tabella delle rotte per accedere all'endpoint.

errore durante l'estrazione dell'immagine conf/errore durante l'estrazione della configurazione dell'immagine

Questo errore indica che è stato raggiunto un limite di velocità o che c'è un errore di rete:

Per risolvere questo problema, consulta [Come posso risolvere l'errore "CannotPullContainerError" nel mio Amazon ECS EC2 Launch Type Task](#).

Contesto annullato

Questo errore indica che il contesto è stato annullato.

La causa comune di questo errore è che il VPC utilizzato dal processo che non ha un percorso per estrarre l'immagine del container da Amazon ECR.

Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS

Ci sono momenti in cui un'attività si interrompe a causa di un problema di connettività di rete. Potrebbe trattarsi di un problema intermittente, ma molto probabilmente è causato dal fatto che l'attività non riesce a connettersi a un endpoint.

Verifica della connettività dell'attività

È possibile utilizzare `AWSSupport-TroubleshootECSTaskFailedToStart` runbook per testare la connettività delle attività. Quando si utilizza il runbook, sono necessarie le seguenti informazioni sulle risorse:

- L'ID dell'attività

Utilizza l'ID dell'ultima operazione non riuscita.

- Il cluster in cui si trovava l'attività

Per informazioni su come usare il runbook, consulta [AWSSupport-TroubleshootECSTaskFailedToStart](#) l'AWS Systems Manager Automation runbook reference.

Il runbook analizza il task. È possibile visualizzare i risultati nella sezione Output per i seguenti problemi che possono impedire l'avvio di un'attività:

- Connettività di rete al registro dei contenitori configurato
- Connettività endpoint VPC
- Configurazione delle regole del gruppo di sicurezza

Risoluzione dei problemi relativi agli endpoint VPC

Quando il risultato del `AWSSupport-TroubleshootECSTaskFailedToStart` runbook indica il problema dell'endpoint VPC, controlla la seguente configurazione:

- Il VPC in cui si crea l'endpoint deve utilizzare il DNS privato.
- Assicurati di avere un AWS PrivateLink endpoint per il servizio a cui l'attività non può connettersi nello stesso VPC dell'attività. Per ulteriori informazioni, consulta una delle seguenti pagine:

Servizio	Informazioni sugli endpoint VPC per il servizio	
Amazon ECR	Endpoint VPC con interfaccia Amazon ECR (AWS PrivateLink)	
Systems Manager	Migliora la sicurezza delle EC2 istanze utilizzando gli endpoint VPC per Systems Manager	
Secrets Manager	Utilizzo di un AWS Secrets Manager endpoint VPC	
CloudWatch	CloudWatch Endpoint VPC	
Amazon S3	AWS PrivateLink per Amazon S3	

- Configura una regola in uscita per la sottorete delle attività che consente l'HTTPS sul traffico DNS (TCP) sulla porta 443. Per ulteriori informazioni, consulta [Configura le regole dei gruppi di sicurezza](#) nella Amazon Elastic Compute Cloud User Guide.
- Se utilizzi un server di dominio con nomi personalizzati, conferma le impostazioni della query DNS. La query deve avere accesso in uscita sulla porta 53 e utilizzare i protocolli UDP e TCP. Inoltre, deve avere accesso HTTPS sulla porta 443. Per ulteriori informazioni, consulta [Configura le regole dei gruppi di sicurezza](#) nella Amazon Elastic Compute Cloud User Guide.
- Se la sottorete dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.

- Una regola in entrata che consente il traffico TCP sulla porta 443.

Per informazioni su come configurare le regole, consulta [Controllare il traffico verso le sottoreti utilizzando la rete ACLs](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Risoluzione dei problemi di rete

Quando il risultato del `AWSSupport-TroubleshootECSTaskFailedToStart` runbook indica un problema di rete, controlla la seguente configurazione:

Attività che utilizzano la modalità di rete `awsvpc` in una sottorete pubblica

Esegui la seguente configurazione in base al runbook:

- Per le attività in sottoreti pubbliche, puoi specificare `ENABLED` (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).
- È necessario un gateway per gestire il traffico Internet. La tabella di routing per la sottorete delle attività deve avere un percorso per il traffico verso il gateway.

Per ulteriori informazioni, consulta [Aggiungere e rimuovere percorsi da una tabella di rotte](#) nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT
Internet Gateway	0.0.0.0/0	ID del gateway Internet

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consente il traffico sulle porte 1024-65535.
 - Una regola in entrata che consente il traffico TCP sulla porta 443.

Per informazioni su come configurare le regole, consulta [Controllare il traffico verso le sottoreti utilizzando la rete ACLs](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Attività che utilizzano la modalità di rete awsvpc in una sottorete privata

Esegui la seguente configurazione in base al runbook:

- Scegli DISABLED per assegnare automaticamente l'IP pubblico all'avvio dell'attività.
- Configura un gateway NAT nel tuo VPC per indirizzare le richieste a Internet. Per ulteriori informazioni, consulta [NAT Gateways](#) nella Amazon Virtual Private Cloud User Guide.
- La tabella di routing per la sottorete delle attività deve avere un percorso per il traffico verso il gateway NAT.

Per ulteriori informazioni, consulta [Aggiungere e rimuovere percorsi da una tabella di rotte](#) nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consente il traffico sulle porte 1024-65535.
 - Una regola in entrata che consente il traffico TCP sulla porta 443.

Per informazioni su come configurare le regole, consulta [Controllare il traffico verso le sottoreti utilizzando la rete ACLs](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Attività che non utilizzano la modalità di rete awsvpc in una sottorete pubblica

Esegui la seguente configurazione in base al runbook:

- Scegli Attiva per l'assegnazione automatica dell'IP in Networking for Amazon EC2 Instances quando crei il cluster.

Questa opzione assegna un indirizzo IP pubblico all'interfaccia di rete primaria dell'istanza.

- È necessario un gateway per gestire il traffico Internet. La tabella di routing per la sottorete dell'istanza deve avere un percorso per il traffico verso il gateway.

Per ulteriori informazioni, consulta [Aggiungere e rimuovere percorsi da una tabella di rotte](#) nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT
Internet Gateway	0.0.0.0/0	ID del gateway Internet

- Se la sottorete dell'istanza dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consente il traffico sulle porte 1024-65535.
 - Una regola in entrata che consente il traffico TCP sulla porta 443.

Per informazioni su come configurare le regole, consulta [Controllare il traffico verso le sottoreti utilizzando la rete ACLs](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Attività che non utilizzano la modalità di rete awsvpc in una sottorete privata

Esegui la seguente configurazione in base al runbook:

- Scegli Disattiva per l'assegnazione automatica dell'IP in Networking for Amazon EC2 Instances quando crei il cluster.
- Configura un gateway NAT nel tuo VPC per indirizzare le richieste a Internet. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon VPC.
- La tabella di routing per la sottorete dell'istanza deve avere un percorso per il traffico verso il gateway NAT.

Per ulteriori informazioni, consulta [Aggiungere e rimuovere percorsi da una tabella di rotte](#) nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consente il traffico sulle porte 1024-65535.
 - Una regola in entrata che consente il traffico TCP sulla porta 443.

Per informazioni su come configurare le regole, consulta [Controllare il traffico verso le sottoreti utilizzando la rete ACLs](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Visualizzazione delle richieste di ruolo IAM per le attività di Amazon ECS

Quando utilizzi un provider per le credenziali dell'attività in un ruolo IAM, le richieste del provider vengono salvate in un registro di controllo. Il log di audit eredita le stesse impostazioni di rotazione del log dell'agente del container. È possibile impostare le variabili di configurazione dell'agente del container `ECS_LOG_ROLLOVER_TYPE`, `ECS_LOG_MAX_FILE_SIZE_MB`, `ECS_LOG_MAX_ROLL_COUNT` in modo da influire sul comportamento del log di audit. Per ulteriori informazioni, consulta [Parametri di configurazione del registro dell'agente container Amazon ECS](#).

Per l'agente del container versione 1.36.0 e successive, il log di audit si trova in `/var/log/ecs/audit.log`. Quando il log viene ruotato, viene aggiunto un timestamp nel formato `YYYY-MM-DD-HH` alla fine del nome del file di log.

Per l'agente del container versione 1.35.0 e precedenti, il log di audit si trova in `/var/log/ecs/audit.log.YYYY-MM-DD-HH`.

Il formato delle voci di log è il seguente:

- Timestamp
- Codice di risposta HTTP
- Indirizzo IP e numero di porta dell'origine della richiesta
- URI relativo del provider di credenziali
- L'agente utente che ha effettuato la richiesta
- L'ARN dell'attività a cui appartiene il container richiedente
- Il nome API `GetCredentials` e il numero di versione
- Il nome del cluster Amazon ECS su cui è registrata l'istanza di container
- L'ARN dell'istanza di container

Puoi utilizzare il comando seguente per visualizzare i file di log.

```
cat /var/log/ecs/audit.log.2016-07-13-16
```

Output:

```
2016-07-13T16:11:53Z 200 172.17.0.5:52444 "/v1/credentials" "python-requests/2.7.0  
CPython/2.7.6 Linux/4.4.14-24.50.amzn1.x86_64" TASK_ARN GetCredentials  
1 CLUSTER_NAME CONTAINER_INSTANCE_ARN
```

Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS

Se stai risolvendo un problema relativo a un servizio, il primo luogo in cui cercare le informazioni di diagnostica è il log di eventi dei servizi. Puoi visualizzare gli eventi di servizio utilizzando l'DescribeServicesAPI AWS CLI, o utilizzando il AWS Management Console.

Quando si visualizzano i messaggi degli eventi di servizio utilizzando l'API Amazon ECS, vengono restituiti solo gli eventi dell'utilità di pianificazione del servizio. Questi includono il posizionamento dei processi e gli eventi di integrità delle istanze più recenti. Tuttavia, la console Amazon ECS visualizza gli eventi di servizio dalle seguenti origini.

- Posizionamento dei processi ed eventi di integrità delle istanze dal pianificatore di servizi Amazon ECS. Questi eventi hanno un prefisso di servizio (**service-name**). Per garantire che questa visualizzazione degli eventi sia utile, mostriamo solo gli eventi 100 più recenti. I messaggi di evento duplicati vengono omessi finché la causa non viene risolta o non sono trascorse sei ore. Se la causa non viene risolta entro sei ore, si riceve un altro messaggio relativo a un evento di servizio relativo a tale causa.
- Eventi di scalabilità automatica dei servizi Questi eventi hanno il prefisso Message e si verificano solo quando un servizio è configurato con una policy di scaling Application Auto Scaling.

Utilizza la procedura seguente per visualizzare i messaggi relativi a eventi del servizio corrente.

Console

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.

4. Scegli il servizio da ispezionare.
5. Scegli Deployments and events (Implementazioni ed eventi), in Events (Eventi), quindi visualizza i messaggi.

AWS CLI

Utilizza il comando [describe-services](#) per visualizzare i messaggi di evento del servizio per un servizio specificato.

L' AWS CLI esempio seguente descrive il *service-name* servizio nel *default* cluster, che fornirà i messaggi relativi agli eventi di servizio più recenti.

```
aws ecs describe-services \
  --cluster default \
  --services service-name \
  --region us-west-2
```

Messaggi sugli eventi del servizio Amazon ECS

Di seguito sono riportati alcuni esempi di messaggi di evento dei servizi che potresti visualizzare nella console Amazon ECS:

service (*service-name*) ha raggiunto uno stato stazionario.

Il service scheduler invia un evento di service (*service-name*) has reached a steady state . servizio quando il servizio è integro e ha il numero di attività desiderato, raggiungendo così uno stato stazionario.

Il pianificatore del servizio segnala periodicamente lo stato, quindi potresti ricevere questo messaggio più volte.

service (*service-name*) non è riuscito a inserire un'attività perché nessuna istanza del contenitore soddisfaceva tutti i requisiti.

Il service scheduler invia questo messaggio di evento quando non riesce a trovare le risorse disponibili per aggiungere un'altra attività. Le possibili cause sono:

Utilizza i provider di capacità per ridimensionare automaticamente le EC2 istanze. Per ulteriori informazioni, consulta [Provider di capacità Amazon ECS per il tipo di EC2 lancio](#).

Se intendevi utilizzare un provider di capacità, assicurati di adottare una strategia per provider di capacità o di avere una strategia predefinita per i provider di capacità associata al cluster e di non prendere in considerazione il tipo di lancio e la strategia del provider di capacità come input

Nessuna istanza di container trovata nel cluster

Se non è registrata alcuna istanza di container nel cluster in cui si tenta di eseguire un'attività, viene visualizzato questo errore. È opportuno aggiungere istanze di container al cluster. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Porte insufficienti

Se la tua attività utilizza una mappatura fissa per le porte host (ad esempio, utilizza la porta 80 sull'host per un server Web), devi disporre almeno di un'istanza di container per ogni attività, poiché solo un container può utilizzare una sola porta host alla volta. È consigliabile aggiungere istanze di container al tuo cluster o ridurre il numero di attività desiderate.

Troppe porte registrate

L'istanza di contenitore più vicina corrispondente per il posizionamento delle attività non può superare il limite massimo consentito di porte riservate di 100 porte host per istanza di contenitore. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione dell'attività di questa attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza del contenitore scelta. Il messaggio dell'evento del servizio avrebbe l'ID dell'istanza di container scelto come parte del messaggio riportato di seguito.

```
The closest matching container-instance is already using a port required by your task.
```

Memoria insufficiente

Se nella definizione di attività sono specificati 1.000 MiB di memoria e le istanze di container nel cluster hanno ognuna 1.024 MiB di memoria, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire la memoria nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

 Note

Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

CPU insufficiente

Un'istanza di container dispone di 1.024 unità CPU per ogni core CPU. Se nella definizione di attività sono specificate 1.000 unità CPU e le istanze di container nel cluster hanno ognuna 1.024 unità CPU, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire le unità CPU nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

Punti di collegamento insufficienti per le interfacce di rete elastiche

Le attività che utilizzano la modalità di rete awsipc ricevono ognuna una propria interfaccia di rete elastica (ENI), che è collegata all'istanza di container che la ospita. EC2 Le istanze Amazon hanno un limite al numero di istanze ENIs che possono essere collegate e non ci sono istanze container nel cluster con capacità ENI disponibile.

Il limite ENI per singole istanze di container dipende dalle condizioni seguenti:

- Se non hai fornito il consenso esplicito all'impostazione dell'account awsipcTrunking, il limite ENI per ogni istanza di container dipende dal tipo di istanza. Per ulteriori informazioni, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Se hai attivato l'impostazione dell'awsipcTrunkingaccount ma non hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo l'attivazione, il limite ENI per ogni istanza di contenitore è ancora al valore predefinito. Per ulteriori informazioni, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Se hai attivato l'impostazione dell'awsipcTrunkingaccount e hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo l'attivazione, ne sono disponibili altre ENIs . Per ulteriori informazioni, consulta [Istanze supportate per interfacce di rete di container Amazon ECS potenziate](#).

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account awsipcTrunking, consulta [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#).

Puoi aggiungere istanze di container al tuo cluster per fornire più adattatori di rete disponibili.

Attributo richiesto mancante nell'istanza di container

Alcuni parametri di definizione di attività richiedono che nell'istanza di container sia installata una specifica versione dell'API remota Docker. Altri, come le opzioni relative ai driver di log, richiedono alle istanze di container di registrare tali driver con la variabile di configurazione dell'agente `ECS_AVAILABLE_LOGGING_DRIVERS`. Se la definizione dell'attività contiene un parametro che richiede uno specifico attributo di istanza del contenitore e non disponi di istanze di contenitore disponibili in grado di soddisfare questo requisito, l'attività non può essere inserita.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di `awsvpc` rete e il tipo di EC2 avvio. Nel cluster specificato non è registrata un'istanza di contenitore nella stessa sottorete specificata al `awsvpcConfiguration` momento della creazione del servizio.

È possibile utilizzare il `AWSSupport-TroubleshootECSContainerInstance` runbook per risolvere i problemi. Il runbook verifica se i dati utente dell'istanza contengono le informazioni corrette sul cluster, se il profilo dell'istanza contiene le autorizzazioni richieste e i problemi di configurazione della rete. Per ulteriori informazioni, consulta [AWSSupport-TroubleshootECSContainerInstance](#) nella Guida per l'utente di riferimento del runbook di automazione AWS Systems Manager .

Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#) e [Configurazione dell'agente del container Amazon ECS](#).

service (***service-name***) non è riuscito a inserire un'attività perché nessuna istanza del contenitore soddisfaceva tutti i requisiti. L'istanza contenitore più vicina al contenitore non ***container-instance-id*** dispone di unità CPU sufficienti.

L'istanza del contenitore più vicina per il posizionamento dell'attività non contiene unità CPU sufficienti per soddisfare i requisiti nella definizione dell'attività. Esaminare i requisiti di CPU in entrambi i parametri dimensioni attività e definizione del container della definizione di attività.

service (***service-name***) non è riuscito a inserire un'attività perché nessuna istanza del contenitore soddisfaceva tutti i requisiti. L'istanza del contenitore più vicina corrispondente ***container-instance-id*** ha rilevato l'errore «AGENT».

L'agente del container Amazon ECS sull'istanza di container con la corrispondenza maggiore per il posizionamento del processo viene disconnesso. Se riesci a connetterti all'istanza di container

tramite SSH, puoi esaminare i log dell'agente; per ulteriori informazioni, consulta [Parametri di configurazione del registro dell'agente container Amazon ECS](#). Devi anche verificare che l'agente sia in esecuzione sull'istanza. Se stai utilizzando l'AMI ottimizzata per Amazon ECS, puoi provare ad arrestare e riavviare l'agente con il comando seguente.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS e l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS

```
sudo stop ecs && sudo start ecs
```

service (**service-name**) (instance**instance-id**) non è integro in (elb**lb-name**) a causa di (motivo: l'istanza non ha superato almeno il UnhealthyThreshold numero di controlli sanitari consecutivi).

Questo servizio è registrato con un load balancer i cui controlli dello stato hanno esito negativo. Per ulteriori informazioni, consulta [Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS](#).

service (**service-name**) non è in grado di avviare le attività in modo coerente e corretto.

Questo servizio contiene attività che non sono state avviate dopo vari tentativi consecutivi. A questo punto, il pianificatore del servizio inizia ad aumentare in modo incrementale il tempo tra i nuovi tentativi. È consigliabile risolvere il motivo per cui le attività non vengono avviate. Per ulteriori informazioni, consulta [Logica di accelerazione del servizio Amazon ECS](#).

Una volta aggiornato il servizio, ad esempio con una definizione aggiornata dell'attività, il pianificatore del servizio riprende il normale funzionamento.

le operazioni di service (**service-name**) vengono limitate. Riproverò più tardi.

Questo servizio non è in grado di avviare più processi a causa dei limiti di limitazione delle API. Una volta che il pianificatore di servizi è in grado di avviare più processi, riprenderà l'esecuzione.

Per richiedere un aumento della quota, apri la pagina [Supporto AWS Center](#), accedi se necessario e scegli Crea caso. Selezionare Service limit increase (Aumento limiti del servizio). Compilare e inviare il modulo.

service (***service-name***) non è stato in grado di interrompere o avviare le attività durante una distribuzione a causa della configurazione della distribuzione del servizio. Aggiorna il valore minimumHealthyPercent o MaximumPercent e riprova.

Questo servizio non è riuscito ad arrestare o avviare i processi durante un'implementazione a causa della configurazione dell'implementazione. La configurazione di distribuzione è costituita dai maximumPercent valori minimumHealthyPercent e, che vengono definiti al momento della creazione del servizio. Questi valori possono essere aggiornati anche su un servizio esistente.

minimumHealthyPercentRappresenta il limite inferiore al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo. È una percentuale del numero desiderato di attività per il servizio. Questo valore viene arrotondato per eccesso. Ad esempio, se la percentuale minima di integrità è 50 e il numero di attività desiderato è quattro, lo scheduler può interrompere due attività esistenti prima di iniziarne due nuove. Allo stesso modo, se la percentuale di integrità minima è 75% e il numero di processi desiderato è due, il pianificatore non può interrompere alcun processo a causa del valore risultante che è anche due.

maximumPercentRappresenta il limite massimo al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo. È una percentuale del numero desiderato di attività per un servizio. Questo valore viene arrotondato per difetto. Ad esempio, se la percentuale massima è 200 e il numero di attività desiderato è quattro, lo scheduler può iniziare quattro nuove attività prima di interrompere quattro attività esistenti. Allo stesso modo, se la percentuale di integrità massima è 125 e il numero di processi desiderato è tre, il pianificatore non può interrompere alcun processo a causa del valore risultante che è esso stesso tre.

Quando si imposta una percentuale di integrità minima o massima, è necessario assicurarsi che il pianificatore possa arrestare o avviare almeno un processo quando viene attivata un'implementazione.

service (**service-name**) non è riuscito a inserire un'attività. Motivo: hai raggiunto il numero limite di attività che puoi eseguire in un dato momento

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta [Quote del servizio](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

service (**service-name**) non è riuscito a inserire un'attività. Motivo: errore interno.

Di seguito è riportata la possibile causa di questo errore:

Il servizio non è in grado di avviare un processo a causa della presenza di una sottorete in una zona di disponibilità non supportata.

Per informazioni sulle regioni Fargate e le zone di disponibilità supportate, consulta [the section called "AWS Regioni di Fargate"](#).

Per informazioni su come visualizzare la zona di disponibilità della sottorete, consulta [Visualizzazione della sottorete](#) nella Guida per l'utente di Amazon VPC.

service (**service-name**) non è riuscito a effettuare un'operazione. Motivo: la configurazione della CPU richiesta è al di sopra del limite.

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta [Quote del servizio](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

service (**service-name**) non è stato in grado di effettuare un'operazione. Motivo: la configurazione di memoria richiesta è al di sopra del limite.

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta [Quote del servizio](#). Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

service (**service-name**) non è stato in grado di effettuare un'operazione. Motivo: hai raggiunto il limite del numero di v CPUs che puoi eseguire contemporaneamente

AWS Fargate sta passando da quote basate sul conteggio delle attività a quote basate su vCPU.

Puoi richiedere un aumento della quota per la quota basata su vCPU Fargate. Per ulteriori informazioni, consulta [Quote del servizio](#). Per richiedere un aumento delle quote Fargate, consulta [Richiesta di aumento delle quote](#) nella Guida per l'utente di Service Quotas.

service (***service-name***) non è riuscito a raggiungere lo stato stazionario perché task set () non è riuscito a scalare. ***taskSet-ID*** Motivo: il numero di attività protette è superiore al numero di attività desiderato

Il servizio ha un numero di attività più protetto rispetto al numero di attività desiderato. Puoi effettuare una delle seguenti operazioni:

- Attendi la scadenza della protezione per le attività correnti, consentendone così la cessazione.
- Determina quali attività possono essere interrotte e utilizza l'UpdateTaskProtectionAPI con l'protectionEnabledopzione impostata su false per annullare la protezione per queste attività.
- Aumenta il numero di attività desiderate del servizio portandolo a un numero maggiore del numero di attività protette.

service (***service-name***) non è riuscito a raggiungere lo stato stazionario. Motivo: non è stata trovata alcuna istanza di container nel provider di capacità.

Il service scheduler invia questo messaggio di evento quando non riesce a trovare le risorse disponibili per aggiungere un'altra attività. Le possibili cause sono:

Non esiste alcun provider di capacità associato al cluster

`describe-services` Da utilizzare per verificare che al cluster sia associato un provider di capacità. È possibile aggiornare la strategia del provider di capacità per il servizio.

Verifica che ci sia capacità disponibile nel provider di capacità. Nel caso del tipo di EC2 avvio, assicurati che le istanze del contenitore soddisfino i requisiti di definizione delle attività.

Nessuna istanza di container trovata nel cluster

Se nessuna istanza del contenitore è registrata nel cluster in cui si tenta di eseguire un'attività, viene visualizzato questo errore. È opportuno aggiungere istanze di container al cluster. Per ulteriori informazioni, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#).

Porte insufficienti

Se l'attività utilizza la mappatura fissa delle porte dell'host (ad esempio, l'attività utilizza la porta 80 sull'host per un server Web), è necessario disporre di almeno un'istanza di contenitore per attività. Solo un contenitore può utilizzare una singola porta host alla volta. È consigliabile aggiungere istanze di container al tuo cluster o ridurre il numero di attività desiderate.

Troppe porte registrate

L'istanza di container più vicina per il posizionamento delle attività non può superare il limite massimo consentito di porte riservate di 100 porte host per istanza di contenitore. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione dell'attività di questa attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza del contenitore scelta. Il messaggio dell'evento del servizio avrebbe l'ID dell'istanza di container scelto come parte del messaggio riportato di seguito.

```
The closest matching container-instance is already using a port required by your task.
```

Memoria insufficiente

Se nella definizione di attività sono specificati 1.000 MiB di memoria e le istanze di container nel cluster hanno ognuna 1.024 MiB di memoria, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire la memoria nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

Note

Per ottimizzare l'utilizzo delle risorse, assegnando al processo quanta più memoria possibile per un determinato tipo di istanza, consulta [Riservare la memoria delle istanze del contenitore Amazon ECS Linux](#).

Punti di collegamento insufficienti per le interfacce di rete elastiche

Le attività che utilizzano la modalità di rete aws-vpc ricevono ognuna una propria interfaccia di rete elastica (ENI), che è collegata all'istanza di container che la ospita. EC2 Le istanze Amazon

hanno un limite al numero di istanze ENIs che possono essere collegate e nel cluster non ci sono istanze container con capacità ENI disponibile.

Il limite ENI per singole istanze di container dipende dalle condizioni seguenti:

- Se non hai fornito il consenso esplicito all'impostazione dell'account `awsvpcTrunking`, il limite ENI per ogni istanza di container dipende dal tipo di istanza. Per ulteriori informazioni, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Se hai attivato l'impostazione dell'account `awsvpcTrunking` ma non hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo l'attivazione, il limite ENI per ogni istanza di contenitore è ancora al valore predefinito. Per ulteriori informazioni, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.
- Se hai attivato l'impostazione dell'account `awsvpcTrunking` e hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo l'attivazione, ne sono disponibili altre ENIs. Per ulteriori informazioni, consulta [Istanze supportate per interfacce di rete di container Amazon ECS potenziate](#).

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsvpcTrunking`, consulta [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#).

Puoi aggiungere istanze di container al tuo cluster per fornire più adattatori di rete disponibili.

Attributo richiesto mancante nell'istanza di container

Alcuni parametri di definizione di attività richiedono che nell'istanza di container sia installata una specifica versione dell'API remota Docker. Altri, come le opzioni relative ai driver di log, richiedono alle istanze di container di registrare tali driver con la variabile di configurazione dell'agente `ECS_AVAILABLE_LOGGING_DRIVERS`. Se la definizione dell'attività contiene un parametro che richiede uno specifico attributo di istanza del contenitore e non sono disponibili istanze di contenitore in grado di soddisfare questo requisito, l'attività non può essere inserita.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di `awsvpc` rete e il tipo di EC2 avvio e nel cluster specificato non è registrata un'istanza di contenitore nella stessa sottorete specificata al `awsvpcConfiguration` momento della creazione del servizio.

È possibile utilizzare il `AWSSupport-TroubleshootECSContainerInstance` runbook per la risoluzione dei problemi. Il runbook verifica se i dati utente dell'istanza contengono le informazioni corrette sul cluster, se il profilo dell'istanza contiene le autorizzazioni richieste e i problemi di configurazione della rete. Per ulteriori informazioni, consulta [AWSSupport-](#)

[TroubleshootECSContainerInstance](#) nella Guida per l'utente di riferimento del runbook di automazione AWS Systems Manager .

Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta [Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate](#) e [Configurazione dell'agente del container Amazon ECS](#).

service (**service-name**) non è riuscito a inserire un'attività. Motivo: la capacità non è al momento disponibile. Riprova più tardi o in un'altra zona di disponibilità.

Al momento non è disponibile alcuna capacità su cui eseguire il servizio.

Puoi effettuare una delle seguenti operazioni:

- Attendi che la capacità o le istanze EC2 container di Fargate diventino disponibili.
- Riavvia il servizio e specifica sottoreti aggiuntive.

distribuzione del servizio (**service-name**) non riuscita: le attività non sono state avviate.

Impossibile avviare le attività del servizio.

Per informazioni su come eseguire il debug delle attività interrotte, vedere. [Messaggi di errore delle attività interrotte da Amazon ECS](#)

service (**service-name**) Timeout in attesa dell'avvio di Amazon ECS Agent. Controlla i log in/var/log/ecs/ecs-agent.log».

L'agente del container Amazon ECS sull'istanza di container con la corrispondenza maggiore per il posizionamento del processo viene disconnesso. Se riesci a connetterti all'istanza del contenitore con SSH, puoi esaminare i log dell'agente. Per ulteriori informazioni, consulta [Parametri di configurazione del registro dell'agente container Amazon ECS](#). Devi anche verificare che l'agente sia in esecuzione sull'istanza. Se stai utilizzando l'AMI ottimizzata per Amazon ECS, puoi provare ad arrestare e riavviare l'agente con il comando seguente.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS

```
sudo stop ecs && sudo start ecs
```

service (*service-name*) task set (*taskSet-ID*) non è integro in target-group (*targetGroup-ARN*) a causa di. **TARGET GROUP IS NOT FOUND**

L'attività impostata per il servizio non supera i controlli di integrità perché il gruppo target non è stato trovato. È necessario eliminare e ricreare il servizio. Non eliminare alcun gruppo target Elastic Load Balancing a meno che il servizio Amazon ECS corrispondente non sia già stato eliminato.

service (*service-name*) task set (*taskSet-ID*) non è integro in target-group (*targetGroup-ARN*) a causa di. **TARGET IS NOT FOUND**

L'attività impostata per il servizio non supera i controlli di integrità perché la destinazione non è stata trovata.

Messaggi sugli eventi del servizio di ribilanciamento del servizio Amazon ECS Availability Zone

Di seguito sono riportati alcuni esempi di messaggi relativi agli eventi di servizio che potresti visualizzare.

service (*service-name*) non è bilanciato in AZ con *number-tasks* le attività *number-tasks* in *Availability Zone 1**Availability Zone 2*, in e *number-tasks* in *Availability Zone 3*. Ribilanciamento AZ in corso.

Il service scheduler invia un evento di service (*service-name*) is not AZ balanced servizio quando il numero di attività non è distribuito uniformemente tra le zone di disponibilità. Non è necessario intraprendere alcuna azione. Si tratta di un evento informativo.

service (*service-name*) è bilanciato da AZ con *number-tasks**number-tasks* task in *Availability Zone 1**Availability Zone 2*, tasks in e *number-tasks* tasks in *Availability Zone 3*.

Il service scheduler invia un evento di service (*service-name*) is AZ balanced servizio al termine del ribilanciamento del servizio nella zona di disponibilità. Non c'è alcuna azione da intraprendere. Si tratta di un evento informativo.

service-name ha avviato ***number-tasks*** attività in ***Availability Zone*** AZ
Rebalance: ***task-ids***

Il service scheduler invia un evento ***service-name/task-set-name*** has started ***number*** tasks in ***Availability Zone*** service quando avvia attività in una zona di disponibilità a causa del ribilanciamento del servizio. Non è necessario intraprendere alcuna azione. Si tratta di un evento informativo.

service-name ha interrotto l'***number-tasks*** esecuzione delle attività a ***Availability Zone*** causa del ribilanciamento AZ: ***task-id***

Il service scheduler invia un evento ***service-name/task-set-name*** has started ***number*** tasks in ***Availability Zone*** service quando interrompe le attività in una zona di disponibilità a causa del ribilanciamento del servizio. Non è necessario intraprendere alcuna azione. Si tratta di un evento informativo.

service (***service-name***) non è in grado di inserire un'attività ***Availability Zone*** perché nessuna istanza del contenitore soddisfa tutti i requisiti.

Il service scheduler invia un evento che non ***service-name*** è in grado di inserire un'attività nel ***Availability Zone*** servizio perché nessuna istanza del contenitore soddisfa tutti i requisiti. Per risolvere il problema, avvia le istanze nella zona di disponibilità.

service (***service-name***) non è in grado di inserire un'attività in ***Availability Zone***.

Il service scheduler invia un ***service-name*** evento Impossibile inserire un'attività in ***Availability Zone*** servizio quando si utilizza il tipo di avvio Fargate e non c'è capacità disponibile.

È possibile aggiungere ulteriori sottoreti nella zona di disponibilità nel messaggio di errore oppure contattare Supporto per ottenere ulteriore capacità.

service (***service-name***) non è riuscito a eseguire AZ Rebalance perché non ***task-set-name*** è riuscito a scalare a causa di ***reason***

Il service scheduler invia un messaggio non riuscito ad AZ Rebalance perché non ***service-name*** è ***task-set-name*** stato in grado di eseguire la scalabilità a causa di un evento di ***reason*** servizio quando si utilizza la protezione scale-in dell'operazione.

Puoi effettuare una delle seguenti operazioni:

- Attendi la scadenza della protezione per le attività correnti, consentendone così la cessazione.
- Determina quali attività possono essere interrotte e utilizza l'UpdateTaskProtectionAPI con l'protectionEnabledopzione impostata per annullare la protezione false per queste attività.
- Aumenta il numero di attività desiderate del servizio portandolo a un numero maggiore del numero di attività protette.

service (**service-name**) ha interrotto AZ Rebalancing.

Il service scheduler invia un evento di servizio AZ Rebalancing **service-name** interrotto quando l'operazione di ribilanciamento della zona di disponibilità si è interrotta. Si tratta di un evento informativo. Amazon ECS invia eventi aggiuntivi che forniscono ulteriori informazioni.

Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS

I servizi Amazon ECS possono registrare i processi mediante un load balancer Elastic Load Balancing. Gli errori di configurazione del load balancer sono una causa comune dell'interruzione delle attività. Se le attività interrotte erano state avviate da servizi che utilizzano un load balancer, considera le seguenti possibili cause.

Il ruolo collegato al servizio Amazon ECS non esiste

Il ruolo collegato ai servizi Amazon ECS consente ai servizi Amazon ECS di registrare le istanze di container con Elastic Load Balancing. Il ruolo collegato ai servizi deve essere creato nel tuo account. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).

Gruppo di sicurezza delle istanze del contenitore

Se il tuo container è associato alla porta 80 della tua istanza di container, il gruppo di sicurezza per le istanze di container deve consentire il traffico in ingresso sulla porta 80 per garantire il superamento dei controlli dello stato del load balancer.

Il sistema di bilanciamento del carico Elastic Load Balancing non è configurato per tutte le zone di disponibilità

Il tuo load balancer deve essere configurato in modo da utilizzare tutte le zone di disponibilità all'interno di una regione o almeno tutte le zone di disponibilità in cui risiedono le tue istanze di container. Se un servizio utilizza un sistema di bilanciamento del carico e avvia un'attività su un'istanza di contenitore che risiede in una zona di disponibilità per la quale il bilanciamento del

carico non è configurato per l'utilizzo, l'attività non supera mai il controllo di integrità. Ciò comporta l'interruzione dell'attività.

Il controllo dello stato del load balancer Elastic Load Balancing non è configurato correttamente

I parametri di controllo dell'integrità del load balancer possono essere troppo restrittivi o puntare a risorse inesistenti. Se un'istanza del contenitore viene ritenuta non integra, viene rimossa dal sistema di bilanciamento del carico. Assicurati di verificare che i parametri seguenti siano configurati correttamente per il load balancer del tuo servizio.

Ping Port (Porta ping)

Il valore Ping Port per il controllo dello stato di un load balancer corrisponde alla porta delle istanze di container verificate dal load balancer per determinare se sono integre. Se questa porta non è configurata correttamente, il load balancer probabilmente revocherà la registrazione dell'istanza di container da se stessa. Questa porta deve essere configurata in modo da utilizzare il valore `hostPort` per il container nella definizione di attività del servizio che stai utilizzando con il controllo dell'integrità.

Ping Path (Percorso ping)

Questo fa parte dello stato di salute del sistema di bilanciamento del carico. È un endpoint dell'applicazione in grado di restituire un codice di stato corretto (ad esempio, 200) quando l'applicazione è integra. Questo valore viene spesso impostato su `index.html`, ma se il tuo servizio non risponde a tale richiesta, il controllo dell'integrità ha esito negativo. Se il tuo container non dispone di un file `index.html`, puoi impostare tale valore su `/` per indirizzare l'URL di base per l'istanza di container.

Response Timeout (Timeout di risposta)

Questo è il tempo entro il quale il tuo container deve restituire una risposta al ping del controllo dello stato. Se questo valore è inferiore al tempo necessario per una risposta, il controllo dello stato ha esito negativo.

Health Check Interval (Intervallo tra controlli dello stato)

Questo è il tempo compreso tra i ping del controllo dello stato. Più brevi sono gli intervalli tra i controlli dello stato, più rapidamente la tua istanza di container sarà in grado di raggiungere il valore `Unhealthy Threshold` (Soglia di mancata integrità).

Unhealthy Threshold (Soglia di mancata integrità)

Questo è il numero di volte che il controllo dello stato può avere esito negativo prima che la tua istanza di container venga considerata non integra. Se hai una soglia non integra di 2 e un

intervallo di controllo dello stato di 30 secondi, l'attività ha 60 secondi per rispondere al ping del controllo sanitario prima che venga considerata non integra. Puoi aumentare la soglia di mancata integrità o l'intervallo tra i controlli dello stato per concedere alle tue attività più tempo per rispondere.

Impossibile aggiornare il servizio **servicename**: il nome o la porta del contenitore Load Balancer sono stati modificati nella definizione dell'attività

Se il servizio utilizza un sistema di bilanciamento del carico, è possibile utilizzare l'SDK AWS CLI o l'SDK per modificare la configurazione del bilanciamento del carico. Per informazioni su come modificare la configurazione, consulta il riferimento [UpdateService](#) all'API di Amazon Elastic Container Service. Se aggiorni la definizione di attività per il servizio, il nome e la porta del container specificati nella configurazione del sistema di bilanciamento del carico devono rimanere nella definizione di attività.

Hai raggiunto il limite del numero di attività che puoi eseguire contemporaneamente.

Per un nuovo account, le quote potrebbero essere inferiori a quelle di servizio. La quota di servizio per l'account può essere visualizzata nella console Service Quotas. Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

Risoluzione dei problemi relativi alla scalabilità automatica del servizio in Amazon ECS

Application Auto Scaling disattiva i processi di scalabilità mentre le implementazioni di Amazon ECS sono in corso e riprendono una volta completata la distribuzione. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta [Sospensione e ripresa del dimensionamento per Application Auto Scaling](#).

Risolvi gli errori di CPU o memoria non validi relativi alla definizione delle attività di Amazon ECS

Quando si registra una definizione di attività utilizzando l'API Amazon ECS o AWS CLI, se si specifica un memory valore cpu o non valido, viene restituito il seguente errore.

```
An error occurred (ClientException) when calling the RegisterTaskDefinition operation:  
Invalid 'cpu' setting for task.
```

 Note

Quando si utilizza Terraform, potrebbe essere restituito il seguente errore.

```
Error: ClientException: No Fargate configuration exists for given values.
```

Per risolvere questo problema, devi specificare un valore di CPU e memoria supportato nella definizione di attività. Il `cpu` valore può essere espresso in unità CPU o v CPUs in una definizione di attività. Viene convertito in un numero intero che indica le unità CPU al momento della registrazione della definizione dell'attività. Il `memory` valore può essere espresso in MiB o GB in una definizione di attività. Viene convertito in un numero intero che indica il MiB quando viene registrata la definizione dell'attività.

Per le definizioni delle attività che specificano FARGATE il `requiresCompatibilities` parametro (anche se EC2 è specificato anche), è necessario utilizzare uno dei valori nella tabella seguente. Questi valori determinano l'intervallo di valori supportati per il parametro CPU e memoria.

Per i processi ospitati su Fargate, nella tabella seguente sono riportate le combinazioni valide di CPU e memoria. I valori di memoria nel file JSON sono specificati in MiB. Puoi convertire il valore in GB in MiB moltiplicando il valore per 1.024. Ad esempio 1 GB = 1.024 MiB.

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
256 (0,25 vCPU)	512 MiB, 1 GB, 2 GB	Linux
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB	Linux
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB	Linux, Windows
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
8192 (8 vCPU)  Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
16384 (16vCPU)  Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

Per le attività ospitate su Amazon EC2, i valori della CPU delle attività supportati sono compresi tra 0,25 v CPUs e 192 v. CPUs

Amazon ECS utilizza il periodo e la quota della CPU per controllare le dimensioni delle attività, i limiti rigidi della CPU. Quando specifichi la vCPU nella definizione dell'attività, Amazon ECS traduce il valore nel periodo della CPU e nelle impostazioni della quota CPU applicabili a `cgroup`

La quota CPU controlla la quantità di tempo di CPU concessa a un `cgroup` durante un determinato periodo di CPU. Entrambe le impostazioni sono espresse in termini di microsecondi. Quando la quota della CPU è uguale al periodo della CPU significa che un `cgroup` può essere eseguito fino al 100% su una vCPU (o qualsiasi altra frazione che ammonta al 100% per più v). CPUs La quota CPU ha un massimo di 1000000us e il periodo della CPU ha un minimo di 1 ms. È possibile utilizzare questi valori per impostare i limiti per il numero di CPU. Quando si modifica il periodo della CPU senza modificare la quota della CPU, i limiti effettivi sono diversi da quelli specificati nella definizione dell'attività.

Il periodo di 100 ms consente un intervallo di v CPUs compreso tra 0,125 e 10.

Note

I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows.

Visualizzazione dei log degli agenti container Amazon ECS

Amazon ECS archivia i log nella cartella `/var/log/ecs` delle tue istanze di container. Sono disponibili log dell'agente dal container Amazon ECS e dal servizio `ecs-init` che controlla lo stato (di avvio/arresto) dell'agente sull'istanza di container. Puoi visualizzare tali file di log connettendoti a un'istanza di container tramite SSH.

Note

Se hai dubbi su come raccogliere tutti i log nelle tue istanze di container, puoi utilizzare il programma di raccolta log di Amazon ECS. Per ulteriori informazioni, consulta [Raccolta dei log dei container con Amazon ECS logs collector](#).

Sistema operativo Linux

Il processo `ecs-init` archivia i log in `/var/log/ecs/ecs-init.log`.

Il `ecs-init.log` file contiene informazioni sulla gestione, la configurazione e il bootstrap del ciclo di vita degli agenti container.

Puoi utilizzare il comando seguente per visualizzare i file di log.

```
cat /var/log/ecs/ecs-init.log
```

Output:

```
2018-02-16T18:13:54Z [INFO] pre-start
2018-02-16T18:13:56Z [INFO] start
2018-02-16T18:13:56Z [INFO] No existing agent container to remove.
```

```
2018-02-16T18:13:56Z [INFO] Starting Amazon Elastic Container Service Agent
```

Sistema operativo Windows

Puoi usare Amazon ECS logs collector per Windows. Per ulteriori informazioni, consulta [Amazon ECS Logs Collector per Windows](#) su Github.

1. Connettiti alla tua istanza.
2. Apri PowerShell ed esegui i seguenti comandi con privilegi amministrativi. I comandi scaricano lo script e raccolgono i log.

```
Invoke-WebRequest -OutFile ecs-logs-collector.ps1 https://  
raw.githubusercontent.com/aws-labs/aws-ecs-logs-collector-for-windows/master/ecs-  
logs-collector.ps1  
.\ecs-logs-collector.ps1
```

Puoi attivare la registrazione di debug per l'agente Amazon ECS e il daemon Docker. Questa opzione consente allo script di raccogliere i log prima di attivare la modalità di debug. Lo script riavvia il daemon Docker e l'agente Amazon ECS, quindi termina tutti i contenitori in esecuzione sull'istanza. Prima di eseguire il comando seguente, svuota l'istanza del contenitore e sposta tutte le attività importanti su altre istanze del contenitore.

Esegui il comando seguente per attivare la registrazione.

```
.\ecs-logs-collector.ps1 -RunMode debug
```

Raccolta dei log dei container con Amazon ECS logs collector

Se hai dubbi su come raccogliere tutti i vari log nelle tue istanze di container, puoi utilizzare il programma di raccolta log di Amazon ECS. [È disponibile sia per Linux che GitHub per Windows.](#) Lo script raccoglie i log generali del sistema operativo, nonché i log degli agenti container Docker e Amazon ECS, che possono essere utili per la risoluzione dei casi. Supporto AWS Quindi comprime e archivia le informazioni raccolte in un singolo file che può essere facilmente condiviso per scopi di diagnostica. Supporta inoltre l'abilitazione della modalità di debug per il daemon Docker e l'agente del container di Amazon ECS nelle varianti di Amazon Linux, ad esempio l'AMI ottimizzata per Amazon ECS. Al momento, il programma di raccolta log di Amazon ECS supporta i seguenti sistemi operativi:

- Amazon Linux

- Red Hat Enterprise Linux 7
- Debian 8
- Ubuntu 14.04
- Ubuntu 16.04
- Ubuntu 18.04
- Windows Server 2016

Note

[Il codice sorgente per il raccoglitore di log di Amazon ECS è disponibile sia GitHub per Linux che per Windows.](#) Consigliamo di inviare le richieste pull per le modifiche che desideri siano incluse. Tuttavia, Amazon Web Services al momento non supporta l'esecuzione di copie modificate di questo software.

Come scaricare ed eseguire il programma di raccolta log di Amazon ECS per Linux

1. Connettiti alla tua istanza di container.
2. Scarica lo script del programma di raccolta di log di Amazon ECS.

```
curl -O https://raw.githubusercontent.com/aws-labs/ecs-logs-collector/master/ecs-logs-collector.sh
```

3. Esegui lo script per raccogliere i log e creare l'archivio.

Note

Per abilitare la modalità di debug per il demone Docker e l'agente contenitore Amazon ECS, aggiungi l'--mode=enable-debugopzione al comando seguente. Ciò potrebbe riavviare il demone Docker, che uccide tutti i contenitori in esecuzione sull'istanza. Prendi in considerazione di svuotare l'istanza di container e di spostare eventuali attività importanti su altre istanze di container prima di abilitare la modalità di debug. Per ulteriori informazioni, consulta [Drenaggio delle istanze di container Amazon ECS](#).

```
[ec2-user ~]$ sudo bash ./ecs-logs-collector.sh
```

⚠ Important

Ti consigliamo di modificare i log e rimuovere tutti i dati sensibili dai file. Puoi cercare dati noti e anche cercare variabili di ambiente come `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, e `AWS_SESSION_TOKEN` nel file.

Dopo aver eseguito lo script, puoi esaminare i log raccolti nella cartella `collect` creata dallo script. Il `collect.tgz` file è un archivio compresso di tutti i registri, che è possibile condividere con il Supporto AWS per facilitare la diagnostica.

Per scaricare ed eseguire il programma di raccolta log di Amazon ECS per Windows

1. Connettiti alla tua istanza di container. Per ulteriori informazioni, consulta [Connect to your Windows using RDP](#) nella Amazon EC2 User Guide.
2. Scarica lo script di raccolta dei log di Amazon ECS utilizzando PowerShell

```
Invoke-WebRequest -OutFile ecs-logs-collector.ps1 https://raw.githubusercontent.com/aws-labs/aws-ecs-logs-collector-for-windows/master/ecs-logs-collector.ps1
```

3. Esegui lo script per raccogliere i log e creare l'archivio.

ℹ Note

Per abilitare la modalità di debug per il demone Docker e l'agente contenitore Amazon ECS, aggiungi l'opzione `-RunMode debug` al comando seguente. Questo comporta il riavvio del daemon Docker, con conseguente interruzione di tutti i container in esecuzione nell'istanza. Prendi in considerazione di svuotare l'istanza di container e di spostare eventuali attività importanti su altre istanze di container prima di abilitare la modalità di debug. Per ulteriori informazioni, consulta [Drenaggio delle istanze di container Amazon ECS](#).

```
.\ecs-logs-collector.ps1
```


⚠ Important

Ti consigliamo di modificare i log e rimuovere tutti i dati sensibili dai file. Puoi cercare dati noti e anche cercare variabili di ambiente come `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, e `AWS_SESSION_TOKEN` nel file.

Dopo aver eseguito lo script, puoi esaminare i log raccolti nella cartella `collect` creata dallo script. Il `collect.tgz` file è un archivio compresso di tutti i registri, che puoi condividere con Support per ricevere AWS assistenza diagnostica.

Recupera i dettagli diagnostici di Amazon ECS con l'introspezione degli agenti

L'API di introspezione dell'agente Amazon ECS fornisce informazioni sullo stato generale dell'agente Amazon ECS e delle istanze del contenitore.

Puoi utilizzare l'API di introspezione dell'agente per ottenere l'ID Docker per un contenitore nell'ambito della tua attività. Puoi utilizzare l'API di introspezione dell'agente connettendoti a un'istanza di container tramite SSH.

⚠ Important

L'istanza di container deve disporre di un ruolo IAM che consenta l'accesso ad Amazon ECS per raggiungere l'API di introspezione. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

L'esempio seguente mostra due attività, una attualmente in esecuzione e una che è stata interrotta.

i Note

Il comando seguente viene reindirizzato tramite il `python -mjson.tool` per una maggiore leggibilità.

```
curl http://localhost:51678/v1/tasks | python -mjson.tool
```

Output:

```

% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload   Total     Spent    Left     Speed
100 1095  100 1095    0     0  117k      0 --:--:-- --:--:-- --:--:-- 133k
{
  "Tasks": [
    {
      "Arn": "arn:aws:ecs:us-west-2:aws_account_id:task/090eff9b-1ce3-4db6-848a-a8d14064fd24",
      "Containers": [
        {
          "DockerId":
            "189a8ff4b5f04affe40e5160a5ffadca395136eb5faf4950c57963c06f82c76d",
          "DockerName": "ecs-console-sample-app-static-6-simple-app-86caf9bcabe3e9c61600",
          "Name": "simple-app"
        },
        {
          "DockerId":
            "f7f1f8a7a245c5da83aa92729bd28c6bcb004d1f6a35409e4207e1d34030e966",
          "DockerName": "ecs-console-sample-app-static-6-busybox-ce83ce978a87a890ab01",
          "Name": "busybox"
        }
      ],
      "Family": "console-sample-app-static",
      "KnownStatus": "STOPPED",
      "Version": "6"
    },
    {
      "Arn": "arn:aws:ecs:us-west-2:aws_account_id:task/1810e302-eaea-4da9-a638-097bea534740",
      "Containers": [
        {
          "DockerId":
            "dc7240fe892ab233dbbcee5044d95e1456c120dba9a6b56ec513da45c38e3aeb",
          "DockerName": "ecs-console-sample-app-static-6-simple-app-f0e5859699a7aecfb101",
          "Name": "simple-app"
        },
        {
          "DockerId":
            "096d685fb85a1ff3e021c8254672ab8497e3c13986b9cf005cbae9460b7b901e",

```

```
        "DockerName": "ecs-console-sample-app-static-6-  
        busybox-92e4b8d0ecd0cce69a01",  
        "Name": "busybox"  
    },  
    ],  
    "DesiredStatus": "RUNNING",  
    "Family": "console-sample-app-static",  
    "KnownStatus": "RUNNING",  
    "Version": "6"  
}  
]  
}
```

Nell'esempio precedente, il task stoppato (*090eff9b-1ce3-4db6-848a-a8d14064fd24*) ha due contenitori. Puoi utilizzare `docker inspect` **container-ID** per visualizzare informazioni dettagliate su ogni container. Per ulteriori informazioni, consulta [Introspezione dei container Amazon ECS](#).

Diagnostica Docker in Amazon ECS

Docker fornisce diversi strumenti di diagnostica che ti aiutano a risolvere i problemi relativi a container e attività. Per ulteriori informazioni su tutte le utilità della riga di comando Docker disponibili, consulta il riferimento alla [CLI Docker](#) nella documentazione Docker. Puoi accedere alle utility a riga di comando Docker connettendoti a un'istanza di container tramite SSH.

Anche i codici di uscita riportati dai container Docker possono fornire informazioni di diagnostica (ad esempio, il codice di uscita 137 indica che il container ha ricevuto un segnale SIGKILL). Per ulteriori informazioni, consulta la sezione relativa allo [stato di uscita](#) nella documentazione Docker.

Elenca i contenitori Docker in Amazon ECS

Puoi utilizzare il comando `docker ps` sulla tua istanza di container per visualizzare l'elenco dei container in esecuzione. Nell'esempio seguente, è in esecuzione solo l'agente container Amazon ECS. Per ulteriori informazioni, consulta la sezione relativa al comando [docker ps](#) nella documentazione Docker.

```
docker ps
```

Output:

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
cee0d6986de0	amazon/amazon-ecs-agent:latest	"/agent"	22 hours ago
Up 22 hours	127.0.0.1:51678->51678/tcp	ecs-agent	

Puoi utilizzare il comando `docker ps -a` per visualizzare tutti i container (anche quelli sospesi o interrotti). Questo comando è utile per visualizzare l'elenco dei container che si arrestano inaspettatamente. Nell'esempio seguente, il container `f7f1f8a7a245` è stato chiuso 9 secondi fa, pertanto non compare nell'output del comando `docker ps` senza il flag `-a`.

```
docker ps -a
```

Output:

CONTAINER ID	IMAGE	COMMAND	
CREATED	STATUS	PORTS	NAMES
db4d48e411b1	amazon/ecs-emptyvolume-base:autogenerated	"not-applicable"	
19 seconds ago			ecs-
console-sample-app-static-6-internalecs-emptyvolume-source-c09288a6b0cba8a53700			
f7f1f8a7a245	busybox:buildroot-2014.02	"\"sh -c '/bin/sh -c	
22 hours ago	Exited (137) 9 seconds ago		ecs-
console-sample-app-static-6-busybox-ce83ce978a87a890ab01			
189a8ff4b5f0	httpd:2	"httpd-foreground"	
22 hours ago	Exited (137) 40 seconds ago		ecs-
console-sample-app-static-6-simple-app-86caf9bcabe3e9c61600			
0c7dca9321e3	amazon/ecs-emptyvolume-base:autogenerated	"not-applicable"	
22 hours ago			ecs-
console-sample-app-static-6-internalecs-emptyvolume-source-90fefaa68498a8a80700			
cee0d6986de0	amazon/amazon-ecs-agent:latest	"/agent"	
22 hours ago	Up 22 hours	127.0.0.1:51678->51678/tcp	ecs-
agent			

Visualizza i log Docker in Amazon ECS

Puoi visualizzare i flussi `STDOUT` e `STDERR` relativi a un container mediante il comando `docker logs`. In questo esempio, i log vengono visualizzati per il `dc7240fe892a` contenitore e reindirizzati tramite il comando per brevità. `head` Per ulteriori informazioni, consulta la sezione relativa al comando [docker logs](#) nella documentazione Docker.

 Note

I log di Docker sono disponibili nell'istanza di container solo se usi il driver di log `json` predefinito. Se le attività sono state configurate per l'utilizzo del driver di `awslogs` registro, i log del contenitore sono disponibili in Logs. CloudWatch Per ulteriori informazioni, consulta [Invia i log di Amazon ECS a CloudWatch](#).

```
docker logs dc7240fe892a | head
```

Output:

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
using 172.17.0.11. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
using 172.17.0.11. Set the 'ServerName' directive globally to suppress this message
[Thu Apr 23 19:48:36.956682 2015] [mpm_event:notice] [pid 1:tid 140327115417472]
AH00489: Apache/2.4.12 (Unix) configured -- resuming normal operations
[Thu Apr 23 19:48:36.956827 2015] [core:notice] [pid 1:tid 140327115417472] AH00094:
Command line: 'httpd -D FOREGROUND'
10.0.1.86 - - [23/Apr/2015:19:48:59 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:48:59 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:49:28 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:49:29 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:49:50 +0000] "-" 408 -
10.0.0.154 - - [23/Apr/2015:19:49:50 +0000] "-" 408 -
10.0.1.86 - - [23/Apr/2015:19:49:58 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:49:59 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:50:28 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:50:29 +0000] "GET / HTTP/1.1" 200 348
time="2015-04-23T20:11:20Z" level="fatal" msg="write /dev/stdout: broken pipe"
```

Ispeziona i contenitori Docker in Amazon ECS

Se disponi dell'ID Docker di un container, puoi ispezionarlo mediante il comando `docker inspect`. L'ispezione dei container fornisce le informazioni più dettagliate sull'ambiente in cui è stato avviato un container. Per ulteriori informazioni, consulta la sezione relativa al comando [docker inspect](#) nella documentazione Docker.

```
docker inspect dc7240fe892a
```

Output:

```
[{
  "AppArmorProfile": "",
  "Args": [],
  "Config": {
    "AttachStderr": false,
    "AttachStdin": false,
    "AttachStdout": false,
    "Cmd": [
      "httpd-foreground"
    ],
    "CpuShares": 10,
    "Cpuset": "",
    "Domainname": "",
    "Entrypoint": null,
    "Env": [
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
local/apache2/bin",
      "HTTPD_PREFIX=/usr/local/apache2",
      "HTTPD_VERSION=2.4.12",
      "HTTPD_BZ2_URL=https://www.apache.org/dist/httpd/httpd-2.4.12.tar.bz2"
    ],
    "ExposedPorts": {
      "80/tcp": {}
    },
    "Hostname": "dc7240fe892a",
    ...
  }
}]
```

Configurazione dell'output verboso dal demone Docker in Amazon ECS

Se hai problemi con i contenitori o le immagini Docker, puoi attivare la modalità di debug sul tuo demone Docker. L'uso del debug fornisce un output più dettagliato dal demone. È possibile utilizzarlo per recuperare i messaggi di errore inviati dai registri dei container, come Amazon ECR.

 Important

Questa procedura è scritta per l'AMI Amazon Linux ottimizzata per Amazon ECS. [Per altri sistemi operativi, consulta *Abilita il debug e Controlla e configura Docker con systemd* nella documentazione Docker.](#)

Per utilizzare la modalità di debug del demone Docker sull'AMI Amazon Linux ottimizzata per Amazon ECS

1. Connettiti alla tua istanza di container.
2. Aprire il file delle opzioni Docker con un editor di testo, ad esempio vi. Per l'AMI Amazon Linux ottimizzata per Amazon ECS, il file delle opzioni Docker si trova in `/etc/sysconfig/docker`.
3. Trova l'istruzione delle opzioni Docker e aggiungi l'opzione `-D` alla stringa, tra virgolette.

 Note

Se l'istruzione delle opzioni Docker inizia con `#`, rimuovi questo carattere dall'istruzione e abilita le opzioni.

Per l'AMI Amazon Linux ottimizzata per Amazon ECS, l'istruzione delle opzioni Docker si chiama `OPTIONS`. Ad esempio:

```
# Additional startup options for the Docker daemon, for example:
# OPTIONS="--ip-forward=true --iptables=true"
# By default we limit the number of open files per container
OPTIONS="-D --default-ulimit nofile=1024:4096"
```

4. Salva il file ed esci dall'editor di testo.
5. Riavvia il daemon Docker.

```
sudo service docker restart
```

L'output è il seguente:

```
Stopping docker:                                [ OK ]
Starting docker: .                             [ OK ]
```

6. Riavvia l'agente Amazon ECS.

```
sudo service ecs restart
```

Il tuo log di Docker dovrebbe ora mostrare un output più dettagliato.

```
time="2015-12-30T21:48:21.907640838Z" level=debug msg="Unexpected response from
server: \"{\\\"errors\\\": [{\\\"code\\\": \\\"DENIED\\\", \\\"message\\\": \\\"User:
arn:aws:sts::1111:assumed-role/ecrReadOnly/i-abcdefg is not authorized to perform:
ecr:InitiateLayerUpload on resource: arn:aws:ecr:us-east-1:1111:repository/nginx_test
\\\"]}]\\\" http.Header{\\\"Connection\\\": []string{\\\"keep-alive\\\"}, \\\"Content-Type\\\":
[]string{\\\"application/json; charset=utf-8\\\"}, \\\"Date\\\": []string{\\\"Wed, 30 Dec 2015
21:48:21 GMT\\\"}, \\\"Docker-Distribution-Api-Version\\\": []string{\\\"registry/2.0\\\"},
\\\"Content-Length\\\": []string{\\\"235\\\"}}"
```

Risolvi i problemi relativi al Docker in Amazon **API error (500): devmapper ECS**

Il seguente errore Docker indica che lo spazio di storage del thin pool della tua istanza di container è pieno e che il daemon Docker non può creare nuovi container:

```
CannotCreateContainerError: API error (500): devmapper: Thin Pool has 4350 free data
blocks which is less than minimum required 4454 free data blocks. Create more free
space in thin pool or use dm.min_free_space option to change behavior
```

Per impostazione predefinita, Amazon Linux ottimizzato per Amazon ECS AMIs dalla versione precedente 2015.09.d e successiva viene lanciato con un volume da 8 GiB per il sistema operativo collegato `/dev/xvda` e montato come radice del file system. È previsto un volume aggiuntivo di 22 GiB collegato a `/dev/xvdcz` che Docker utilizza per l'archiviazione di immagini e metadati. Se questo spazio di storage è pieno, il daemon Docker non può creare nuovi container.

Il modo più semplice per aggiungere storage alle istanze di container è quello di terminare le istanze esistenti e avviarne di nuove con volumi di storage dei dati superiori. Tuttavia, se non riesci a completare questa operazione, puoi aggiungere archiviazione al gruppo di volumi utilizzati da Docker ed estendere il relativo volume logico seguendo le procedure riportate in [Linux ottimizzato per Amazon ECS AMIs](#).

Se lo storage dell'istanza di container si sta riempiendo troppo rapidamente, puoi eseguire alcune operazioni per ridurre questo effetto:

- Per visualizzare le informazioni di thin poll, esegui il comando seguente nella tua istanza di container:

```
docker info
```

- (Agente container Amazon ECS 1.8.0 e versioni successive) Puoi ridurre la quantità di tempo in cui i container fermi o usciti rimangono sulle tue istanze di container. La variabile di configurazione dell'agente ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION imposta il tempo di attesa dall'interruzione di un'attività fino alla rimozione del container Docker (di default, questo valore è 3 ore). Questo rimuove i dati del container di Docker. Se questo valore è impostato su un valore troppo basso, potresti non essere in grado di ispezionare i contenitori fermi o visualizzare i log prima che vengano rimossi. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).
- Puoi rimuovere i contenitori non in esecuzione e le immagini non utilizzate dalle istanze dei contenitori. Puoi utilizzare i seguenti comandi di esempio per rimuovere manualmente i container interrotti e le immagini inutilizzate. I container eliminati non possono essere esaminati in un secondo momento e le immagini eliminate devono essere recuperate nuovamente prima di poter avviare i nuovi container da esse.

Per rimuovere i container non in esecuzione, esegui il comando riportato nell'istanza di container:

```
docker rm $(docker ps -aq)
```

Per rimuovere le immagini inutilizzate, emetti il comando seguente nell'istanza di container:

```
docker rmi $(docker images -q)
```

- Puoi rimuovere i blocchi di dati non utilizzati all'interno dei contenitori. Puoi utilizzare il seguente comando per eseguire fstrim in qualsiasi container in esecuzione e ignorare tutti i blocchi di dati inutilizzati dal file system del container.

```
sudo sh -c "docker ps -q | xargs docker inspect --format='{{ .State.Pid }}' | xargs -IZ fstrim /proc/Z/root/"
```

Risolvi i problemi di Amazon ECS Exec

Di seguito sono riportate note sulla risoluzione dei problemi che consentono di diagnosticare il motivo per cui potrebbe essere visualizzato un errore durante l'utilizzo di ECS Exec.

Verifica utilizzando Exec Checker

Lo script ECS Exec Checker fornisce un modo per verificare e convalidare che il cluster e l'attività Amazon ECS soddisfino i prerequisiti per l'utilizzo della funzionalità ECS Exec. Lo script ECS Exec Checker verifica che l'AWS CLI ambiente e il cluster e le attività siano pronte per ECS Exec, chiamandone diverse per tuo conto. APIs Lo strumento richiede la versione più recente di e che sia disponibile. AWS CLI jq Per ulteriori informazioni, consulta [ECS Exec Checker on GitHub](#).

Errore durante la chiamata a **execute-command**

Di seguito sono riportate le cause possibili se si verifica un errore `The execute command failed`.

- Il processo non dispone delle autorizzazioni necessarie. Verifica che nella definizione di attività utilizzata per avviare il processo sia definito un ruolo IAM del processo e che il ruolo disponga delle autorizzazioni richieste. Per ulteriori informazioni, consulta [Autorizzazioni ECS Exec](#).
- L'agente SSM non è installato o non è in esecuzione.
- Esiste un'interfaccia Amazon VPC endpoint per Amazon ECS, ma non ce n'è una per Systems Manager Session Manager.

Risolvi i problemi relativi ad Amazon ECS Anywhere

Amazon ECS Anywhere fornisce supporto per la registrazione di un'istanza esterna come un server locale o una macchina virtuale (VM) nel tuo cluster Amazon ECS. Di seguito sono riportati i problemi più comuni riscontrati e i suggerimenti generali per la risoluzione dei problemi.

Argomenti

- [Problemi di registrazione delle istanze esterne](#)
- [Problemi di rete delle istanze esterne](#)
- [Problemi durante l'esecuzione di processi sull'istanza esterna](#)

Problemi di registrazione delle istanze esterne

Durante la registrazione di un'istanza esterna nel tuo cluster Amazon ECS, è necessario soddisfare i seguenti requisiti:

- È necessario AWS Systems Manager recuperare un'attivazione, che consiste in un ID di attivazione e un codice di attivazione. Si utilizza per registrare l'istanza esterna come istanza gestita da Systems Manager. Quando viene richiesta l'attivazione di Systems Manager, specificare un limite di registrazione e una data di scadenza. Il limite di registrazione specifica il numero massimo di istanze che possono essere registrate mediante l'attivazione. Il valore predefinito per il limite di registrazione è 1 instance. La data di scadenza specifica la data di scadenza dell'attivazione. Il valore di default è 24 ore. Se l'attivazione di Systems Manager utilizzata per registrare l'istanza esterna non è valida, richiederne una nuova. Per ulteriori informazioni, consulta [Registrazione di un'istanza esterna in un cluster Amazon ECS](#).
- Una policy IAM viene utilizzata per fornire all'istanza esterna le autorizzazioni necessarie per comunicare con le operazioni AWS API. Se questa policy gestita non viene creato correttamente e non contiene le autorizzazioni richieste, la registrazione dell'istanza esterna non riesce. Per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#).
- Amazon ECS fornisce uno script di installazione che installa Docker, l'agente del container Amazon ECS e Systems Manager Agent nell'istanza esterna. Se lo script di installazione non riesce, è probabile che non possa essere eseguito nuovamente sulla stessa istanza senza che si verifichi un errore. In tal caso, segui la procedura di pulizia per eliminare AWS le risorse dall'istanza in modo da poter eseguire nuovamente lo script di installazione. Per ulteriori informazioni, consulta [Annullamento della registrazione di un'istanza esterna Amazon ECS](#).

Note

Tieni presente che, se lo script di installazione ha richiesto e utilizzato correttamente l'attivazione di Systems Manager, l'esecuzione dello script di installazione una seconda volta utilizza nuovamente l'attivazione di Systems Manager. Ciò potrebbe a sua volta causare il raggiungimento del limite di registrazione per tale attivazione. Se questo limite viene raggiunto, sarà necessario creare una nuova attivazione.

- Quando si esegue lo script di installazione su un'istanza esterna per carichi di lavoro GPU, se il driver NVIDIA non viene rilevato o configurato correttamente, si verificherà un errore. Lo script di installazione utilizza il comando `nvidia-smi` per confermare l'esistenza del driver NVIDIA.

Problemi di rete delle istanze esterne

Per comunicare eventuali modifiche, l'istanza esterna richiede una connessione di rete a AWS. Se l'istanza esterna perde la connessione di rete a AWS, le attività in esecuzione sulle istanze continuano comunque a essere eseguite a meno che non vengano interrotte manualmente. Dopo il ripristino della AWS connessione a, le AWS credenziali utilizzate dall'agente contenitore Amazon ECS e dall'agente Systems Manager sull'istanza esterna si rinnovano automaticamente. Per ulteriori informazioni sui AWS domini utilizzati per la comunicazione tra l'istanza esterna e, consulta [AWSRete](#)

Problemi durante l'esecuzione di processi sull'istanza esterna

Se i processi o i container non vengono eseguiti nell'istanza esterna, le cause più comuni sono relative alla rete o alle autorizzazioni. Se i tuoi container stanno estraendo le loro immagini da Amazon ECR o sono configurati per inviare i log dei container a CloudWatch Logs, la definizione dell'attività deve specificare un ruolo IAM valido per l'esecuzione delle attività. Senza un ruolo IAM di esecuzione processo valido, i container non saranno avviati. Per ulteriori informazioni sui problemi della rete, consulta [Problemi di rete delle istanze esterne](#).

Important

Amazon ECS fornisce lo strumento di raccolta dei log di Amazon ECS. È possibile utilizzarlo per raccogliere i log dalle istanze esterne a scopo di risoluzione dei problemi. Per ulteriori informazioni, consulta [Raccolta dei log dei container con Amazon ECS logs collector](#).

AWS Fargate limitazione delle quote

AWS Fargate limita le attività di Amazon ECS e i tassi di lancio dei pod Amazon EKS a quote (precedentemente denominate limiti) utilizzando un [algoritmo token bucket](#) per ogni AWS account in base alla regione. Con questo algoritmo, il tuo account ha un bucket che contiene un numero specifico di token. Il numero di token nel bucket rappresenta la tua quota di velocità in un dato secondo. Ogni account cliente ha un bucket di token per le attività e per i pod che si esaurisce in base al numero di attività e pod avviati dall'account cliente. Questo bucket di token prevede un massimo, che consente di effettuare periodicamente un numero maggiore di richieste, e una velocità di ricarica, che consente di sostenere una velocità costante di richieste per tutto il tempo necessario.

Ad esempio, la dimensione del bucket di token di attività e pod per un account cliente Fargate è di 100 token e la velocità di ricarica è di 20 token al secondo. Pertanto, puoi avviare immediatamente fino a 100 attività Amazon ECS e pod Amazon EKS per ogni account cliente, con una velocità di avvio sostenuta di 20 attività Amazon ECS e pod Amazon EKS al secondo.

Azioni	Capacità massima del bucket (o velocità di espansione)	Velocità di ricarica del bucket (o velocità sostenuta)
Quota di velocità delle risorse di Fargate per le attività Amazon ECS on demand e i pod Amazon EKS ¹	100	20
Quota di velocità delle risorse di Fargate per le attività Spot Amazon ECS	100	20

¹Gli account che avviano solo pod Amazon EKS hanno una velocità di espansione uguale a 20, con una velocità di avvio sostenuta di 20 avvii di pod al secondo, quando si utilizzano le versioni della piattaforma indicate nella sezione [Versioni della piattaforma Amazon EKS](#).

Limitazione dell'RunTaskAPI in Fargate

Inoltre, Fargate limita la velocità delle richieste durante l'avvio di attività tramite l'API RunTask di Amazon ECS, che utilizza una quota separata. Fargate limita le richieste RunTask API Amazon ECS per ogni AWS account in base alla regione. Ogni richiesta effettuata rimuove un token dal bucket. Ciò consente di aiutare le prestazioni del servizio e garantire un utilizzo equo a tutti i clienti di Fargate. Le chiamate API sono soggette alle quote di richieste, indipendentemente dal fatto che abbiano origine dalla console Amazon Elastic Container Service, da uno strumento a riga di comando o da un'applicazione di terze parti. La quota di velocità per le chiamate verso l'API RunTask di Amazon ECS è di 20 chiamate al secondo (di espansione e sostenuta). Tuttavia, ogni chiamata a questa API può avviare fino a 10 attività. Ciò significa che è possibile avviare 100 attività in un secondo effettuando 10 chiamate a questa API, richiedendo l'avvio di 10 attività in ogni chiamata. Allo stesso modo, è possibile effettuare 20 chiamate a questa API richiedendo l'avvio di 5 attività in ogni chiamata. Per ulteriori informazioni sulla limitazione delle API per l'API Amazon ECS, consulta la sezione RunTask Limitazione delle [richieste API nell'Amazon ECS API Reference](#).

Nella pratica, le velocità di avvio delle attività e dei pod dipendono anche da altri fattori, come le immagini dei container da scaricare e decomprimere, i controlli dell'integrità e altre integrazioni abilitate, come la registrazione di attività o di pod con un load balancer. I clienti riscontrano variazioni nei tassi di avvio di task e pod rispetto alle quote rappresentate in precedenza, in base alle funzionalità abilitate dai clienti.

Adeguamento delle quote tariffarie a Fargate

Puoi richiedere un aumento delle quote di limitazione di velocità di Fargate per il tuo account AWS . Per ulteriori informazioni, consulta [Richiesta di un aumento di quota](#) nel Manuale dell'utente di Service Quotas .

Gestisci i problemi di limitazione di Amazon ECS

Gli errori di limitazione si dividono in due categorie principali: limitazione sincrona e limitazione asincrona.

Limitazione sincrona

Quando si verifica una limitazione sincrona, ricevi immediatamente una risposta di errore da Amazon ECS. Questa categoria si verifica in genere quando chiami Amazon ECS APIs durante l'esecuzione di attività o la creazione di servizi. Per ulteriori informazioni sulla limitazione coinvolta e sui relativi limiti di accelerazione, consulta [Request throttling for](#) the Amazon ECS API.

Quando l'applicazione avvia richieste API, ad esempio utilizzando l'SDK AWS CLI o un AWS SDK, puoi porre rimedio alla limitazione delle API. Puoi farlo progettando l'applicazione in modo da gestire gli errori o implementando una strategia esponenziale di backoff e jitter con logica di ripetizione per le chiamate API. Per ulteriori informazioni, consulta [Timeouts](#), retries e backoff with jitter.

Se utilizzi un AWS SDK, la logica di riprova automatica è integrata e configurabile.

Throttling asincrono

La limitazione asincrona si verifica a causa di flussi di lavoro asincroni in cui Amazon ECS o AWS CloudFormation potrebbe chiamare per tuo conto per fornire risorse. APIs È importante sapere cosa Amazon ECS richiama per tuo conto. AWS APIs Ad esempio, l>CreateNetworkInterfaceAPI viene richiamata per attività che utilizzano la modalità di awsvpc rete e l'DescribeTargetHealthAPI viene richiamata quando si eseguono controlli di integrità per attività registrate su un sistema di bilanciamento del carico.

Quando i carichi di lavoro raggiungono una scala considerevole, queste operazioni API potrebbero essere limitate. Cioè, potrebbero essere sufficientemente limitati da violare i limiti imposti da Amazon ECS o da Servizio AWS quello che viene chiamato. Ad esempio, se distribuisce centinaia di servizi, ciascuno con centinaia di attività contemporaneamente che utilizzano la modalità di `aws_vpc` rete, Amazon ECS richiama le operazioni dell'API Amazon come le operazioni dell' EC2 API Elastic Load `CreateNetworkInterface` Balancing come o per registrare rispettivamente l'interfaccia di rete `DescribeTargetHealth` elastica e `RegisterTarget` il load balancer. Queste chiamate API possono superare i limiti dell'API, con conseguenti errori di limitazione. Di seguito è riportato un esempio di errore di limitazione di Elastic Load Balancing incluso nel messaggio relativo all'evento del servizio.

```
{
  "userIdentity":{
    "arn":"arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForECS/ecs-service-scheduler",
    "eventTime":"2022-03-21T08:11:24Z",
    "eventSource":"elasticloadbalancing.amazonaws.com",
    "eventName":" DescribeTargetHealth ",
    "awsRegion":"us-east-1",
    "sourceIPAddress":"ecs.amazonaws.com",
    "userAgent":"ecs.amazonaws.com",
    "errorCode":"ThrottlingException",
    "errorMessage":"Rate exceeded",
    "eventID":"0aeb38fc-229b-4912-8b0d-2e8315193e9c"
  }
}
```

Quando queste chiamate API condividono i limiti con altro traffico API nel tuo account, potrebbero essere difficili da monitorare anche se vengono emesse come eventi di servizio.

Monitora la limitazione

È importante identificare quali richieste API sono limitate e chi le emette. Puoi utilizzare AWS CloudTrail Which per monitorare la limitazione e integrarsi con Amazon CloudWatch Athena e Amazon. EventBridge Puoi configurare l'invio di eventi specifici CloudTrail ai registri. CloudWatch CloudWatch Logs, log insights, analizza e analizza gli eventi. Questo identifica i dettagli negli eventi di limitazione, come l'utente o il ruolo IAM che ha effettuato la chiamata e il numero di chiamate API effettuate. Per ulteriori informazioni, consulta [Monitoraggio dei file di CloudTrail registro](#) con Logs. CloudWatch

Per ulteriori informazioni su CloudWatch Logs Insights e istruzioni su come interrogare i file di registro, vedi [Analisi dei dati di log con CloudWatch Logs Insights](#).

Con Amazon Athena, puoi creare query e analizzare dati utilizzando SQL standard. Ad esempio, puoi creare una tabella Athena per analizzare CloudTrail gli eventi. Per ulteriori informazioni, vedere [Utilizzo della CloudTrail console per creare una tabella Athena per i CloudTrail log](#).

Dopo aver creato una tabella Athena, è possibile utilizzare query SQL come la seguente per analizzare gli errori. `ThrottlingException`

Sostituiscili *user-input* con i tuoi valori.

```
select eventname, errorcode,eventsource,awsregion, useragent,COUNT(*) count
FROM cloudtrail_table-name
where errorcode = 'ThrottlingException'
AND eventtime between '2024-09-24T00:00:08Z' and '2024-09-23T23:15:08Z'
group by errorcode, awsregion, eventsource, useragent, eventname
order by count desc;
```

Amazon ECS invia anche notifiche di eventi ad Amazon. EventBridge Esistono eventi di modifica dello stato delle risorse ed eventi di azione del servizio. Includono eventi di limitazione delle API come `ECS_OPERATION_THROTTLED`. `SERVICE_DISCOVERY_OPERATION_THROTTLED` Per ulteriori informazioni, consulta [Eventi di azione del servizio Amazon ECS](#).

Questi eventi possono essere utilizzati da un servizio, ad esempio AWS Lambda per eseguire azioni in risposta. Per ulteriori informazioni, consulta [Gestione degli eventi Amazon ECS](#).

Se esegui attività autonome, alcune operazioni API, ad esempio, `RunTask` sono asincrone e le operazioni di riprova non vengono eseguite automaticamente. In questi casi, puoi utilizzare servizi come AWS Step Functions with EventBridge integration per riprovare le operazioni limitate o non riuscite. Per ulteriori informazioni, consulta [Manage a container task \(Amazon ECS, Amazon SNS\)](#).

CloudWatch Da utilizzare per monitorare la limitazione

CloudWatch offre il monitoraggio dell'utilizzo delle API **Usage** nello spazio dei nomi `By Resource`. AWS Queste metriche vengono registrate con il tipo di API e il nome della metrica. `CallCount` Puoi creare allarmi che si attivano ogni volta che queste metriche raggiungono una determinata soglia. Per ulteriori informazioni, consulta [Visualizzazione delle quote di servizio e impostazione degli allarmi](#).

CloudWatch offre anche il rilevamento delle anomalie. Questa funzionalità utilizza l'apprendimento automatico per analizzare e stabilire linee di base in base al comportamento particolare della metrica

su cui l'hai abilitata. In caso di attività insolite dell'API, puoi utilizzare questa funzionalità insieme agli allarmi. CloudWatch Per ulteriori informazioni, consulta [Utilizzo del rilevamento CloudWatch delle anomalie](#).

Monitorando in modo proattivo gli errori di limitazione, potete contattarci Supporto per aumentare i limiti di limitazione pertinenti e ricevere anche indicazioni per le vostre esigenze applicative specifiche.

Motivi di errore dell'API Amazon ECS

Quando un'azione API che hai attivato tramite l'API, la console o la console di Amazon ECS AWS CLI esce con un messaggio di `failures` errore, quanto segue può aiutarti a risolvere la causa. L'errore restituisce un motivo e l'Amazon Resource Name (ARN) della risorsa associata all'errore.

Molte risorse sono specifiche della regione, quindi accertati che la console sia impostata sulla regione corretta per le tue risorse. Quando usi il AWS CLI, assicurati che i AWS CLI comandi vengano inviati alla regione corretta con il `--region region` parametro.

Per ulteriori informazioni sulla struttura del tipo di dati `Failure`, consulta [Errore](#) nella Documentazione di riferimento dell'API di Amazon Elastic Container Service.

Di seguito sono riportati alcuni esempi di messaggi di errore che potreste ricevere durante l'esecuzione di comandi API.

Azione API	Motivo dell'errore o dell'interruzione	Causa
<code>DescribeClusters</code>	MISSING	Il cluster specificato non è stato trovato. Verifica l'ortografia del nome del cluster.
<code>DescribeInstances</code>	MISSING	L'istanza del container specificata non è stata trovata. Verifica che sia stato specificato il cluster in cui è registrata l'istanza di container e che l'ARN o l'ID dell'istanza di container sia corretto.

Azione API	Motivo dell'errore o dell'interruzione	Causa
DescribeServices	MISSING	Il servizio specificato non è stato trovato. Verifica che sia specificato il cluster o la regione corretta e che l'ARN del servizio o il nome sia valido.
DescribeTasks	MISSING	Il processo specificato non è stato trovato. Verifica che sia specificato il cluster o la regione corretta e che l'ARN del processo o il nome sia valido.

Azione API	Motivo dell'errore o dell'interruzione	Causa
DescribeTasks	TaskFailedToStart: RESOURCE:*	In caso di RESOURCE:CPU errori, il numero di CPUs richieste dall'attività non è disponibile nelle istanze del contenitore. Ciò si verifica in genere quando il requisito dell'unità CPU nella definizione dell'attività è maggiore della dimensione della CPU delle EC2 istanze Amazon definite nel gruppo Auto Scaling mappato al provider di capacità. È necessario verificare la configurazione del provider di capacità. Per gli errori RESOURCE: MEMORY indica che la quantità di memoria richiesta dall'attività non è disponibile sulle istanze di container. Ciò si verifica in genere quando la quantità di memoria richiesta nella definizione dell'attività è maggiore della memoria supportata nelle EC2 istanze Amazon definite nel gruppo Auto Scaling mappato al provider di capacità. È necessario verificare la configurazione del provider di capacità.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	<code>TaskFailedToStart: AGENT</code>	L'istanza di container su cui hai provato ad avviare un processo ha un agente attualmente disconnesso. Per prevenire lunghi tempi di attesa per il posizionamento delle attività, la richiesta è stata respinta. Per informazioni su come risolvere i problemi relativi a un agente disconnesso, consulta How do I troubleshoot a disconnected Amazon ECS agent? (Come posso risolvere i problemi di un agente Amazon ECS disconnesso).
	<code>TaskFailedToStart: MemberOf placement constraint unsatisfied</code>	Non esiste un'istanza di contenitore che soddisfi i vincoli di posizionamento definiti nella definizione dell'attività.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	<code>TaskFailedToStart: ATTRIBUTE</code>	La tua definizione di attività contiene un parametro che richiede un determinato attributo dell'istanza di container che non è disponibile nelle tue istanze di container. Ad esempio, se la tua attività utilizza la modalità di rete <code>awsvpc</code> , ma non sono presenti istanze nelle sottoreti da te specificate con l'attributo <code>ecs.capability.task-eni</code> . Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate e Configurazione dell'agente del container Amazon ECS .
	<code>TaskFailedToStart: NO ACTIVE INSTANCES</code>	Non ci sono istanze attive nel provider di capacità. Per informazioni su come gestire i gruppi di Auto Scaling, consulta i gruppi di Auto Scaling nella Amazon Auto Scaling User EC2 Guide .

Azione API	Motivo dell'errore o dell'interruzione	Causa
GetTaskProtection	TaskFailedToStart: EMPTY_CAPACITY_PROVIDER	Nel cluster non sono presenti istanze. Ciò è probabilmente dovuto a un provider di capacità vuoto o al fatto che le istanze del provider di capacità non sono registrate nel cluster. Per informazioni su come gestire i gruppi di Auto Scaling, consulta i gruppi di Auto Scaling nella Amazon Auto Scaling User EC2 Guide.
	MISSING	Il processo specificato non è stato trovato. Verifica che il nome o l'ARN del cluster e l'ARN o l'ID dell'attività siano validi.
	TASK_NOT_VALID	L'attività specificata non fa parte di un servizio Amazon ECS. Solo le attività gestite dai servizi Amazon ECS possono essere protette. Verifica l'ARN o l'ID dell'attività e riprova.

Azione API	Motivo dell'errore o dell'interruzione	Causa
RunTask o StartTask	RESOURCE : *	La risorsa o le risorse richieste dal processo non sono disponibili sull'istanza di container nel cluster. Se la risorsa corrisponde a CPU, memoria, porte o interfacce di rete elastiche, potrebbe essere necessario aggiungere e istanze di container al tuo cluster. Per gli errori RESOURCE : ENI , il tuo cluster non dispone di punti di collegamento per l'interfaccia di rete elastica che sono necessari per i processi che utilizzano la modalità di rete aws vpc. EC2 Le istanze Amazon hanno un limite al numero di interfacce di rete che possono essere collegate e l'interfaccia di rete principale conta come una. Per ulteriori informazioni su quante interfacce di rete sono supportate per ogni tipo di istanza, consulta Indirizzi IP per interfaccia di rete per tipo di istanza nella Amazon EC2 User Guide. In caso di RESOURCE : GPU errori, il numero di GPUs richieste dall'attività non

Azione API	Motivo dell'errore o dell'interruzione	Causa
		è disponibile e potrebbe essere necessario aggiungere istanze di container abilitate per GPU al cluster. Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU.
	AGENT	L'istanza di container su cui hai provato ad avviare un processo ha un agente attualmente disconnesso. Per prevenire lunghi tempi di attesa per il posizionamento delle attività, la richiesta è stata respinta. Per informazioni su come risolvere i problemi relativi a un agente disconnesso, consulta How do I troubleshoot a disconnected Amazon ECS agent? (Come posso risolvere i problemi di un agente Amazon ECS disconnesso).
	LOCATION	L'istanza di container su cui hai provato ad avviare un processo si trova in una zona di disponibilità diversa rispetto alle sottoreti specificate in <code>awsVpcConfiguration</code>.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	ATTRIBUTE	La tua definizione di attività contiene un parametro che richiede un determinato attributo dell'istanza di container che non è disponibile nelle tue istanze di container. Ad esempio, se la tua attività utilizza la modalità di rete <code>awsvpc</code> , ma non sono presenti istanze nelle sottoreti da te specificate con l'attributo <code>ecs.capability.task-eni</code> . Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS per il tipo di lancio Fargate e Configurazione dell'agente del container Amazon ECS .
StartTask	MISSING	L'istanza del contenitore su cui hai tentato di avviare l'attività non è stata trovata. Controlla se è stato specificato il cluster o la regione errati o se l'ARN o l'ID dell'istanza del contenitore è stato digitato in modo errato.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	INACTIVE	La registrazione dell'istanza di container su cui hai provato ad avviare un processo è stata precedentemente revocata con Amazon ECS quindi non può essere utilizzata.
StopServiceDeployment	ECS deployment failed	L'API è stata eseguita da un account fraudolento. StopServiceDeployment
TagResource	InvalidParameterException	L'ARN del servizio che stai taggando ha il formato breve. È necessario migrare al formato lungo. Per informazioni su come migrare l'ARN, vedere. Esegui la migrazione di un ARN di servizio breve di Amazon ECS a un ARN lungo
UpdateTaskProtection	DEPLOYMENT_BLOCKED	Impossibile impostare la protezione delle attività poiché una o più attività protette impediscono alla distribuzione del servizio di raggiungere uno stato stazionario. Annulla l'impostazione della protezione e delle attività sulle attività esistenti o attendi la scadenza della protezione.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	MISSING	Il processo specificato non è stato trovato. Verifica che il nome o l'ARN del cluster e l'ARN o l'ID dell'attività siano validi.
	TASK_NOT_VALID	L'attività specificata non fa parte di un servizio Amazon ECS. Solo le attività gestite dai servizi Amazon ECS possono essere protette. Verifica l'ARN o l'ID dell'attività e riprova.

 Note

Oltre agli scenari di errore descritti qui, le operazioni API possono fallire anche a causa di eccezioni, con conseguenti risposte di errore. Per un elenco di tali eccezioni, consulta [Errori comuni](#).

Sicurezza in Amazon Elastic Container Service

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei [programmi di conformità AWS](#). Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon Elastic Container Service, consulta [Servizi AWS coperti dal programma di conformità](#).
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon ECS. Gli argomenti seguenti descrivono come configurare Amazon ECS per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon ECS.

Argomenti

- [Identity and Access Management per Amazon Elastic Container Registry](#)
- [Registrazione e monitoraggio in Amazon Elastic Container Service](#)
- [Convalida della conformità per Amazon Elastic Container Service](#)
- [AWS Fargate Standard federale per l'elaborazione delle informazioni \(FIPS-140\)](#)
- [Sicurezza dell'infrastruttura in Amazon Elastic Container Service](#)
- [AWS modello di responsabilità condivisa per Amazon ECS](#)
- [Best practice di sicurezza di rete per Amazon ECS](#)
- [Best practice per la sicurezza di attività e container di Amazon ECS](#)

Identity and Access Management per Amazon Elastic Container Registry

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) a utilizzare risorse Amazon ECS. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come funziona Amazon Elastic Container Service con IAM](#)
- [Esempi di policy basate su identità per Amazon Elastic Container Service](#)
- [AWS politiche gestite per Amazon Elastic Container Service](#)
- [Uso di ruoli collegati ai servizi per Amazon ECS](#)
- [Ruoli IAM per Amazon ECS](#)
- [Autorizzazioni necessarie per la console Amazon ECS](#)
- [Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS](#)
- [Concessione dell'autorizzazione all'assegnazione di tag alle risorse al momento della creazione](#)
- [Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service](#)
- [Best practice IAM per Amazon ECS](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon ECS.

Utente del servizio: se utilizzi il servizio Amazon ECS per svolgere il tuo lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità di Amazon ECS utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni.

La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon ECS, consulta [Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service](#).

Amministratore dei servizi: se sei il responsabile delle risorse Amazon ECS presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon ECS. Il compito dell'utente è determinare le caratteristiche e le risorse di Amazon MQ a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon ECS, consulta [Come funziona Amazon Elastic Container Service con IAM](#).

Amministratore IAM: se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad Amazon ECS. Per visualizzare policy basate su identità di Amazon ECS di esempio che possono essere utilizzate in IAM, consulta [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sul metodo consigliato per la firma delle richieste, consulta [Signature Version 4 AWS per le richieste API](#) nella Guida per l'utente IAM.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\)AWS in IAM](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta [Cos'è IAM Identity Center?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con

utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, potresti avere un gruppo denominato IAMAdminse concedere a quel gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Per assumere temporaneamente un ruolo IAM in AWS Management Console, puoi [passare da un ruolo utente a un ruolo IAM \(console\)](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Create a role for a third-party identity provider \(federation\)](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center.
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.

- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è normale che quel servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso inoltrato (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).
- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un' EC2 istanza e che AWS CLI effettuano richieste AWS API. È preferibile archiviare le chiavi di accesso all'interno dell' EC2 istanza. Per assegnare un AWS ruolo a un' EC2 istanza e renderlo disponibile per tutte le sue applicazioni, create un profilo di istanza collegato all'istanza. Un profilo di istanza contiene il ruolo e consente ai programmi in esecuzione sull' EC2 istanza di ottenere credenziali temporanee. Per ulteriori

informazioni, consulta [Utilizzare un ruolo IAM per concedere le autorizzazioni alle applicazioni in esecuzione su EC2 istanze Amazon](#) nella IAM User Guide.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su

come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Elenchi di controllo degli accessi (ACLs)

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano. AWS WAF ACLs Per ulteriori informazioni ACLs, consulta la [panoramica della lista di controllo degli accessi \(ACL\)](#) nella Amazon Simple Storage Service Developer Guide.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.

- **Politiche di controllo del servizio (SCPs):** SCPs sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più di proprietà dell'Account AWS azienda. Se abiliti tutte le funzionalità di un'organizzazione, puoi applicare le politiche di controllo del servizio (SCPs) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità presenti negli account dei membri, inclusa ciascuna di esse. Utente root dell'account AWS. Per ulteriori informazioni su Organizations and SCPs, consulta [le politiche di controllo dei servizi](#) nella Guida AWS Organizations per l'utente.
- **Politiche di controllo delle risorse (RCPs):** RCPs sono politiche JSON che puoi utilizzare per impostare le autorizzazioni massime disponibili per le risorse nei tuoi account senza aggiornare le politiche IAM allegate a ciascuna risorsa di tua proprietà. L'RCP limita le autorizzazioni per le risorse negli account dei membri e può influire sulle autorizzazioni effettive per le identità, incluse le Utente root dell'account AWS, indipendentemente dal fatto che appartengano o meno all'organizzazione. Per ulteriori informazioni su Organizations e RCPs, incluso un elenco di Servizi AWS tale supporto RCPs, vedere [Resource control policies \(RCPs\)](#) nella Guida per l'AWS Organizations utente.
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta la [logica di valutazione delle policy](#) nella IAM User Guide.

Come funziona Amazon Elastic Container Service con IAM

Prima di utilizzare IAM per gestire l'accesso ad Amazon ECS, scopri quali funzioni IAM sono disponibili per l'uso con Amazon ECS.

Funzionalità IAM	Supporto di Amazon ECS
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Parziale
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come Amazon ECS e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta [AWS i servizi che funzionano con IAM nella IAM User Guide](#).

Policy basate su identità per Amazon ECS

Supporta le policy basate su identità: sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte.

Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta [Guida di riferimento agli elementi delle policy JSON IAM](#) nella Guida per l'utente di IAM.

Esempi di policy basate su identità per Amazon ECS

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

Policy basate su risorse all'interno di Amazon ECS

Supporta le policy basate su risorse: no

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Esempi di policy basate sulle risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le operazioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Operazioni di policy per Amazon ECS

Supporta le operazioni di policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le operazioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di operazioni di Amazon ECS, consulta [Operazioni definite da Amazon Elastic Container Service](#) nella documentazione di riferimento alla autorizzazione di servizio.

Le operazioni delle policy in Amazon ECS utilizzano il seguente prefisso prima dell'operazione:

```
ecs
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [  
    "ecs:action1",  
    "ecs:action2"  
]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "ecs:Describe*"
```

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

Risorse di policy per Amazon ECS

Supporta le risorse politiche: parziale

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). È possibile eseguire questa operazione per operazioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*" 
```

Per visualizzare un elenco dei tipi di risorse Amazon ECS e relativi ARNs, consulta [Resources defined by Amazon Elastic Container Service nel Service Authorization Reference](#). Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta [Operazioni definite da Amazon Elastic Container Service](#).

Alcune operazioni API di Amazon ECS supportano più risorse. Ad esempio, è possibile selezionare più cluster quando si chiama l'operazione API `DescribeClusters`. Per specificare più risorse in una singola istruzione, separale ARNs con virgole.

```
"Resource": [
    "EXAMPLE-RESOURCE-1",
    "EXAMPLE-RESOURCE-2" ]
```

Ad esempio, la risorsa del cluster Amazon ECS dispone del seguente ARN:

```
arn:${Partition}:ecs:${Region}:${Account}:cluster/${clusterName}
```

Per specificare `my-cluster-1` e `my-cluster-2` raggruppare nella tua dichiarazione, usa quanto segue: ARNs

```
"Resource": [
    "arn:aws:ecs:us-east-1:123456789012:cluster/my-cluster-1",
    "arn:aws:ecs:us-east-1:123456789012:cluster/my-cluster-2" ]
```

Per specificare tutti i cluster che appartengono ad un account specifico, utilizza il carattere jolly (*):


```
"Resource": "arn:aws:ecs:us-east-1:123456789012:cluster/*"
```

Per le definizioni di attività, puoi specificare la revisione più recente o una revisione specifica.

Per specificare tutte le revisioni della definizione dell'attività, usa il carattere jolly (*):

```
"Resource:arn:${Partition}:ecs:${Region}:${Account}:task-definition/  
${TaskDefinitionFamilyName}:*"
```

Per specificare una revisione specifica della definizione dell'attività, usa \$ {}:

TaskDefinitionRevisionNumber

```
"Resource:arn:${Partition}:ecs:${Region}:${Account}:task-definition/  
${TaskDefinitionFamilyName}:${TaskDefinitionRevisionNumber}"
```

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

Chiavi di condizione delle policy per Amazon ECS

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento `Condition` (o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition` è facoltativo. È possibile compilare espressioni condizionali che utilizzano [operatori di condizione](#), ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition` in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione AND logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. OR Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

È possibile anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, è possibile autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo

nome utente IAM. Per ulteriori informazioni, consulta [Elementi delle policy IAM: variabili e tag](#) nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di [contesto delle condizioni AWS globali nella Guida](#) per l'utente IAM.

Amazon ECS supporta le seguenti chiavi di condizione specifiche del servizio, utilizzabili per fornire filtri granulari per le tue policy IAM:

Chiave di condizione	Descrizione	Tipi di valutazione
aws: RequestTag /\$ {} TagKey	La chiave di contesto è formattata "aws : RequestTag/ <i>tag-key</i>": "<i>tag-value</i> " dove <i>tag-key</i> e è una <i>tag-value</i> coppia di tag chiave-valore. Verifica che la coppia chiave-valore del tag sia presente in una richiesta. AWS Ad esempio, è possibile controllare che la richiesta includa la chiave di un tag "Dept" e che abbia il valore "Accounting" .	Stringa
aws: ResourceTag /\$ {} TagKey	La chiave di contesto è formattata "aws : ResourceTag/ <i>tag-key</i>": "<i>tag-value</i> " dove <i>tag-key</i> e è una <i>tag-value</i> coppia di tag chiave-valore. Controlla che il tag collegato alla risorsa dell'identità (utente o ruolo) corrisponda al nome e al valore della chiave specificata.	Stringa
leggi: TagKeys	Questa chiave di contesto è formattata "aws : TagKeys": " <i>tag-key</i> " dove <i>tag-key</i> è presente un elenco di chiavi di tag senza valori (ad esempio, ["Dept", "Cost-Center"]). Controlla le chiavi dei tag presenti in una AWS richiesta.	Stringa

Chiave di condizione	Descrizione	Tipi di valutazione
ecs: ResourceTag /\$ {} TagKey	La chiave di contesto è formattata "ecs:ResourceTag/ <i>tag-key</i>": "<i>tag-value</i> " dove <i>tag-key</i> e è una coppia <i>tag-value</i> di tag chiave-valore. Controlla che il tag collegato alla risorsa dell'identità (utente o ruolo) corrisponda al nome e al valore della chiave specificata.	Stringa
ecs:account-setting	La chiave di contesto è formattata "ecs:account-setting": " <i>account-setting</i> " dove si <i>account-setting</i> trova il nome dell'impostazione dell'account.	Stringa
Decs: auto-assign-public-ip	La chiave di contesto è formattata "ecs:auto-assign-public-ip": " <i>value</i> " dove <i>value</i> - è "" o true "»false.	Stringa
ecs: fornitore di capacità	La chiave di contesto è formattata "ecs:capacity-provider": " <i>capacity-provider-arn</i> " dove si <i>capacity-provider-arn</i> trova l'ARN per il provider di capacità.	ARN, Null
ecs:cluster	La chiave di contesto è formattata "ecs:cluster": " <i>cluster-arn</i> " dove si <i>cluster-arn</i> trova l'ARN per il cluster Amazon ECS.	ARN, Null
ecs: compatibilità informatica	La chiave di contesto è formattata "ecs:compute-compatibility": " <i>compatibility-type</i> " dove <i>compatibility-type</i> è "FARGATE«,"o"». EC2 EXTERNAL	Stringa
ecs:container-instances	La chiave di contesto è formattata "ecs:container-instances": " <i>container-instance-arns</i> " dove <i>container-instance-arns</i> si trovano una o più istanze del contenitore. ARNs	ARN, Null

Chiave di condizione	Descrizione	Tipi di valutazione
ecs:container-name	La chiave di contesto è formattata "ecs:container-name":" <i>container-name</i> " dove <i>container-instance-</i> è il nome di un contenitore Amazon ECS definito nella definizione dell'attività.	Stringa
ecs:enable-execute-command	La chiave di contesto è formattata "ecs:enable-execute-command":" <i>value</i> " dove <i>value-</i> è "" o true "»false.	Stringa
ecs:enable-service-connect	La chiave di contesto è formattata "ecs:enable-service-connect":" <i>value</i> " dove <i>value</i> è "true" o. "false"	Stringa
ecs:enable-ecs-volumes	La chiave di contesto è formattata "ecs:enable-ecs-volumes":" <i>value</i> " dove <i>value</i> è "true" o. "false"	Stringa
ecs:enable-managed-tags	La chiave di contesto è formattata "ecs:enable-managed-tags":" <i>value</i> " dove <i>value</i> è "true" o. "false"	Stringa
ecs:enable-vpc-lattice	La chiave di contesto è formattata "ecs:vpc-lattice":" <i>value</i> " dove <i>value</i> è "true" o. "false"	Stringa
ecs:-key-fargate-ephemeral-storage-kms	La chiave di contesto è formattata "ecs:fargate-ephemeral-storage-kms-key":" <i>key</i> " dove si <i>key</i> trova l'ID della chiave. AWS KMS	Stringa
ecs:namespace	La chiave di contesto è formattata "ecs:namespace":" <i>namespace-arn</i> " dove si <i>namespace-arn</i> trova l'ARN per AWS Cloud Map lo spazio dei nomi.	ARN, Null

Chiave di condizione	Descrizione	Tipi di valutazione
ecs:propagate-tags	La chiave di contesto è formattata "ecs:propagate-tags": " <i>value</i> " dove è, o. <i>value</i> "TASK_DEFINITION" "SERVICE" "NONE"	Stringa
ecs:service	La chiave di contesto è formattata "ecs:service": " <i>service-arn</i> " dove si <i>service-arn</i> trova l'ARN per il servizio Amazon ECS.	ARN, Null
ecs:task-definition	La chiave di contesto è formattata "ecs:task-definition": " <i>task-definition-arn</i> " dove si <i>task-definition-arn</i> trova l'ARN per la definizione del task di Amazon ECS.	ARN, Null
ecs:subnet	La chiave di contesto è formattata "ecs:subnet": " <i>subnets</i> " dove si <i>subnets</i> trovano le sottoreti di un'attività o di un servizio che utilizza la modalità. awsvpc	Stringa
ecs:task	La chiave di contesto è formattata "ecs:task": " <i>task-arn</i> " dove si <i>task-arn</i> trova l'ARN per il servizio Amazon ECS.	ARN, Null
ecs:task-cpu	La chiave di contesto è formattata "ecs:task-cpu": " <i>task-cpu</i> " dove si <i>task-cpu</i> trova l'attività cpu, come numero intero con 1024 = 1 vCPU.	Numero intero
ecs:task-memory	La chiave di contesto è formattata "ecs:task-memory": " <i>task-memory</i> " dove si <i>task-memory</i> trova la memoria delle attività in MiB.	Numero intero

Per un elenco completo delle chiavi di condizione di Amazon ECS, consulta [Chiavi di condizione per Amazon Elastic Container Service](#) nella documentazione di riferimento alla autorizzazione di servizio. Per informazioni su operazioni e risorse con cui è possibile utilizzare una chiave di condizione, consulta [Operazioni definite da Amazon Elastic Container Service](#).

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

Elenchi di controllo degli accessi (ACLs) in Amazon ECS

Supporti ACLs: no

Le liste di controllo degli accessi (ACLs) controllano quali principali (membri dell'account, utenti o ruoli) dispongono delle autorizzazioni per accedere a una risorsa. ACLs sono simili alle politiche basate sulle risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato su attributi (ABAC) con Amazon ECS

Important

Amazon ECS supporta il controllo degli accessi basato su attributi per tutte le risorse Amazon ECS. Per determinare se puoi utilizzare gli attributi per definire l'ambito di un'azione, utilizza la tabella [Operazioni definite da Amazon ECS](#) in Service Authorization Reference. Verifica innanzitutto che vi sia una risorsa nella colonna Risorsa. Quindi, utilizza la colonna Chiavi di condizione per vedere le chiavi per la combinazione operazione/risorsa.

Supporta ABAC (tag nelle policy): sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'[elemento condizione](#) di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente IAM. Per visualizzare un tutorial con le fasi per l'impostazione di ABAC, consulta [Utilizzo del controllo degli accessi basato su attributi \(ABAC\)](#) nella Guida per l'utente di IAM.

Per ulteriori informazioni sull'aggiunta di tag alle risorse di Amazon ECS, consulta [Taggare le risorse Amazon ECS](#).

Per visualizzare una policy basata sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta [Descrizione dei servizi Amazon ECS basati su tag](#).

Utilizzo di credenziali temporanee con Amazon ECS

Supporta le credenziali temporanee: sì

Alcune Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla [Servizi AWS compatibilità con IAM nella IAM](#) User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta [Passaggio da un ruolo utente a un ruolo IAM \(console\)](#) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza provvisorie in IAM](#).

Sessioni di accesso diretto per Amazon ECS

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni

con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta [Forward access sessions](#).

Ruoli di servizio per Amazon ECS

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Create a role to delegate permissions to an Servizio AWS](#) nella Guida per l'utente IAM.

Warning

La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere la funzionalità di Amazon ECS. Modifica i ruoli del servizio solo quando Amazon ECS fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per Amazon ECS

Supporta ruoli collegati ai servizi: Sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per maggiori dettagli su come creare e gestire i ruoli collegati ai servizi Amazon ECS, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).

Esempi di policy basate su identità per Amazon Elastic Container Service

Di default, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse di Amazon ECS. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o l'AWS API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy IAM \(console\)](#) nella Guida per l'utente IAM.

Per dettagli sulle azioni e sui tipi di risorse definiti da Amazon ECS, incluso il formato di ARNs per ogni tipo di risorsa, consulta [Azioni, risorse e chiavi di condizione per Amazon Elastic Container Service nel Service](#) Authorization Reference.

Argomenti

- [Best practice relative alle policy di Amazon ECS](#)
- [Consenti agli utenti di Amazon ECS di visualizzare le proprie autorizzazioni](#)
- [Esempi di cluster Amazon ECS](#)
- [Esempi di istanze di container Amazon ECS](#)
- [Esempi di definizione delle attività in Amazon ECS](#)
- [Esempio di attività di esecuzione di Amazon ECS](#)
- [Esempio di attività di avvio di Amazon ECS](#)
- [Elenca e descrivi esempi di attività di Amazon ECS](#)
- [Crea un esempio di servizio Amazon ECS](#)
- [Esempio di aggiornamento del servizio Amazon ECS](#)
- [Descrizione dei servizi Amazon ECS basati su tag](#)
- [Esempio di sostituzione dello spazio dei nomi di Amazon ECS Service Connect Deny](#)

Best practice relative alle policy di Amazon ECS

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon ECS nell'account. Queste azioni possono comportare costi aggiuntivi per l'Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo Account AWS. Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni

che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente IAM.

- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse è possibile aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Protezione dell'accesso API con MFA](#) nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Consenti agli utenti di Amazon ECS di visualizzare le proprie autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
```

```

        "iam:GetUserPolicy",
        "iam:ListGroupsForUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
    ],
    "Resource": ["arn:aws:iam::*:user/${aws:username}"]
},
{
    "Sid": "NavigateInConsole",
    "Effect": "Allow",
    "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
}

```

Esempi di cluster Amazon ECS

La seguente policy IAM concede le autorizzazioni per creare ed elencare i cluster. Le operazioni `CreateCluster` e `ListClusters` non accettano risorse, pertanto la definizione delle risorse è impostata su `*` per tutte le risorse.

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:ListClusters"
            ],
            "Resource": ["*"]
        }
    ]
}

```

```
]
}
```

La seguente policy IAM concede le autorizzazioni per descrivere ed eliminare un determinato cluster. Le DeleteCluster azioni DescribeClusters e accettano il cluster ARNs come risorse.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs>DeleteCluster"
      ],
      "Resource": ["arn:aws:ecs:us-east-1:<aws_account_id>:cluster/
<cluster_name>"]
    }
  ]
}
```

La seguente policy IAM può essere collegata a un utente o a un gruppo per permettere solo a tale utente o gruppo di eseguire operazioni su un determinato cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ecs:Describe*",
        "ecs:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    },
    {
      "Action": [
        "ecs>DeleteCluster",
        "ecs:DeregisterContainerInstance",
        "ecs:ListContainerInstances",
        "ecs:RegisterContainerInstance",
        "ecs:SubmitContainerStateChange",
        "ecs:SubmitTaskStateChange"
      ]
    }
  ]
}
```

```

    ],
    "Effect": "Allow",
    "Resource": "arn:aws:ecs:us-east-1:<aws_account_id>:cluster/default"
  },
  {
    "Action": [
      "ecs:DescribeContainerInstances",
      "ecs:DescribeTasks",
      "ecs:ListTasks",
      "ecs:UpdateContainerAgent",
      "ecs:StartTask",
      "ecs:StopTask",
      "ecs:RunTask"
    ],
    "Effect": "Allow",
    "Resource": "*",
    "Condition": {
      "ArnEquals": {"ecs:cluster": "arn:aws:ecs:us-
east-1:<aws_account_id>:cluster/default"}
    }
  }
]
}

```

Esempi di istanze di container Amazon ECS

La registrazione delle istanze di container viene gestita dall'agente Amazon ECS ma, in alcuni casi, puoi permettere a un utente di revocare manualmente la registrazione di un'istanza in un cluster. Ad esempio, se l'istanza di container è stata accidentalmente registrata nel cluster sbagliato o se l'istanza è stata interrotta con attività ancora in esecuzione.

La seguente policy IAM permette a un utente di elencare e revocare la registrazione delle istanze di container in un determinato cluster:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeregisterContainerInstance",
        "ecs:ListContainerInstances"
      ],

```

```

        "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>"]
    }
}

```

La seguente policy IAM permette a un utente di descrivere una specifica istanza di container in un determinato cluster. Per estendere questa autorizzazione a tutte le istanze di container in un cluster, puoi sostituire l'UUID dell'istanza di container con `*`.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:DescribeContainerInstances"],
      "Condition": {
        "ArnEquals": {"ecs:cluster":
"arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
      },
      "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:container-instance/
<cluster_name>/<container_instance_UUID>"]
    }
  ]
}

```

Esempi di definizione delle attività in Amazon ECS

Le policy IAM delle definizioni di attività non supportano le autorizzazioni a livello di risorsa, ma le seguenti policy IAM permettono a un utente di registrare, elencare e descrivere le definizioni di attività:

Se utilizzi la console, devi aggiungere `CloudFormation: CreateStack` come Action.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:RegisterTaskDefinition",
        "ecs:ListTaskDefinitions",

```

```

        "ecs:DescribeTaskDefinition"
      ],
      "Resource": ["*"]
    }
  ]
}

```

Esempio di attività di esecuzione di Amazon ECS

Le risorse per `RunTask` sono definizioni di attività. Per limitare i cluster su cui un utente può eseguire le definizioni di attività, puoi specificarli nel blocco `Condition`. Il vantaggio è che non dovrai elencare sia le definizioni di attività che i cluster nelle tue risorse per consentire un accesso appropriato. Puoi utilizzarne uno o entrambi.

La seguente policy IAM concede l'autorizzazione per eseguire qualsiasi revisione di una specifica definizione di attività in un determinato cluster:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:RunTask"],
      "Condition": {
        "ArnEquals": {"ecs:cluster":
"arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
      },
      "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task-definition/
<task_family>:*"]
    }
  ]
}

```

Esempio di attività di avvio di Amazon ECS

Le risorse per `StartTask` sono definizioni di attività. Per limitare i cluster e le istanze di container su cui un utente può avviare le definizioni di attività, puoi specificarli nel blocco `Condition`. Il vantaggio è che non dovrai elencare sia le definizioni di attività che i cluster nelle tue risorse per consentire un accesso appropriato. Puoi utilizzarne uno o entrambi.

La seguente policy IAM concede l'autorizzazione per avviare qualsiasi revisione di una specifica definizione di attività in un cluster o un'istanza di container determinati.

Note

Per questo esempio, quando chiami l'StartTaskAPI con il AWS CLI o un altro AWS SDK, devi specificare la revisione della definizione dell'attività in modo che la Resource mappatura corrisponda.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:StartTask"],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>",
          "ecs:container-instances":
            ["arn:aws:ecs:<region>:<aws_account_id>:container-instance/<cluster_name>/
<container_instance_UUID>"]
        }
      },
      "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task-definition/
<task_family>:*"]
    }
  ]
}
```

Elenca e descrivi esempi di attività di Amazon ECS

La seguente policy IAM permette a un utente di elencare i processi per un determinato cluster:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:ListTasks"],
      "Condition": {
        "ArnEquals": {"ecs:cluster":
          "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
      }
    }
  ]
}
```



```

        },
        "Resource": ["*"]
    }
]
}

```

La seguente policy IAM permette a un utente di descrivere uno specifico processo in un determinato cluster:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:DescribeTasks"],
      "Condition": {
        "ArnEquals": {"ecs:cluster":
"arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
      },
      "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task/<cluster_name>/<task_UUID>"]
    }
  ]
}

```

Crea un esempio di servizio Amazon ECS

La seguente policy IAM permette a un utente di creare servizi Amazon ECS nella AWS Management Console:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:Describe*",
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling:RegisterScalableTarget",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "ecs:List*",

```

```

        "ecs:Describe*",
        "ecs:CreateService",
        "elasticloadbalancing:Describe*",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:ListRoles",
        "iam:ListGroups",
        "iam:ListUsers"
    ],
    "Resource": ["*"]
}
]
}

```

Esempio di aggiornamento del servizio Amazon ECS

La seguente policy IAM permette a un utente di aggiornare i servizi Amazon ECS nella AWS Management Console:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "application-autoscaling:Describe*",
        "application-autoscaling:PutScalingPolicy",
        "application-autoscaling>DeleteScalingPolicy",
        "application-autoscaling:RegisterScalableTarget",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "ecs:List*",
        "ecs:Describe*",
        "ecs:UpdateService",
        "iam:GetPolicy",
        "iam:GetPolicyVersion",
        "iam:GetRole",
        "iam:ListAttachedRolePolicies",
        "iam:ListRoles",
        "iam:ListGroups",
        "iam:ListUsers"
      ]
    }
  ]
}

```

```

    ],
    "Resource": ["*"]
  }
]
}

```

Descrizione dei servizi Amazon ECS basati su tag

Puoi utilizzare condizioni nella policy basata su identità per controllare l'accesso alle risorse Amazon ECS in base ai tag. Questo esempio illustra il modo in cui è possibile creare una policy che consente di descrivere i servizi. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag del servizio `Owner` è quello del nome utente dell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DescribeServices",
      "Effect": "Allow",
      "Action": "ecs:DescribeServices",
      "Resource": "*"
    },
    {
      "Sid": "ViewServiceIfOwner",
      "Effect": "Allow",
      "Action": "ecs:DescribeServices",
      "Resource": "arn:aws:ecs:*:*:service/*",
      "Condition": {
        "StringEquals": {"ecs:ResourceTag/Owner": "${aws:username}"}
      }
    }
  ]
}

```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard-roe` prova a descrivere un servizio Amazon ECS, il servizio deve avere il tag `Owner=richard-roe` o `owner=richard-roe`. In caso contrario l'accesso è negato. La chiave di tag di condizione `Owner` corrisponde a `Owner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

Esempio di sostituzione dello spazio dei nomi di Amazon ECS Service Connect Deny

La seguente policy IAM impedisce a un utente di sovrascrivere lo spazio dei nomi predefinito di Service Connect in una configurazione di servizio. Lo spazio dei nomi predefinito è impostato nel cluster. Tuttavia, puoi sovrascrivere il parametro in una configurazione del servizio. Per coerenza, valuta la possibilità di impostare tutti i nuovi servizi in modo che utilizzino lo stesso spazio dei nomi. Utilizza le seguenti chiavi di contesto per richiedere ai servizi di utilizzare uno spazio dei nomi specifico. Nel seguente esempio, sostituisci `<region>`, `<aws_account_id>`, `<cluster_name>` e `<namespace_id>` con i tuoi valori.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateService",
        "ecs:UpdateService"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>",
          "ecs:namespace":
            "arn:aws:servicediscovery:<region>:<aws_account_id>:namespace/<namespace_id>"
        }
      },
      "Resource": "*"
    }
  ]
}
```

AWS politiche gestite per Amazon Elastic Container Service

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite piuttosto che scrivere le politiche autonomamente. Creare [policy gestite dal cliente IAM](#) per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso

comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le [policy AWS gestite](#) nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi aggiungono occasionalmente autorizzazioni aggiuntive a una policy AWS gestita per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una politica AWS gestita quando viene lanciata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy ReadOnlyAccess AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione [Policy gestite da AWS per funzioni di processi](#) nella Guida per l'utente di IAM.

Amazon ECS e Amazon ECR forniscono diverse politiche gestite e relazioni di fiducia che puoi associare a utenti, gruppi, ruoli, EC2 istanze Amazon e attività di Amazon ECS che consentono diversi livelli di controllo sulle risorse e sulle operazioni API. Puoi applicare queste policy direttamente oppure utilizzarle come punto di partenza per la creazione di tue policy. Per ulteriori informazioni sulle policy gestite da Amazon ECR, consulta [Policy gestite da Amazon ECR](#).

Amazon ECS_FullAccess

È possibile allegare la policy AmazonECS_FullAccess alle identità IAM. Questa policy concede l'accesso amministrativo alle risorse di Amazon ECS e concede a un'identità IAM (ad esempio un utente, un gruppo o un ruolo) l'accesso ai servizi con AWS cui è integrato Amazon ECS per utilizzare tutte le funzionalità di Amazon ECS. L'utilizzo di questa policy consente di accedere a tutte le funzioni Amazon ECS disponibili nella AWS Management Console.

Per visualizzare le autorizzazioni per questa politica, consulta [FullAccessAmazonECS](#) nel Managed Policy Reference.AWS

Amazon ECSInfrastructure RolePolicyForVolumes

Puoi allegare la policy AmazonECSInfrastructureRolePolicyForVolumes gestita alle tue entità IAM.

La policy concede le autorizzazioni necessarie ad Amazon ECS per effettuare chiamate AWS API per tuo conto. Puoi collegare questa policy al ruolo IAM che fornisci con la configurazione del volume quando avvii attività e servizi Amazon ECS. Il ruolo consente ad Amazon ECS di gestire i volumi collegati alle tue attività. Per ulteriori informazioni, consulta il [ruolo IAM dell'infrastruttura Amazon ECS](#).

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon EC2Infrastructure RolePolicyForVolumes](#) nel AWS Managed Policy Reference.

EC2ContainerServiceforEC2Ruolo di Amazon

È possibile allegare la policy `AmazonEC2ContainerServiceforEC2Role` alle identità IAM. Questa politica concede autorizzazioni amministrative che consentono alle istanze di container Amazon ECS di effettuare chiamate per AWS tuo conto. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).

Amazon ECS associa questa politica a un ruolo di servizio che consente ad Amazon ECS di eseguire azioni per tuo conto contro istanze Amazon o EC2 istanze esterne.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon EC2 ContainerServicefor EC2 Role](#) nel AWS Managed Policy Reference.

Considerazioni

È opportuno considerare i seguenti suggerimenti e considerazioni quando usi la policy IAM gestita da `AmazonEC2ContainerServiceforEC2Role`.

- Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, è possibile modificare la policy gestita da `AmazonEC2ContainerServiceforEC2Role` per soddisfare le esigenze specifiche. Se una qualsiasi delle autorizzazioni concesse nella policy gestita non è necessaria per il caso d'uso, crea una policy personalizzata e aggiungi solo le autorizzazioni richieste. Ad esempio, l'autorizzazione `UpdateContainerInstancesState` è fornita per lo svuotamento dell'istanza Spot. Se tale autorizzazione non è necessaria per il tuo caso d'uso, escludila utilizzando una policy personalizzata.
- I container in esecuzione sulle tue istanze di container hanno accesso a tutte le autorizzazioni che vengono fornite al ruolo dell'istanza di container tramite i [metadati dell'istanza](#). Consigliamo di limitare le autorizzazioni nel ruolo dell'istanza di container all'elenco minimo delle autorizzazioni fornito nella policy `AmazonEC2ContainerServiceforEC2Role` gestita. Se i container nei tuoi processi hanno bisogno di autorizzazioni aggiuntive non elencate di seguito, ti consigliamo di

fornire a tali processi i relativi ruoli IAM. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

È possibile impedire ai contenitori nel bridge docker0 l'accesso alle autorizzazioni fornite al ruolo dell'istanza di container. È possibile farlo pur concedendo le autorizzazioni fornite da [Ruolo IAM dell'attività Amazon ECS](#) eseguendo il seguente comando iptables sulle istanze del container. I container non possono eseguire query sui metadati dell'istanza con questa regola in vigore. Questo comando presuppone la configurazione del bridge Docker di default e non funziona per i container che utilizzano la modalità di rete host. Per ulteriori informazioni, consulta [Modalità di rete](#).

```
sudo yum install -y iptables-services; sudo iptables --insert DOCKER USER 1 --in-interface docker+ --destination 169.254.169.254/32 --jump DROP
```

Per fare in modo che la regola iptables venga conservata dopo un riavvio, devi salvarla sull'istanza di container. Per l'AMI ottimizzata per Amazon ECS, utilizza il comando riportato di seguito. Per gli altri sistemi operativi, consulta la relativa documentazione specifica.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now iptables
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS:

```
sudo service iptables save
```

Amazon EC2 ContainerServiceEventsRole

È possibile allegare la policy AmazonEC2ContainerServiceEventsRole alle identità IAM. Questa politica concede autorizzazioni che consentono ad Amazon EventBridge (precedentemente CloudWatch Events) di eseguire attività per tuo conto. Questa policy può essere associata al ruolo IAM specificato quando si creano processi pianificati. Per ulteriori informazioni, consulta [Ruolo EventBridge IAM di Amazon ECS](#).

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon EC2 ContainerServiceEventsRole](#) nel AWS Managed Policy Reference.

Amazon ECSTask ExecutionRolePolicy

La policy IAM AmazonECSTaskExecutionRolePolicy gestita concede le autorizzazioni necessarie all'agente container Amazon ECS e agli agenti AWS Fargate container per effettuare chiamate AWS API per tuo conto. Questa policy può essere aggiunta al ruolo IAM di esecuzione dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSTask ExecutionRolePolicy](#) nel AWS Managed Policy Reference.

Amazon ECSService RolePolicy

La policy IAM gestita AmazonECSServiceRolePolicy consente ad Amazon Elastic Container Service di gestire il cluster. Questa policy può essere aggiunta al ruolo IAM di esecuzione dell'attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSService RolePolicy](#) nel AWS Managed Policy Reference.

AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity

Puoi collegare la policy

AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity anche alle tue entità IAM. Questa policy concede l'accesso amministrativo a AWS Private Certificate Authority Secrets Manager e ad altri AWS servizi necessari per gestire le funzionalità TLS di Amazon ECS Service Connect per tuo conto.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructure RolePolicyForServiceConnectTransportLayerSecurity](#) nel AWS Managed Policy Reference.

AWSApplicationAutoscalingECSServicePolicy

Non è possibile collegare AWSApplicationAutoscalingECSServicePolicy alle entità IAM. Questa policy è collegata a un ruolo collegato ai servizi che consente ad Application Auto Scaling di eseguire operazioni per tuo conto. Per ulteriori informazioni, consulta [Ruoli collegati ai servizi per Application Auto Scaling](#).

Per visualizzare le autorizzazioni per questa politica, consulta [AWSApplicationAutoscaling Policy nel Managed ECSService Policy Reference](#).AWS

AWSCodeDeployRoleForECS

Non è possibile collegare `AWSCodeDeployRoleForECS` alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente di eseguire azioni CodeDeploy per conto dell'utente. Per ulteriori informazioni, consulta [Creare un ruolo di servizio CodeDeploy nella Guida per l'AWS CodeDeploy utente](#).

Per visualizzare le autorizzazioni per questa politica, consulta [AWSCodeDeployRoleForECS](#) nel AWS Managed Policy Reference.

AWSCodeDeployRoleForECSLimited

Non è possibile collegare `AWSCodeDeployRoleForECSLimited` alle entità IAM. Questa policy è associata a un ruolo collegato al servizio che consente di eseguire azioni CodeDeploy per tuo conto. Per ulteriori informazioni, consulta [Creare un ruolo di servizio CodeDeploy nella Guida per l'AWS CodeDeploy utente](#).

Per vedere le autorizzazioni per questa policy, consulta [AWSCodeDeployRoleForECSLimited](#) nella Guida di riferimento sulle policy gestite da AWS .

AmazonECSInfrastructureRolePolicyForVpcLattice

Puoi collegare la policy `AmazonECSInfrastructureRolePolicyForVpcLattice` anche alle tue entità IAM. Questa policy fornisce l'accesso ad altre risorse di AWS servizio necessarie per gestire la funzionalità VPC Lattice nei carichi di lavoro Amazon ECS per tuo conto.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructureRolePolicyForVpcLattice](#) nel AWS Managed Policy Reference.

Fornisce l'accesso ad altre risorse AWS di servizio necessarie per gestire la funzionalità VPC Lattice nei carichi di lavoro Amazon ECS per tuo conto.

Aggiornamenti Amazon ECS alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle politiche AWS gestite per Amazon ECS da quando questo servizio ha iniziato a tracciare queste modifiche. Per gli avvisi automatici sulle modifiche apportate a questa pagina, sottoscrivi il feed RSS sulla pagina della cronologia dei documenti di Amazon ECS.

Modifica	Descrizione	Data
Aggiungi un nuovo Amazon ECSInfrastructure RolePolicyForVpcLattice	Fornisce l'accesso ad altre risorse AWS di servizio necessarie per gestire la funzionalità VPC Lattice nei carichi di lavoro Amazon ECS per tuo conto.	18 novembre 2024
Aggiungi autorizzazioni ad Amazon ECSInfrastructure RolePolicyForVolumes	La AmazonECSInfrastructureRolePolicyForVolumes policy è stata aggiornata per consentire ai clienti di creare un volume Amazon EBS da uno snapshot.	10 ottobre 2024
Autorizzazioni aggiunte a the section called “Amazon ECS_FullAccess”	La AmazonECS_FullAccess policy è stata aggiornata per aggiungere iam:PassRole alle autorizzazioni per i ruoli IAM per un ruolo denominato. ecsInfras tructureRole Questo è il ruolo IAM predefinito creato da AWS Management Console che deve essere utilizzato come ruolo dell'infrastruttura ECS che consente ad Amazon ECS di gestire i volumi Amazon EBS collegati alle attività ECS.	13 agosto 2024
Aggiungi una nuova ECSInfras tructure RolePolicyForServiceConnectTransport LayerSecurity politica Amazon	È stata aggiunta una nuova ECSInfrastructure RolePolicyForServiceConnectTransport LayerSecurity policy di	22 gennaio 2024

Modifica	Descrizione	Data
	Amazon che fornisce l'accesso amministrativo a AWS KMS Secrets Manager e consente alle funzionalità TLS di Amazon ECS Service Connect di funzionare correttamente. AWS Private Certificate Authority	
Aggiungi una nuova politica Amazon EC2InfrastructureRolePolicyForVolumes	La AmazonEC2InfrastructureRolePolicyForVolumes politica è stata aggiunta. La policy concede le autorizzazioni necessarie ad Amazon ECS per effettuare chiamate AWS API per gestire i volumi Amazon EBS associati ai carichi di lavoro Amazon ECS.	11 gennaio 2024
Aggiungi autorizzazioni ad Amazon EC2ServiceRolePolicy	La policy IAM AmazonEC2ServiceRolePolicy gestita è stata aggiornata con nuove events autorizzazioni e autorizzazioni autoscaling e autoscaling-plans aggiuntive.	4 dicembre 2023
Aggiungi autorizzazioni ad Amazon EC2 Container ServiceEventsRole	La policy IAM AmazonEC2ServiceRolePolicy gestita è stata aggiornata per consentire l'accesso al funzionamento dell' AWS Cloud Map DiscoverInstancesRevision API.	4 ottobre 2023

Modifica	Descrizione	Data
Aggiungi autorizzazioni ad Amazon EC2 Container ServiceforEC2 Role	La AmazonEC2ContainerServiceforEC2Role politica è stata modificata per aggiungere l'ecs:TagResource autorizzazione, che include una condizione che limita l'autorizzazione solo ai cluster appena creati e alle istanze di container registrate.	6 marzo 2023
Aggiunta di autorizzazioni a the section called “Amazon ECS_FullAccess”	La AmazonECS_FullAccess politica è stata modificata per aggiungere l'elasticloadbalancing:AddTags autorizzazione, che include una condizione che limita l'autorizzazione solo ai sistemi di bilanciamento del carico, ai gruppi target, alle regole e ai listener creati di recente. Questa autorizzazione non consente l'aggiunta di tag a risorse Elastic Load Balancing già create.	4 gennaio 2023
Amazon ECS ha cominciato a tenere traccia delle modifiche	Amazon ECS ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	8 giugno 2021

Politiche IAM AWS gestite eliminate gradualmente per Amazon Elastic Container Service

Le seguenti politiche IAM AWS gestite vengono eliminate gradualmente. Queste policy vengono ora sostituite da policy aggiornate. È consigliabile aggiornare gli utenti o i ruoli in modo che utilizzino le policy aggiornate.

Amazon EC2 ContainerServiceFullAccess

Important

La policy IAM gestita da `AmazonEC2ContainerServiceFullAccess` è stata gradualmente eliminata a partire dal 29 gennaio 2021, in risposta a un risultato di sicurezza con l'autorizzazione `iam:passRole`. Questa autorizzazione consente di accedere a tutte le risorse, incluse le credenziali ai ruoli nell'account. Ora che la policy è stata eliminata gradualmente, non sarà più possibile collegarla a nuovi utenti o ruoli. Qualsiasi utente o ruolo che dispone già della policy può continuare a utilizzarlo. Tuttavia, consigliamo di aggiornare gli utenti o i ruoli perché utilizzino invece la policy gestita da `AmazonECS_FullAccess`. Per ulteriori informazioni, consulta [Migrazione alla policy gestita da AmazonECS_FullAccess](#).

Amazon EC2 ContainerServiceRole

Important

La policy IAM gestita da `AmazonEC2ContainerServiceRole` viene gradualmente eliminata. Ora è sostituita dal ruolo collegato al servizio Amazon ECS. Per ulteriori informazioni, consulta [Uso di ruoli collegati ai servizi per Amazon ECS](#).

Amazon EC2 ContainerServiceAutoscaleRole

Important

La policy IAM gestita da `AmazonEC2ContainerServiceAutoscaleRole` viene gradualmente eliminata. Ora è sostituita dal ruolo collegato al servizio Application Auto Scaling per Amazon ECS. Per ulteriori informazioni, consulta [Ruoli collegati ai servizi per Application Auto Scaling](#) nella Guida per l'utente di Application Auto Scaling.

Migrazione alla policy gestita da **AmazonECS_FullAccess**

La policy IAM gestita da `AmazonEC2ContainerServiceFullAccess` è stata gradualmente eliminata a partire dal 29 gennaio 2021, in risposta a un risultato di sicurezza con l'autorizzazione `iam:passRole`. Questa autorizzazione consente di accedere a tutte le risorse, incluse le credenziali ai ruoli nell'account. Ora che la policy è stata eliminata gradualmente, non sarà più possibile collegarla a nuovi gruppi, utenti o ruoli. Qualsiasi gruppo, utente o ruolo che dispone già della policy collegata può continuare a utilizzarla. Tuttavia, consigliamo di aggiornare i gruppi, gli utenti o i ruoli perché utilizzino invece la policy gestita da `AmazonECS_FullAccess`.

Le autorizzazioni concesse dalla policy `AmazonECS_FullAccess` includono l'elenco completo di autorizzazioni necessarie per utilizzare ECS come amministratore. Se attualmente utilizzi autorizzazioni concesse dalla `AmazonEC2ContainerServiceFullAccess` politica che non sono incluse nella `AmazonECS_FullAccess` politica, puoi aggiungerle a una dichiarazione politica in linea. Per ulteriori informazioni, consulta [AWS politiche gestite per Amazon Elastic Container Service](#).

Utilizza la seguente procedura per determinare se disponi di gruppi, utenti o ruoli che attualmente utilizzano la policy IAM gestita da `AmazonEC2ContainerServiceFullAccess`. Quindi, aggiornali per scollegare la policy precedente e collega la policy `AmazonECS_FullAccess`.

Per aggiornare un gruppo, un utente o un ruolo per utilizzare la policy di `FullAccess AmazonECS_` (`AWS Management Console`)

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione, scegli Policy e cerca e seleziona la policy `AmazonEC2ContainerServiceFullAccess`.
3. Seleziona la scheda Utilizzo della policy che riporta qualsiasi ruolo IAM che sta attualmente utilizzando questa policy.
4. Per ogni ruolo IAM che attualmente utilizza la `AmazonEC2ContainerServiceFullAccess` policy, seleziona il ruolo e segui i seguenti passaggi per scollegare la policy eliminata gradualmente e allegarla. `AmazonECS_FullAccess`
 - a. Nella scheda Autorizzazioni, scegli la X accanto alla `EC2 ContainerServiceFullAccess` politica di Amazon.
 - b. Scegli Aggiungi autorizzazioni.
 - c. Scegli Allega direttamente le politiche esistenti, cerca e seleziona la `FullAccess` politica `AmazonECS_`, quindi scegli Avanti: revisione.

- d. Rivedi le modifiche e scegli Aggiungi autorizzazioni.
- e. Ripeti questa procedura per ogni gruppo, utente o ruolo che utilizza la policy `AmazonEC2ContainerServiceFullAccess`.

Aggiornamento di un gruppo, un utente o un ruolo per utilizzare la policy **AmazonECS_FullAccess** (AWS CLI)

1. Utilizzo dell'[generate-service-last-accessed-details](#) comando per generare un report che includa dettagli sull'ultima volta che è stata utilizzata la politica di eliminazione graduale.

```
aws iam generate-service-last-accessed-details \
  --arn arn:aws:iam::aws:policy/AmazonEC2ContainerServiceFullAccess
```

Output di esempio:

```
{
  "JobId": "32bb1fb0-1ee0-b08e-3626-ae83EXAMPLE"
}
```

2. Utilizza l'ID del lavoro dell'output precedente con [get-service-last-accessed-details](#) comando per recuperare l'ultimo report del servizio a cui si accede. Questo report mostra l'Amazon Resource Name (ARN) delle entità IAM che hanno utilizzato per l'ultima volta la politica di eliminazione graduale.

```
aws iam get-service-last-accessed-details \
  --job-id 32bb1fb0-1ee0-b08e-3626-ae83EXAMPLE
```

3. Usa uno dei seguenti comandi per scollegare la policy `AmazonEC2ContainerServiceFullAccess` da un gruppo, un utente o un ruolo.
 - [detach-group-policy](#)
 - [detach-role-policy](#)
 - [detach-user-policy](#)
4. Usa uno dei seguenti comandi per collegare la policy `AmazonECS_FullAccess` a un gruppo, un utente o un ruolo.
 - [attach-group-policy](#)
 - [attach-role-policy](#)

- [attach-user-policy](#)

Uso di ruoli collegati ai servizi per Amazon ECS

Amazon Elastic Container Service utilizza AWS Identity and Access Management ruoli [collegati ai servizi](#) (IAM). Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente ad Amazon ECS. I ruoli collegati ai servizi sono definiti automaticamente da Amazon ECS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon ECS perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon ECS definisce le autorizzazioni dei ruoli collegati ai servizi e, salvo diversamente definito, solo Amazon ECS può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni su altri servizi che supportano ruoli collegati ai servizi, consulta i [AWS servizi che funzionano con IAM](#) e cerca i servizi con Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni del ruolo collegato ai servizi per Amazon ECS

Amazon ECS utilizza il ruolo collegato al servizio denominato ECS. AWSService RoleFor

Il ruolo collegato AWSService RoleFor ai servizi ECS si affida ai seguenti servizi per assumere il ruolo:

- `ecs.amazonaws.com`

La politica di autorizzazione dei ruoli denominata Amazon ECSService RolePolicy consente ad Amazon ECS di completare le seguenti azioni sulle risorse specificate:

- Azione: quando utilizzi la modalità di rete `awsvpc` per le attività Amazon ECS, Amazon ECS gestisce il ciclo di vita delle interfacce di rete elastiche associate all'attività. Ciò include anche i tag che Amazon ECS aggiunge alle interfacce di rete elastiche.
- Azione: quando si utilizza un sistema di bilanciamento del carico con il servizio Amazon ECS, Amazon ECS gestisce la registrazione e l'annullamento della registrazione delle risorse con il sistema di bilanciamento del carico.

- Azione: quando si utilizza Amazon ECS Service Discovery, Amazon ECS gestisce la Route 53 e AWS Cloud Map le risorse necessarie per il funzionamento del service discovery.
- Operazione: quando si utilizza il dimensionamento automatico del servizio Amazon ECS, Amazon ECS gestisce le risorse con dimensionamento automatico richieste.
- Azione: Amazon ECS crea e gestisce CloudWatch allarmi e flussi di log che aiutano nel monitoraggio delle tue risorse Amazon ECS.
- Operazione: quando si utilizza Amazon ECS Exec, Amazon ECS gestisce le autorizzazioni necessarie per avviare le sessioni Amazon ECS Exec per le attività.
- Operazione: quando si utilizza Amazon ECS Service Connect, Amazon ECS gestisce le risorse AWS Cloud Map richieste per utilizzare la funzionalità.
- Azione: quando si utilizzano i provider di capacità Amazon ECS, Amazon ECS gestisce le autorizzazioni necessarie per modificare il gruppo Auto Scaling e le relative istanze Amazon. EC2

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per Amazon ECS

Nella maggior parte dei casi, non devi creare manualmente un ruolo collegato ai servizi. Quando crei un cluster o crei o aggiorni un servizio nell' AWS Management Console AWS API AWS CLI, Amazon ECS crea il ruolo collegato al servizio per te. Se non vedi il ruolo `AWSServiceRoleForECS` dopo aver creato un cluster, esegui le seguenti operazioni per risolvere il problema:

- Verifica e configura le autorizzazioni per consentire ad Amazon ECS di creare, modificare o eliminare un ruolo collegato ai servizi per tuo conto. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente di IAM.
- Ritenta l'operazione di creazione del cluster o crea manualmente il ruolo collegato al servizio.

Puoi utilizzare la console IAM per creare il ruolo collegato al servizio `AWSServiceRoleForECS`. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `ecs.amazonaws.com` Per ulteriori informazioni, consulta [Creazione di un ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

 Important

Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo.

Se elimini questo ruolo collegato al servizio, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un cluster oppure quando crei o aggiorni un servizio, Amazon ECS crea il ruolo collegato al servizio per tuo conto.

Se elimini questo ruolo collegato al servizio, puoi utilizzare lo stesso processo IAM per crearlo nuovamente.

Modifica di un ruolo collegato al servizio per Amazon ECS

Amazon ECS non consente di modificare il ruolo collegato al servizio AWSService RoleFor ECS. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Aggiornare un ruolo collegato al servizio nella Guida per l'utente IAM](#).

Eliminazione di un ruolo collegato ai servizi per Amazon ECS

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

 Note

Se il servizio Amazon ECS utilizza tale ruolo quando provi a eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per verificare se il ruolo collegato al servizio dispone di una sessione attiva

1. Aprire la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione, scegli Ruoli e scegli il nome AWSService RoleFor ECS (non la casella di controllo).

3. Nella pagina Riepilogo, seleziona Consulente accessi ed esamina l'attività recente per il ruolo collegato al servizio.

Note

Se non sei sicuro che Amazon ECS stia utilizzando il ruolo AWSService RoleFor ECS, puoi provare a eliminarlo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni in cui il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato al servizio.

Per rimuovere le risorse Amazon ECS utilizzate dal ruolo collegato al AWSService RoleFor servizio ECS

È necessario eliminare tutti i cluster Amazon ECS in tutte le AWS regioni prima di poter eliminare il ruolo AWSService RoleFor ECS.

1. Dimensione a 0 il conteggio di tutti i servizi Amazon ECS in tutte le regioni, quindi elimina i servizi. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS tramite la console](#) e [Eliminazione di un servizio Amazon ECS tramite la console](#).
2. Forza l'annullamento della registrazione di tutte le istanze di container da tutti i cluster in tutte le regioni. Per ulteriori informazioni, consulta [Annullamento della registrazione di un'istanza di container Amazon ECS](#).
3. Elimina tutti i cluster Amazon ECS in tutte le regioni. Per ulteriori informazioni, consulta [Eliminazione di un cluster Amazon ECS](#).

Eliminazione manuale del ruolo collegato al servizio con IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al servizio AWSService RoleFor ECS. Per ulteriori informazioni, consulta [Eliminazione del ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon ECS

Amazon ECS supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta [Regioni ed endpoint di AWS](#).

Ruoli IAM per Amazon ECS

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità utilizzate. Utilizza la tabella seguente per determinare quali ruoli IAM sono necessari per Amazon ECS.

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri AWS servizi per tuo conto.	La tua attività è ospitata su AWS Fargateo su istanze esterne e: • estrae un'immagine del contenuto da un repository privato Amazon ECR. • estrae un'immagine del contenuto da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività. • invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. <code>awslogs</code>	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		La tua attività è ospitata su una AWS Fargate EC2 delle istanze Amazon e: • utilizza l'autenticazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di Secrets Manager o i parametri AWS Systems Manager Parameter Store.	
Ruolo del processo	Questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi.	L'applicazione accede ad altri AWS servizi, come Amazon S3.	Ruolo IAM dell'attività Amazon ECS
Ruolo dell'istanza di container	Questo ruolo consente alle EC2 istanze o alle istanze esterne di registrarsi nel cluster.	La tua attività è ospitata su EC2 istanze Amazon o su un'istanza esterna.	Ruolo IAM delle istanze di container Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di Amazon ECS Anywhere	Questo ruolo consente l'accesso alle istanze esterne. AWS APIs	La tua attività è ospitata su istanze esterne.	Ruolo IAM di Amazon ECS Anywhere
Ruolo di Amazon ECS CodeDeploy	Questo ruolo consente CodeDeploy di aggiornare i tuoi servizi.	Si utilizza il tipo di distribuzione CodeDeploy blu/verde per distribuire i servizi.	Ruolo CodeDeploy IAM di Amazon ECS
Ruolo di Amazon ECS EventBridge	Questo ruolo consente EventBridge di aggiornare i tuoi servizi.	Utilizzi le EventBridge regole e gli obiettivi per pianificare le tue attività.	Ruolo EventBridge IAM di Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo dell'infrastruttura a Amazon ECS	Questo ruolo consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività. • Desideri utilizzare e Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect. • Vuoi creare gruppi target VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS

Le migliori pratiche per i ruoli IAM in Amazon ECS

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità utilizzate. Ti consigliamo di creare ruoli separati nella tabella anziché condividerli.

Ruolo	Definizione	Se necessario	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri	La tua attività è ospitata su AWS Fargateo su istanze esterne e:	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Se necessario	Ulteriori informazioni
	AWS servizi per tuo conto.	• estrae un'immagine del contenitore da un repository privato Amazon ECR. • estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività. • invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. <code>awslogs</code> La tua attività è ospitata su una AWS Fargate EC2 delle istanze Amazon e: • utilizza l'autenticazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di	

Ruolo	Definizione	Se necessario	Ulteriori informazioni
		Secrets Manager o i parametri AWS Systems Manager Parameter Store.	
Ruolo del processo	Questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi.	L'applicazione accede ad altri AWS servizi, come Amazon S3.	Ruolo IAM dell'attività Amazon ECS
Ruolo dell'istanza di container	Questo ruolo consente alle EC2 istanze o alle istanze esterne di registrarsi nel cluster.	La tua attività è ospitata su EC2 istanze Amazon o su un'istanza esterna.	Ruolo IAM delle istanze di container Amazon ECS
Ruolo di Amazon ECS Anywhere	Questo ruolo consente l'accesso alle istanze esterne. AWS APIs	La tua attività è ospitata su istanze esterne.	Ruolo IAM di Amazon ECS Anywhere
Ruolo di Amazon ECS CodeDeploy	Questo ruolo consente CodeDeploy di aggiornare i tuoi servizi.	Si utilizza il tipo di distribuzione CodeDeploy blu/verde per distribuire i servizi.	Ruolo CodeDeploy IAM di Amazon ECS
Ruolo di Amazon ECS EventBridge	Questo ruolo consente EventBridge di aggiornare i tuoi servizi.	Utilizzi le EventBridge regole e gli obiettivi per pianificare le tue attività.	Ruolo EventBridge IAM di Amazon ECS

Ruolo	Definizione	Se necessario	Ulteriori informazioni
Ruolo dell'infrastruttura a Amazon ECS	Questo ruolo consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività. • Desideri utilizzare e Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect. • Vuoi creare gruppi target VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS

Ruolo del processo

Ti consigliamo di assegnare un ruolo da svolgere. Il suo ruolo può essere distinto dal ruolo dell' EC2 istanza Amazon su cui è in esecuzione. L'assegnazione di un ruolo a ciascuna attività è conforme al principio dell'accesso con privilegi minimi e consente un controllo più granulare su operazioni e risorse.

Quando assegni ruoli IAM a un'attività, devi utilizzare la seguente policy di fiducia in modo che ciascuna delle tue attività possa assumere un ruolo IAM diverso da quello utilizzato dall' EC2 istanza. In questo modo, l'attività non eredita il ruolo dell'istanza. EC2

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Sid": "",
    "Effect": "Allow",
    "Principal": {
      "Service": "ecs-tasks.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
  }
]
```

Quando aggiungi un ruolo di attività a una definizione di attività, l'agente di container Amazon ECS crea in automatico un token con un ID di credenziali univoco (ad esempio 12345678-90ab-cdef-1234-567890abcdef) per l'attività. Il token e le credenziali del ruolo vengono quindi aggiunti alla cache interna dell'agente. L'agente popola la variabile di ambiente `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` nel container con l'URI dell'ID delle credenziali (ad esempio `/v2/credentials/12345678-90ab-cdef-1234-567890abcdef`).

Puoi recuperare manualmente le credenziali temporanee del ruolo da un container aggiungendo la variabile di ambiente all'indirizzo IP dell'agente di container Amazon ECS ed eseguendo il comando `curl` sulla stringa risultante.

```
curl 169.254.170.2$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

L'output previsto è il seguente:

```
{
  "RoleArn": "arn:aws:iam::123456789012:role/SSMTaskRole-SSMFargateTaskIAMRole-DASWWSF2WGD6",
  "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
  "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxrFiCYEXAMPLEKEY",
  "Token": "IQoJb3JpZ2luX2VjEEM/Example==",
  "Expiration": "2021-01-16T00:51:53Z"
}
```

Le versioni più recenti di recuperano AWS SDKs automaticamente queste credenziali dalla variabile di ambiente `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` quando si effettuano chiamate API. AWS [Per informazioni su come rinnovare le credenziali, consulta Rinnovo delle credenziali su RePost. AWS](#)

L'output include una coppia di chiavi di accesso composta da un ID di chiave di accesso segreto e da una chiave segreta utilizzata dall'applicazione per accedere alle risorse. AWS Include anche un token che AWS viene utilizzato per verificare la validità delle credenziali. Per impostazione predefinita, le credenziali assegnate alle attività che utilizzano ruoli di attività sono valide per sei ore. Successivamente, vengono ruotate in automatico dall'agente di container Amazon ECS.

Ruolo di esecuzione di attività

Il ruolo di esecuzione dell'attività viene utilizzato per concedere all'agente container Amazon ECS l'autorizzazione a richiamare azioni AWS API specifiche per tuo conto. Ad esempio, quando si utilizza AWS Fargate, Fargate necessita di un ruolo IAM che le consenta di estrarre immagini da Amazon ECR e scrivere log in Logs. CloudWatch Un ruolo IAM è necessario anche quando un'attività fa riferimento a un segreto archiviato in AWS Secrets Manager, ad esempio un image pull secret.

Note

Se stai estraendo immagini in qualità di utente autenticato, è meno probabile che tu subisca l'impatto delle modifiche apportate ai [limiti del tasso di estrazione di Docker Hub](#). Per ulteriori informazioni, consulta [Autenticazione di registri privati per istanze di container](#).

Utilizzando Amazon ECR e Amazon ECR Public, puoi evitare i limiti imposti da Docker. Se estrai immagini da Amazon ECR, ciò contribuisce anche ad abbreviare i tempi di estrazione della rete e a ridurre le modifiche al trasferimento di dati quando il traffico esce dal tuo VPC.

Important

Quando utilizzi Fargate, devi autenticarti in un registro di immagini privato utilizzando `repositoryCredentials`. Non è possibile impostare le variabili di ambiente dell'agente di container Amazon ECS `ECS_ENGINE_AUTH_TYPE` o `ECS_ENGINE_AUTH_DATA` o modificare il file `ecs.config` per le attività in hosting su Fargate. Per ulteriori informazioni, consulta [Autenticazione del registro privato per le attività](#).

Ruolo dell'istanza di container

La policy IAM gestita da `AmazonEC2ContainerServiceforEC2Role` deve includere le seguenti autorizzazioni: Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, la policy gestita `AmazonEC2ContainerServiceforEC2Role` può essere utilizzata come guida. Se

una qualsiasi delle autorizzazioni concesse nella policy gestita non è necessaria per il caso d'uso, crea una policy personalizzata e aggiungi solo le autorizzazioni richieste.

- `ec2:DescribeTags`— (Facoltativo) Consente a un responsabile di descrivere i tag associati a un' EC2 istanza Amazon. Questa autorizzazione viene utilizzata dall'agente del container Amazon ECS per supportare la propagazione dei tag delle risorse. Per ulteriori informazioni, consulta [Assegnazione di tag alle risorse](#).
- `ecs:CreateCluster`— (Facoltativo) Consente a un principale di creare un cluster Amazon ECS. Questa autorizzazione viene utilizzata dall'agente del container di Amazon ECS per creare un cluster default, se non è già presente.
- `ecs:DeregisterContainerInstance`— (Facoltativo) Consente a un principale di annullare la registrazione di un'istanza di container Amazon ECS da un cluster. L'agente container Amazon ECS non chiama questa operazione API, ma questa autorizzazione rimane per garantire la compatibilità con le versioni precedenti.
- `ecs:DiscoverPollEndpoint`— (Obbligatorio) Questa azione restituisce gli endpoint utilizzati dall'agente container Amazon ECS per il polling degli aggiornamenti.
- `ecs:Poll`— (Obbligatorio) Consente all'agente container Amazon ECS di comunicare con il piano di controllo di Amazon ECS per segnalare le modifiche allo stato delle attività.
- `ecs:RegisterContainerInstance`— (Obbligatorio) Consente a un principale di registrare un'istanza di container in un cluster. Questa autorizzazione viene utilizzata dall'agente container Amazon ECS per registrare l' EC2 istanza Amazon in un cluster e per supportare la propagazione dei tag di risorse.
- `ecs:StartTelemetrySession`— (Facoltativo) Consente all'agente container Amazon ECS di comunicare con il piano di controllo di Amazon ECS per riportare informazioni e parametri sullo stato di salute per ogni container e attività.

Sebbene questa autorizzazione non sia richiesta, ti consigliamo di aggiungerla per consentire ai parametri dell'istanza del contenitore di avviare azioni di scalabilità e ricevere anche report relativi ai comandi di controllo dello stato.

- `ecs:TagResource`— (Facoltativo) Consente all'agente container Amazon ECS di etichettare il cluster al momento della creazione e di etichettare le istanze di container quando sono registrate in un cluster.
- `ecs:UpdateContainerInstancesState`: consente a un principale di modificare lo stato di un'istanza di container Amazon ECS. Questa autorizzazione viene utilizzata dall'agente del container di Amazon ECS per il drenaggio dell'istanza Spot.

- `ecs:Submit*`— (Obbligatorio) Ciò include le azioni `SubmitAttachmentStateChangesSubmitContainerStateChange`, e `SubmitTaskStateChange` API. Vengono utilizzati dall'agente del container di Amazon ECS per segnalare le modifiche di stato per ogni risorsa al piano di controllo Amazon ECS. L'`SubmitContainerStateChange` autorizzazione non viene più utilizzata dall'agente container Amazon ECS, ma serve a garantire la compatibilità con le versioni precedenti.
- `ecr:GetAuthorizationToken`— (Facoltativo) Consente a un principale di recuperare un token di autorizzazione. Un token di autorizzazione rappresenta le credenziali di autenticazione IAM e può essere utilizzato per accedere a qualsiasi registro Amazon ECR a cui ha accesso il principale IAM. Il token di autorizzazione ricevuto è valido per 12 ore.
- `ecr:BatchCheckLayerAvailability`— (Facoltativo) Quando l'immagine di un contenitore viene inviata a un repository privato Amazon ECR, ogni livello di immagine viene controllato per verificare se è già stato inviato. Se lo è, il livello dell'immagine viene ignorato.
- `ecr:GetDownloadUrlForLayer`— (Facoltativo) Quando l'immagine di un contenitore viene estratta da un repository privato Amazon ECR, questa API viene chiamata una volta per ogni livello di immagine che non è già memorizzato nella cache.
- `ecr:BatchGetImage`— (Facoltativo) Quando l'immagine di un contenitore viene estratta da un repository privato Amazon ECR, questa API viene chiamata una volta per recuperare il manifesto dell'immagine.
- `logs:CreateLogStream`— (Facoltativo) Consente a un principale di creare un flusso di log di CloudWatch Logs per un gruppo di log specificato.
- `logs:PutLogEvents`— (Facoltativo) Consente a un principale di caricare un batch di eventi di registro in un flusso di log specificato.

La seguente politica contiene le autorizzazioni richieste.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DiscoverPollEndpoint",
        "ecs:Poll",
        "ecs:RegisterContainerInstance",
        "ecs:UpdateContainerInstancesState",
        "ecs:Submit*"
      ]
    }
  ]
}
```

```
    ],  
    "Resource": "*"    
  }  
]  
}
```

Ruoli collegati ai servizi

Puoi utilizzare il ruolo collegato al servizio per Amazon ECS per concedere al servizio Amazon ECS l'autorizzazione a chiamare altri servizi APIs per tuo conto. Amazon ECS necessita delle autorizzazioni per creare ed eliminare interfacce di rete, oltre che per registrare e annullare la registrazione delle destinazioni con un gruppo di destinazione. Inoltre, necessita delle autorizzazioni necessarie per creare ed eliminare policy di dimensionamento. Queste autorizzazioni vengono concesse attraverso il ruolo collegato al servizio. Tale ruolo viene creato per tuo conto la prima volta che utilizzi il servizio.

Note

Se elimini inavvertitamente il ruolo collegato al servizio, puoi ricrearlo. Per ricevere istruzioni, consulta [Creazione di un ruolo collegato al servizio](#).

Consigli sui ruoli

Consigliamo di completare le seguenti operazioni quando configuri i ruoli e le policy IAM dell'attività.

Blocca l'accesso ai EC2 metadati di Amazon

Quando esegui le tue attività su EC2 istanze Amazon, ti consigliamo vivamente di bloccare l'accesso ai EC2 metadati di Amazon per evitare che i contenitori ereditino il ruolo assegnato a tali istanze. Se le tue applicazioni devono richiamare un'azione AWS API, utilizza invece i ruoli IAM per le attività.

Per impedire alle attività in esecuzione in modalità bridge di accedere ai EC2 metadati di Amazon, esegui il comando seguente o aggiorna i dati utente dell'istanza. Per ulteriori istruzioni sull'aggiornamento dei dati utente di un'istanza, consulta il seguente [articolo del Supporto AWS](#). Per ulteriori informazioni sulla modalità bridge per la definizione delle attività, consulta [Modalità di rete per la definizione delle attività](#).

```
sudo yum install -y iptables-services; sudo iptables --insert FORWARD 1 --in-interface  
docker+ --destination 169.254.169.254/32 --jump DROP
```

Affinché questa modifica persista dopo il riavvio, esegui il seguente comando specifico per la tua Amazon Machine Image (AMI):

- Amazon Linux 2

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now iptables
```

- Amazon Linux

```
sudo service iptables save
```

Per le attività che utilizzano la modalità di rete `awsvpc`, imposta la variabile di ambiente `ECS_AWSVPC_BLOCK_IMDS` su `true` nel file `/etc/ecs/ecs.config`.

È necessario impostare la `ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST` variabile su `false` nel `ecs-agent config` file per evitare che i contenitori in esecuzione all'interno della host rete accedano ai EC2 metadati di Amazon.

Usa la modalità **awsvpc** di rete

Utilizza la modalità di rete `awsvpc` per limitare il flusso di traffico tra diverse attività o tra le attività e altri servizi eseguiti all'interno di Amazon VPC. In questo modo si aggiunge un ulteriore livello di sicurezza. La modalità `awsvpc` di rete fornisce l'isolamento della rete a livello di attività per le attività eseguite su Amazon. EC2 È la modalità predefinita attiva AWS Fargate. È l'unica modalità di rete che puoi usare per assegnare un gruppo di sicurezza alle attività.

Utilizza le informazioni relative all'ultimo accesso per rifinire i ruoli

Consigliamo di rimuovere tutte le operazioni che non sono mai state utilizzate o che non sono state utilizzate per un certo periodo di tempo. Ciò impedisce che si verifichino accessi indesiderati. A tale scopo, esamina le ultime informazioni di accesso fornite da IAM, quindi rimuovi le azioni che non sono mai state utilizzate o che non sono state utilizzate di recente. Per farlo, completa la procedura seguente.

Utilizza il comando seguente per generare un report indicante le informazioni di accesso più recenti per la policy di riferimento:

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/ExamplePolicy1
```


utilizza il valore `JobId` presente nell'output per eseguire il comando seguente. Dopo aver eseguito questa operazione, puoi visualizzare i risultati del report.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Per ulteriori informazioni, consulta [Perfezionare le autorizzazioni AWS utilizzando le informazioni dell'ultimo accesso](#).

Monitora le attività AWS CloudTrail sospette

Puoi monitorare qualsiasi attività AWS CloudTrail sospetta. La maggior parte delle chiamate AWS API viene registrata AWS CloudTrail come eventi. Vengono analizzate da AWS CloudTrail Insights e l'utente viene avvisato di eventuali comportamenti sospetti associati write alle chiamate API. Ciò potrebbe includere un picco nel volume delle chiamate. Questi avvisi includono informazioni come l'ora in cui si è verificata l'attività insolita e l'ARN principale dell'identità che ha contribuito alla. APIs

Puoi identificare le operazioni eseguite dalle attività con un ruolo IAM in AWS CloudTrail esaminando la proprietà `userIdentity` dell'evento. Nell'esempio seguente, l'arn include il nome del ruolo assunto, `s3-write-go-bucket-role`, seguito dal nome dell'attività, `7e9894e088ad416eb5cab92afExample`.

```
"userIdentity": {  
  "type": "AssumedRole",  
  "principalId": "ARO36C6WWEJ2YEXAMPLE:7e9894e088ad416eb5cab92afExample",  
  "arn": "arn:aws:sts::123456789012:assumed-role/s3-write-go-bucket-  
role/7e9894e088ad416eb5cab92afExample",  
  ...  
}
```

Note

Quando attività che presuppongono un ruolo vengono eseguite su istanze di EC2 container Amazon, una richiesta viene registrata dall'agente container Amazon ECS nel registro di controllo dell'agente che si trova a un indirizzo nel formato. `/var/log/ecs/audit.log.YYYY-MM-DD-HH` Per ulteriori informazioni, consulta [Log di ruoli IAM di attività e Registrazione di eventi di approfondimenti per i trail](#).

Ruolo IAM di esecuzione di attività Amazon ECS

Il ruolo di esecuzione del processo concede all'agente del container di Amazon ECS e agli agenti Fargate l'autorizzazione per effettuare chiamate API AWS per tuo conto. Il ruolo IAM di esecuzione di attività è necessario a seconda dei requisiti dell'attività. È possibile disporre di più ruoli di esecuzione di attività per scopi e servizi diversi associati all'account.

Note

Queste autorizzazioni non sono accessibili dai contenitori inclusi nell'attività. Per conoscere le autorizzazioni IAM necessarie per l'esecuzione dell'applicazione, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Di seguito sono elencati i casi di utilizzo comune per un ruolo IAM di esecuzione di attività:

- L'attività è ospitata su AWS Fargate su un'istanza esterna e:
 - estrae un'immagine del contenitore da un repository privato Amazon ECR.
 - estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività.
 - invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. `awslogs` Per ulteriori informazioni, consulta [Invia i log di Amazon ECS a CloudWatch](#).
- Le tue attività sono ospitate su una delle AWS Fargate EC2 istanze di Amazon e:
 - utilizza l'autenticazione del registro privato. Per ulteriori informazioni, consulta [Autorizzazioni di autenticazione del registro privato](#).
 - utilizza Runtime Monitoring.
 - la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di Secrets Manager o i parametri AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Autorizzazioni Secrets Manager o Systems Manager](#).

Note

Il ruolo di esecuzione dei processi è supportato dall'agente del container di Amazon ECS versione 1.16.0 e successiva.

Amazon ECS fornisce la policy gestita denominata Amazon ECSTask ExecutionRolePolicy che contiene le autorizzazioni richieste dai casi d'uso comuni sopra descritti. Per ulteriori informazioni, consulta [Amazon ECSTask ExecutionRolePolicy](#) nella AWS Managed Policy Reference Guide. Potrebbe essere necessario aggiungere politiche in linea al ruolo di esecuzione delle attività per casi d'uso speciali.

La console Amazon ECS crea un ruolo di esecuzione delle attività. Puoi allegare manualmente la policy IAM gestita per le attività per consentire ad Amazon ECS di aggiungere autorizzazioni per funzionalità e miglioramenti futuri man mano che vengono introdotti. Puoi utilizzare la ricerca della console IAM per cercare `ecsTaskExecutionRole` e verificare se il tuo account ha già il ruolo di esecuzione delle attività. Per ulteriori informazioni, consulta la [ricerca nella console IAM](#) nella guida per l'utente IAM.

Se richiami le immagini come utente autenticato, è meno probabile che tu sia influenzato dalle modifiche apportate ai limiti di [pull rate di Docker Hub](#). Per ulteriori informazioni, consulta [Autenticazione di registri privati per istanze di container](#).

Utilizzando Amazon ECR e Amazon ECR Public, puoi evitare i limiti imposti da Docker. Se estrai immagini da Amazon ECR, ciò contribuisce anche ad abbreviare i tempi di estrazione della rete e a ridurre le modifiche al trasferimento di dati quando il traffico esce dal tuo VPC.

Quando utilizzi Fargate, devi autenticarti in un registro di immagini privato utilizzando `repositoryCredentials`. Non è possibile impostare le variabili di ambiente dell'agente di container Amazon ECS `ECS_ENGINE_AUTH_TYPE` o `ECS_ENGINE_AUTH_DATA` o modificare il file `ecs.config` per le attività in hosting su Fargate. Per ulteriori informazioni, consulta [Autenticazione del registro privato per le attività](#).

Creazione del ruolo di esecuzione attività

Se il tuo account non ha già un ruolo di esecuzione delle attività, utilizza i seguenti passaggi per creare il ruolo.

AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.

4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Scegli Next (Successivo).
6. Nella sezione Aggiungi autorizzazioni, cerca Amazon ECSTask ExecutionRolePolicy, quindi seleziona la politica.
7. Scegli Next (Successivo).
8. Per Nome del ruolo, inserisci ecsTaskExecutionRuolo.
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato ecs-tasks-trust-policy.json contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Crea un ruolo IAM denominato ecsTaskExecutionRole utilizzando la policy di attendibilità creata nel passaggio precedente.

```
aws iam create-role \
  --role-name ecsTaskExecutionRole \
  --assume-role-policy-document file://ecs-tasks-trust-policy.json
```

3. Allega la AmazonECSTaskExecutionRolePolicy policy AWS gestita al ecsTaskExecutionRole ruolo.

```
aws iam attach-role-policy \
  --role-name ecsTaskExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Usa le credenziali di Secrets Manager per accedere all'archivio privato delle immagini del contenitore	Autorizzazioni di autenticazione del registro privato
Trasmetti dati sensibili con Systems Manager o Secrets Manager	Autorizzazioni Secrets Manager o Systems Manager
Fai in modo che le attività di Fargate recuperino le immagini di Amazon ECR dagli endpoint dell'interfaccia	Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia
Ospita i file di configurazione in un bucket Amazon S3	Autorizzazioni per lo storage di file Amazon S3
Configura Container Insights per visualizzare gli eventi del ciclo di vita di Amazon ECS	Autorizzazioni necessarie per configurare Approfondimenti sui container per visualizzare gli eventi del ciclo di vita di Amazon ECS
Visualizza gli eventi del ciclo di vita di Amazon ECS in Container Insights	Autorizzazioni necessarie per visualizzare gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container

Autorizzazioni di autenticazione del registro privato

Le autorizzazioni seguenti devono essere aggiunte manualmente come policy inline al ruolo per l'esecuzione di attività, per fornire l'accesso ai segreti che crei. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: obbligatorio solo se la chiave utilizza una chiave KMS personalizzata e non quella di default. Il nome della risorsa Amazon (ARN) per la chiave personalizzata deve essere aggiunto come risorsa.

Di seguito viene riportata una policy inline che aggiunge le autorizzazioni.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
        "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
      ]
    }
  ]
}
```

Autorizzazioni Secrets Manager o Systems Manager

L'autorizzazione per consentire all'agente del contenitore di estrarre le risorse necessarie AWS Systems Manager o Secrets Manager. Per ulteriori informazioni, consulta [Trasferisci dati sensibili a un contenitore Amazon ECS](#).

Uso di Secrets Manager

Per fornire l'accesso ai segreti di Gestione di segreti che crei, aggiungi manualmente la seguente autorizzazione al ruolo di esecuzione dell'attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente IAM.

- `secretsmanager:GetSecretValue`: obbligatorio se si fa riferimento a un segreto di Gestione dei segreti. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

La policy dell'esempio seguente aggiunge le autorizzazioni necessarie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"
      ]
    }
  ]
}
```

Uso di Systems Manager

Important

Per le attività che utilizzano il tipo di EC2 avvio, è necessario utilizzare la variabile di configurazione dell'agente ECS

ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true per utilizzare questa funzionalità. Puoi aggiungerlo al file `./etc/ecs/ecs.config` durante la creazione dell'istanza di container oppure aggiungerlo a un'istanza esistente e quindi riavviare l'agente ECS. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Per fornire l'accesso ai parametri di Archivio dei parametri Systems Manager che hai creato, aggiungi manualmente le autorizzazioni seguenti come policy al ruolo per l'esecuzione di attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente IAM.

- `ssm:GetParameters`: obbligatorio quando si fa riferimento a un parametro di Archivio dei parametri Systems Manager in una definizione di attività. Aggiunge l'autorizzazione per recuperare i parametri di Systems Manager.
- `secretsmanager:GetSecretValue`: obbligatorio quando si fa riferimento a un segreto di Gestione dei segreti direttamente o se il parametro di Archivio dei parametri Systems Manager

fa riferimento a un segreto di Gestione dei segreti in una definizione di attività. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave gestita personalizzata e non quella predefinita. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa. Aggiunge l'autorizzazione per decrittografare la chiave gestita dal cliente.

La seguente policy di esempio aggiunge le autorizzazioni necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:region:aws_account_id:parameter/parameter_name",
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name",
        "arn:aws:kms:region:aws_account_id:key/key_id"
      ]
    }
  ]
}
```

Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia

Quando avvii processi che usano il tipo di avvio Fargate che estraggono immagini da Amazon ECR quando Amazon ECR è configurato per utilizzare un endpoint VPC dell'interfaccia, puoi limitare l'accesso a un VPC o endpoint VPC specifico. Per fare questo, crea un ruolo di esecuzione dei processi per i processi che utilizzano chiavi di condizione IAM.

Utilizza le chiavi di condizione globali IAM seguenti per limitare l'accesso a un VPC o endpoint VPC specifico. Per ulteriori informazioni, consulta [Chiavi di contesto delle condizioni globali AWS](#).

- `aws:SourceVpc`: limita l'accesso a un VPC specifico. È possibile limitare il VPC al VPC che ospita l'attività e l'endpoint.

- `aws:SourceVpce`: limita l'accesso a un endpoint VPC specifico.

La policy del ruolo di esecuzione delle attività fornisce un esempio per l'aggiunta di chiavi di condizione:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "aws:sourceVpce": "vpce-xxxxxx",
          "aws:sourceVpc": "vpc-xxxxxx"
        }
      }
    }
  ]
}
```

Autorizzazioni per lo storage di file Amazon S3

Quando specifichi un file di configurazione ospitato in Amazon S3, il ruolo di esecuzione dell'attività deve includere `s3:GetObject` autorizzazione per il file di configurazione e `s3:GetBucketLocation` autorizzazione per il bucket Amazon S3 in cui si trova il file. Per ulteriori informazioni, consulta [le azioni politiche per Amazon S3](#) nella Guida per l'utente di Amazon Simple Storage Service.

La seguente policy di esempio aggiunge le autorizzazioni necessarie per il recupero di un file da Amazon S3. Specifica il nome del bucket Amazon S3 e il nome del file di configurazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket/folder_name/config_file_name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::amzn-s3-demo-bucket"
      ]
    }
  ]
}
```

Autorizzazioni necessarie per configurare Approfondimenti sui container per visualizzare gli eventi del ciclo di vita di Amazon ECS

Le seguenti autorizzazioni sono necessarie nel ruolo dell'attività per configurare gli eventi del ciclo di vita:

- eventi: PutRule
- eventi: PutTargets
- registri: CreateLogGroup

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Effect": "Allow",
  "Action": [
    "events:PutRule",
    "events:PutTargets",
    "logs:CreateLogGroup"
  ],
  "Resource": "*"
}
```

Autorizzazioni necessarie per visualizzare gli eventi del ciclo di vita di Amazon ECS in
Approfondimenti sui container

Per visualizzare gli eventi del ciclo di vita sono necessarie le seguenti autorizzazioni. Aggiungi le autorizzazioni seguenti come policy in linea al ruolo di esecuzione di attività. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- eventi: DescribeRule
- eventi: ListTargetsByRule
- registri: DescribeLogGroups

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Ruolo IAM dell'attività Amazon ECS

Le attività Amazon ECS possono avere un ruolo IAM associato. Le autorizzazioni concesse nel ruolo IAM sono assunte dai container in esecuzione nell'attività. Questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi. Il ruolo dell'attività è necessario quando l'applicazione accede ad altri AWS servizi, come Amazon S3.

Note

A queste autorizzazioni non accedono il container Amazon ECS e gli agenti Fargate. Per conoscere le autorizzazioni IAM necessarie ad Amazon ECS per estrarre le immagini di container ed eseguire l'attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Di seguito sono riportati i vantaggi dell'utilizzo dei ruoli di attività:

- Isolamento delle credenziali: un container può recuperare solo le credenziali per il ruolo IAM stabilito nella definizione di attività a cui appartiene; non ha mai accesso alle credenziali destinate a un altro container appartenente a un'altra attività.
- Autorizzazione: le credenziali del ruolo IAM definite per altre attività non sono accessibili ai container privi di autorizzazione.
- Controllo: la registrazione degli accessi e degli eventi è disponibile CloudTrail per garantire un controllo retrospettivo. Le credenziali dell'attività hanno un contesto `taskArn` che è allegato alla sessione, quindi CloudTrail i registri mostrano quale attività utilizza quale ruolo.

Note

Quando specifichi un ruolo IAM per un'attività, l'utente AWS CLI o l'altro SDKs nei contenitori per quell'attività utilizza esclusivamente le AWS credenziali fornite dal ruolo dell'attività e non eredita più alcuna autorizzazione IAM dall'Amazon EC2 o dall'istanza esterna su cui è in esecuzione.

Creazione del ruolo IAM dell'attività

Quando crei una policy IAM per le tue attività da utilizzare, la policy deve includere le autorizzazioni che desideri vengano assunte dai contenitori delle tue attività. Puoi utilizzare una policy AWS gestita

esistente oppure puoi creare una policy personalizzata partendo da zero che soddisfi le tue esigenze specifiche. Per ulteriori informazioni, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

 Important

Per le attività di Amazon ECS (per tutti i tipi di avvio), ti consigliamo di utilizzare il ruolo e la policy IAM per le tue attività. Queste credenziali consentono all'attività di effettuare richieste AWS API senza `sts:AssumeRole` dover chiamare per assumere lo stesso ruolo già associato all'attività. Se l'attività richiede che un ruolo assuma se stesso, devi creare una policy di attendibilità che consenta esplicitamente a tale ruolo di assumere se stesso. Per ulteriori informazioni, consulta [Aggiornamento di una policy di trust per i ruoli](#) nella Guida per l'utente IAM.

Una volta creata la policy IAM, puoi creare un ruolo IAM che include la policy a cui fai riferimento nella definizione dell'attività Amazon ECS. Puoi creare il ruolo utilizzando il caso d'uso Attività Elastic Container Service nella console IAM. Quindi, puoi collegare la tua politica IAM specifica al ruolo che fornisce ai contenitori della tua attività le autorizzazioni desiderate. Di seguito viene descritto come procedere.

In caso di più definizioni di attività o servizi che richiedono autorizzazioni IAM, è consigliabile creare un ruolo per ogni specifica definizione di attività o per ogni specifico servizio con il numero minimo di autorizzazioni necessarie per il funzionamento dell'attività, in modo da ridurre al minimo l'accesso fornito per ciascuna attività.

Per informazioni sull'endpoint di servizio per la tua regione, consulta [Service endpoints nella Guida](#). Riferimenti generali di Amazon Web Services

Il ruolo dell'attività IAM deve avere una policy di attendibilità che specifica il servizio `ecs-tasks.amazonaws.com`. L'`sts:AssumeRole` autorizzazione consente alle tue attività di assumere un ruolo IAM diverso da quello utilizzato dall' EC2 istanza Amazon. In questo modo, la tua attività non eredita il ruolo associato all' EC2 istanza Amazon. Di seguito è illustrato un esempio di policy di attendibilità. Sostituisci l'identificatore della regione e specifica il numero di AWS account che usi per avviare le attività.

 Important

Quando si crea il ruolo IAM dell'attività, si consiglia di utilizzare le chiavi `aws:SourceAccount` o `aws:SourceArn` condition presenti nella politica di relazione di

fiducia associata al ruolo per definire ulteriormente l'ambito delle autorizzazioni ed evitare il confuso problema di sicurezza secondaria. L'uso della chiave di condizioni `aws:SourceArn` per specificare un cluster specifico non è correntemente supportato, è necessario utilizzare il carattere jolly per specificare tutti i cluster. Per saperne di più sul problema del vicedirettore confuso e su come proteggere il tuo AWS account, consulta [Il problema del vice confuso](#) nella Guida per l'utente di IAM.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "ecs-tasks.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "ArnLike": {
          "aws:SourceArn": "arn:aws:ecs:us-west-2:111122223333:*"
        },
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

Utilizza la seguente procedura per creare una policy per recuperare oggetti da Amazon S3 con una policy di esempio. Sostituisci tutti i *user input* con i valori in tuo possesso.

AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo. <https://console.aws.amazon.com/iam/>
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-task-secrets-bucket/*"
      ]
    }
  ]
}
```

6. Scegli Next (Successivo).

Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

AWS CLI

Sostituisci tutti i *user input* con i valori in tuo possesso.

1. Crea un file denominato `s3-policy.json`, con il seguente contenuto:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-task-secrets-bucket/*"
      ]
    }
  ]
}
```

2. Utilizza il seguente comando per creare la policy IAM utilizzando il file di documento della policy JSON. Sostituisci tutti i *user input* con i valori in tuo possesso.

```
aws iam create-policy \
  --policy-name taskRolePolicy \
  --policy-document file://s3-policy.json
```

Utilizzare la procedura seguente per creare il ruolo di servizio.

AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.

5. Scegli Next (Successivo).
6. Per Aggiungi autorizzazioni, cerca e scegli la policy che hai creato.
7. Scegli Next (Successivo).
8. In Nome ruolo, immetti un nome per il ruolo. Per questo esempio, digita `AmazonECSTaskS3BucketRole` per il nome del ruolo.
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

AWS CLI

1. Crea un file denominato `ecs-tasks-trust-policy.json` che contenga la policy di fiducia da utilizzare per il ruolo IAM dell'attività. Il file deve contenere quanto segue. Sostituisci l'identificatore della regione e specifica il numero di AWS account da utilizzare per l'avvio delle attività.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-task-secrets-bucket/*"
      ]
    }
  ]
}
```

2. Crea un ruolo IAM denominato `ecsTaskRole` utilizzando la policy di attendibilità creata nel passaggio precedente.

```
aws iam create-role \
  --role-name ecsTaskRole \
  --assume-role-policy-document file://ecs-tasks-trust-policy.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando. **taskRolePolicy** Sostituiscilo con il nome della policy che hai creato.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`taskRolePolicy`].Arn'
```

4. Allega la policy IAM che hai creato al ecsTaskRole ruolo. Sostituisci policy-arn con l'ARN della policy che hai creato.

```
aws iam attach-role-policy \
  --role-name ecsTaskRole \
  --policy-arn arn:aws:iam:111122223333:aws:policy/taskRolePolicy
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Usa ECS Exec	Autorizzazioni ECS Exec
Usa un'immagine da un repository Amazon ECR privato	Autorizzazioni Amazon ECR
Usa EC2 istanze (Windows e Linux)	Configurazione aggiuntiva EC2 delle istanze Amazon
Usa istanze esterne	Configurazione aggiuntiva dell'istanza esterna
Usa istanze Windows EC2	Configurazione aggiuntiva dell'istanza Amazon EC2 Windows

Autorizzazioni Amazon ECR

Le seguenti autorizzazioni sono necessarie quando si desidera utilizzare un'immagine in un repository privato. È necessario aggiungere le seguenti autorizzazioni a un ruolo IAM del processo e includere il ruolo IAM del processo nella definizione di attività. Per ulteriori informazioni, consulta [Aggiungere e rimuovere le politiche IAM nella Guida](#) per l'utente IAM.

Utilizza la seguente politica per il ruolo IAM dell'attività per aggiungere le autorizzazioni Amazon ECR richieste.

```
{
```

```
"Version": "2012-10-17",
"Statement": [
  {
    "Effect": "Allow",
    "Action": [
      "ecr:BatchGetImage",
      "ecr:GetDownloadUrlForLayer",
      "ecr:GetAuthorizationToken"
    ],
    "Resource": "*"
  }
]
```

Autorizzazioni ECS Exec

La funzionalità [ECS Exec](#) richiede un ruolo Task IAM per concedere ai contenitori le autorizzazioni necessarie per la comunicazione tra l'agente SSM gestito (agente) `execute-command` e il servizio SSM. È necessario aggiungere le seguenti autorizzazioni a un ruolo IAM del processo e includere il ruolo IAM del processo nella definizione di attività. Per ulteriori informazioni, consulta [Aggiungere e rimuovere le politiche IAM nella Guida per l'utente IAM](#).

Utilizza la seguente policy per il ruolo IAM del processo e aggiungere le autorizzazioni di SSM richieste.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssmmessages:CreateControlChannel",
        "ssmmessages:CreateDataChannel",
        "ssmmessages:OpenControlChannel",
        "ssmmessages:OpenDataChannel"
      ],
      "Resource": "*"
    }
  ]
}
```

Configurazione aggiuntiva EC2 delle istanze Amazon

Consigliamo di limitare le autorizzazioni nel ruolo dell'istanza di container all'elenco minimo delle autorizzazioni fornito nella policy IAM gestita da `AmazonEC2ContainerServiceforEC2Role`.

Le tue EC2 istanze Amazon richiedono almeno la versione `1.11.0` dell'agente container per utilizzare il task role; tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#). Se utilizzi un'AMI ottimizzata per Amazon ECS, l'istanza richiede almeno una `1.11.0-1` parte del `ecs-init` pacchetto. Se le istanze di container sono avviate dall'AMI ottimizzata per Amazon ECS più recente, contengono le versioni richieste dell'agente del container e di `ecs-init`. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Se non utilizzi l'AMI ottimizzata per Amazon ECS per le tue istanze di container, aggiungi l'`--net=host` opzione al `docker run` comando che avvia l'agente e le seguenti variabili di configurazione dell'agente per la configurazione desiderata (per ulteriori informazioni, consulta): [Configurazione dell'agente del container Amazon ECS](#)

`ECS_ENABLE_TASK_IAM_ROLE=true`

Utilizza i ruoli IAM per le attività per i container con le modalità di rete `bridge` e `default`.

`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST=true`

Utilizza i ruoli IAM per le attività per i container con la modalità di rete `host`. Questa variabile è supportata sull'agente solo a partire dalla versione `1.12.0`.

Per un esempio di comando di esecuzione, vedi [Aggiornamento manuale dell'agente container Amazon ECS \(per prodotti non ottimizzati per Amazon ECS\) AMIs](#). Dovrai inoltre impostare i seguenti comandi di rete sull'istanza del contenitore in modo che i contenitori utilizzati nelle tue attività possano recuperare le proprie credenziali: AWS

```
sudo sysctl -w net.ipv4.conf.all.route_localnet=1
sudo iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-destination 127.0.0.1:51679
sudo iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 51679
```

Per fare in modo che le regole iptables vengano conservate dopo un riavvio, devi salvarle sull'istanza di container. Puoi salvare e ripristinare le regole iptables all'avvio tramite i comandi iptables-save e iptables-restore. Per ulteriori informazioni, consulta la documentazione specifica per il tuo sistema operativo.

Per impedire ai contenitori eseguiti da attività che utilizzano la modalità di awsvpc rete di accedere alle informazioni sulle credenziali fornite al profilo dell' EC2 istanza Amazon, pur consentendo le autorizzazioni fornite dal ruolo dell'attività, imposta la variabile di configurazione dell'ECS_AWSVPC_BLOCK_IMDS agente su true nel file di configurazione dell'agente e riavvia l'agente. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Per impedire ai contenitori eseguiti da attività che utilizzano la modalità di bridge rete di accedere alle informazioni sulle credenziali fornite al profilo dell' EC2 istanza Amazon, pur consentendo le autorizzazioni fornite dal ruolo dell'attività, esegui il seguente iptables comando sulle tue istanze Amazon EC2 . Questo comando non influisce sui container nelle attività che usano la modalità di rete host o awsvpc. Per ulteriori informazioni, consulta [Modalità di rete](#).

- ```
sudo yum install -y iptables-services; sudo iptables --insert DOCKER-USER 1 --in-interface docker+ --destination 169.254.169.254/32 --jump DROP
```

Devi salvare questa iptables regola sulla tua EC2 istanza Amazon affinché sopravviva al riavvio. Quando utilizzi l'AMI ottimizzata per Amazon ECS, puoi utilizzare il comando riportato di seguito. Per gli altri sistemi operativi, consulta la relativa documentazione specifica.

```
sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now iptables
```

## Configurazione aggiuntiva dell'istanza esterna

Le istanze esterne richiedono almeno la versione 1.11.0 dell'agente contenitore per utilizzare i ruoli Task IAM; tuttavia, consigliamo di utilizzare la versione più recente dell'agente contenitore. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta [Aggiornamento dell'agente del container Amazon ECS](#). Se utilizzi l'AMI ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione 1.11.0-1 del pacchetto ecs-init. Se le istanze di container sono avviate dall'AMI ottimizzata per Amazon ECS più recente, contengono le versioni richieste dell'agente del container e di ecs-init. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Se non utilizzi l'AMI ottimizzata per Amazon ECS per le tue istanze di container, aggiungi l' --net=host opzione al docker run comando che avvia l'agente e le seguenti variabili di configurazione dell'agente per la configurazione desiderata (per ulteriori informazioni, consulta): [Configurazione dell'agente del container Amazon ECS](#)

ECS\_ENABLE\_TASK\_IAM\_ROLE=true

Utilizza i ruoli IAM per le attività per i container con le modalità di rete bridge e default.

ECS\_ENABLE\_TASK\_IAM\_ROLE\_NETWORK\_HOST=true

Utilizza i ruoli IAM per le attività per i container con la modalità di rete host. Questa variabile è supportata sull'agente solo a partire dalla versione 1.12.0.

Per un esempio di comando di esecuzione, vedi [Aggiornamento manuale dell'agente container Amazon ECS \(per prodotti non ottimizzati per Amazon ECS\) AMIs](#). Dovrai inoltre impostare i seguenti comandi di rete sull'istanza del contenitore in modo che i contenitori utilizzati nelle tue attività possano recuperare le proprie credenziali: AWS

```
sudo sysctl -w net.ipv4.conf.all.route_localnet=1
sudo iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-destination 127.0.0.1:51679
sudo iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 51679
```

Per fare in modo che le regole iptables vengano conservate dopo un riavvio, devi salvarle sull'istanza di container. Puoi salvare e ripristinare le regole iptables all'avvio tramite i comandi iptables-save e iptables-restore. Per ulteriori informazioni, consulta la documentazione specifica per il tuo sistema operativo.

Configurazione aggiuntiva dell'istanza Amazon EC2 Windows

#### Important

Questo vale solo per i contenitori Windows EC2 che utilizzano ruoli di attività.

Il ruolo di attività con funzionalità di Windows richiede una configurazione aggiuntiva EC2.

- Quando avvii le istanze di container, devi configurare l'opzione `-EnableTaskIAMRole` nello script di dati utente dell'istanza di container. La `EnableTaskIAMRole` attiva la funzionalità ruoli IAM dei processi per i processi. Ad esempio:

```
<powershell>
Import-Module ECSTools
Initialize-ECSAgent -Cluster 'windows' -EnableTaskIAMRole
</powershell>
```

- Devi eseguire il bootstrap del container con i comandi di rete forniti in [Script di bootstrap per contenitori Amazon ECS](#).
- Devi creare un ruolo IAM e una policy per i processi. Per ulteriori informazioni, consulta [Creazione del ruolo IAM dell'attività](#).
- I ruoli IAM per il provider di credenziali dei processi utilizzano la porta 80 nell'istanza di container. Pertanto, se configuri i ruoli IAM per i processi nell'istanza di container, i container non potranno utilizzare la porta 80 come porta host in qualsiasi mappatura delle porte. Per esporre i container sulla porta 80, consigliamo di configurare un servizio che utilizzi il bilanciamento del carico. Puoi utilizzare la porta 80 sul load balancer. In questo modo, il traffico può essere instradato a un'altra porta host nelle istanze di container. Per ulteriori informazioni, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).
- Se l'istanza di Windows viene riavviata, è necessario eliminare l'interfaccia proxy e inizializzare nuovamente l'agente del container di Amazon ECS per ripristinare il proxy delle credenziali.

## Script di bootstrap per contenitori Amazon ECS

Prima che i container possano accedere al proxy di credenziali nell'istanza di container per ottenere le credenziali, devi eseguire il bootstrap del container con i comandi di rete richiesti. Il seguente script di esempio di codice deve essere eseguito nei container quando si avviano.

### Note

Non è necessario eseguire questo script quando si utilizza la modalità di rete `awsvpc` su Windows.

Se esegui container Windows che includono Powershell, utilizza lo script seguente:

```
Copyright Amazon.com Inc. or its affiliates. All Rights Reserved.
```

```
#
Licensed under the Apache License, Version 2.0 (the "License"). You may
not use this file except in compliance with the License. A copy of the
License is located at
#
http://aws.amazon.com/apache2.0/
#
or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.

$gateway = (Get-NetRoute | Where { $_.DestinationPrefix -eq '0.0.0.0/0' } | Sort-Object
 RouteMetric | Select NextHop).NextHop
$ifIndex = (Get-NetAdapter -InterfaceDescription "Hyper-V Virtual Ethernet*" | Sort-
 Object | Select ifIndex).ifIndex
New-NetRoute -DestinationPrefix 169.254.170.2/32 -InterfaceIndex $ifIndex -NextHop
 $gateway -PolicyStore ActiveStore # credentials API
New-NetRoute -DestinationPrefix 169.254.169.254/32 -InterfaceIndex $ifIndex -NextHop
 $gateway -PolicyStore ActiveStore # metadata API
```

Se esegui container Windows con solo Command shell, utilizza lo script seguente:

```
Copyright Amazon.com Inc. or its affiliates. All Rights Reserved.
#
Licensed under the Apache License, Version 2.0 (the "License"). You may
not use this file except in compliance with the License. A copy of the
License is located at
#
http://aws.amazon.com/apache2.0/
#
or in the "license" file accompanying this file. This file is distributed
on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
express or implied. See the License for the specific language governing
permissions and limitations under the License.

for /f "tokens=1" %i in ('netsh interface ipv4 show interfaces ^| findstr /x /r
 ".*vEthernet.*"') do set interface=%i
for /f "tokens=3" %i in ('netsh interface ipv4 show addresses %interface% ^| findstr /
 x /r ".*Default.Gateway.*"') do set gateway=%i
netsh interface ipv4 add route prefix=169.254.170.2/32 interface="%interface%"
 nexthop="%gateway%" store=active # credentials API
```



```
netsh interface ipv4 add route prefix=169.254.169.254/32 interface="%interface%"
nexthop="%gateway%" store=active # metadata API
```

## Ruolo IAM delle istanze di container Amazon ECS

Le istanze di container Amazon ECS, incluse le istanze Amazon EC2 e quelle esterne, eseguono l'agente container Amazon ECS e richiedono un ruolo IAM affinché il servizio sappia che l'agente appartiene a te. Prima di avviare le istanze di container e registrarle in un cluster, devi creare un ruolo IAM per le istanze da utilizzare. Il ruolo viene creato nell'account che usi per accedere alla console o eseguire i comandi. AWS CLI

### Important

Se registri le istanze esterne nel cluster, il ruolo IAM utilizzato richiede anche le autorizzazioni di Systems Manager. Per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#).

Amazon ECS fornisce la policy IAM gestita da `AmazonEC2ContainerServiceforEC2Role` che contiene le autorizzazioni necessarie per utilizzare il set completo di funzionalità di Amazon ECS. Questa policy gestita può essere associata a un ruolo IAM e alle istanze del container. In alternativa, è possibile utilizzare la policy gestita come guida durante la creazione di una policy personalizzata da utilizzare. Il ruolo dell'istanza del contenitore fornisce le autorizzazioni necessarie affinché l'agente contenitore Amazon ECS e il daemon Docker possano effettuare chiamate per tuo conto. AWS APIs Per ulteriori informazioni sulla policy gestita, consulta [EC2ContainerServiceforEC2Ruolo di Amazon](#).

Amazon ECS supporta il lancio di istanze di container con una maggiore densità ENI utilizzando i tipi di istanze Amazon EC2 supportati. Quando utilizzi questa funzionalità, ti consigliamo di creare 2 ruoli di istanza di container. Abilita l'impostazione `awsvpctrunking` dell'account per un ruolo e utilizza quel ruolo per attività che richiedono il trunking ENI. Per informazioni sull'impostazione dell'`awsvpctrunkingaccount`, vedere. [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#)

Creare il ruolo dell'istanza del contenitore

### Important

Se registri istanze esterne nel cluster, consulta [Ruolo IAM di Amazon ECS Anywhere](#).

Puoi creare manualmente il ruolo e collegare la policy IAM gestita per le istanze di container in modo da consentire ad Amazon ECS di aggiungere le autorizzazioni per funzionalità e miglioramenti futuri man mano che vengono introdotti. Utilizza la seguente procedura per allegare la policy IAM gestita, se necessario.

## AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Servizio o caso d'uso, scegli Elastic Container Service, quindi scegli il caso d'uso EC2 Role for Elastic Container Service.
5. Scegli Next (Successivo).
6. Nella sezione Politiche di autorizzazione, verifica che sia selezionata la politica di Amazon EC2 ContainerServicefor EC2 Role.

### Important

La policy gestita di Amazon EC2 ContainerServicefor EC2 Role deve essere allegata al ruolo IAM dell'istanza del contenitore, altrimenti riceverai un errore durante l'utilizzo AWS Management Console di per creare cluster.

7. Scegli Next (Successivo).
8. Per il nome del ruolo, inserisci ecsInstanceRole
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

## AWS CLI

Sostituisci tutti i *user input* con i valori in tuo possesso.

1. Crea un file denominato `instance-role-trust-policy.json` con i seguenti contenuti.

```
{
 "Version": "2012-10-17",
```

```
"Statement": [
 {
 "Effect": "Allow",
 "Principal": { "Service": "ec2.amazonaws.com"},
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Utilizza il comando seguente per creare il ruolo IAM dell'istanza utilizzando il documento di policy di fiducia.

```
aws iam create-role \
 --role-name ecsInstanceRole \
 --assume-role-policy-document file://instance-role-trust-policy.json
```

3. Crea un profilo di istanza denominato `ecsInstanceRole-profile` utilizzando il [create-instance-profile](#) comando.

```
aws iam create-instance-profile --instance-profile-name ecsInstanceRole-profile
```

#### Example response

```
{
 "InstanceProfile": {
 "InstanceId": "AIPAJTLBPJLEGREXAMPLE",
 "Roles": [],
 "CreateDate": "2022-04-12T23:53:34.093Z",
 "InstanceProfileName": "ecsInstanceRole-profile",
 "Path": "/",
 "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole-profile"
 }
}
```

4. Aggiungi il ruolo *ecsInstanceRole* al profilo dell'istanza *ecsInstanceRole-profile*.

```
aws iam add-role-to-instance-profile \
 --instance-profile-name ecsInstanceRole-profile \
 --role-name ecsInstanceRole
```

5. Allega la politica `AmazonEC2ContainerServiceRoleForEC2Role` gestita al ruolo utilizzando il comando seguente.

```
aws iam attach-role-policy \
 --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
 --role-name ecsInstanceRole
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

| Funzionalità                                                      | Autorizzazioni aggiuntive                                                     |
|-------------------------------------------------------------------|-------------------------------------------------------------------------------|
| Amazon ECR ha l'immagine del contenitore                          | <a href="#">Autorizzazioni Amazon ECR</a>                                     |
| Fai in modo che CloudWatch Logs monitori le istanze dei container | <a href="#">Monitoraggio delle autorizzazioni delle istanze dei container</a> |
| Ospita i file di configurazione in un bucket Amazon S3            | <a href="#">Accesso in sola lettura ad Amazon S3</a>                          |

## Autorizzazioni Amazon ECR

Il ruolo dell'istanza di container Amazon ECS che usi con le tue istanze di container deve avere le seguenti autorizzazioni della policy IAM per Amazon ECR.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecr:BatchCheckLayerAvailability",
 "ecr:BatchGetImage",
 "ecr:GetDownloadUrlForLayer",
 "ecr:GetAuthorizationToken"
],
 "Resource": "*"
 }
]
}
```

```
}
```

Se utilizzi la policy gestita `AmazonEC2ContainerServiceforEC2Role` per le tue istanze di container, il tuo ruolo dispone già delle opportune autorizzazioni. Per verificare se il tuo ruolo supporta Amazon ECR, consulta [Ruolo IAM per le istanze di container di Amazon ECS](#) nella Guida per lo sviluppatore di Amazon Elastic Container.

### Accesso in sola lettura ad Amazon S3

Archiviare le informazioni di configurazione in un bucket privato in Amazon S3 e concedere l'accesso in sola lettura al ruolo IAM dell'istanza di container è un modo sicuro e conveniente per permettere la configurazione delle istanze di container all'ora di avvio. Puoi archiviare una copia del tuo `ecs.config` file in un bucket privato, utilizzare i dati EC2 utente di Amazon per installarlo AWS CLI e quindi copiare le informazioni di configurazione all'/etc/ecs/ecs.configavvio dell'istanza.

Per ulteriori informazioni sulla creazione di un file `ecs.config`, sull'archiviazione di tale file in Amazon S3 e sull'avvio di istanze con questa configurazione, consulta [Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3](#).

Puoi utilizzare il seguente AWS CLI comando per consentire l'accesso in sola lettura ad Amazon S3 per il tuo ruolo di istanza di contenitore. `ecsInstanceRole` Sostituiscilo con il nome del ruolo che hai creato.

```
aws iam attach-role-policy \
 --role-name ecsInstanceRole \
 --policy-arn arn:aws:iam::aws:policy/AmazonS3ReadOnlyAccess
```

Puoi anche utilizzare la console IAM per aggiungere Amazon S3 read-only access (`AmazonS3ReadOnlyAccess`) al tuo ruolo. Per ulteriori informazioni, consulta [Aggiornamento delle autorizzazioni per un ruolo nella Guida per l'utente](#).AWS Identity and Access Management

### Monitoraggio delle autorizzazioni delle istanze dei container

Prima che le istanze del contenitore possano inviare dati di log a CloudWatch Logs, devi creare una policy IAM per consentire all'agente Amazon ECS di scrivere i log delle applicazioni del cliente CloudWatch (normalmente gestiti tramite il driver). `awslogs` Dopo aver creato la policy, allegala a `ecsInstanceRole`

## AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents",
 "logs:DescribeLogStreams"
],
 "Resource": ["arn:aws:logs:*:*:*"]
 }
]
}
```

6. Scegli Next (Successivo).

### Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al ruolo dell'istanza del contenitore. Per informazioni su come allegare la policy al ruolo, consulta [Aggiornamento delle autorizzazioni per un ruolo](#) nella Guida per l'AWS Identity and Access Management utente.

## AWS CLI

1. Crea un file denominato `instance-cw-logs.json`, con il seguente contenuto:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogGroup",
 "logs:CreateLogStream",
 "logs:PutLogEvents",
 "logs:DescribeLogStreams"
],
 "Resource": ["arn:aws:logs:*:*:*"]
 }
]
}
```

2. Usa il comando seguente per creare la policy IAM utilizzando il file del documento della policy JSON.

```
aws iam create-policy \
 --policy-name cwlogspolicy \
 --policy-document file://instance-cw-logs.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando.  
*cwlogspolicy* Sostituiscilo con il nome della policy che hai creato.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cwlogspolicy`].Arn'
```

4. Usa il comando seguente per allegare la policy al ruolo IAM dell'istanza del contenitore utilizzando la policy ARN.

```
aws iam attach-role-policy \
 --role-name ecsInstanceRole \
 --policy-arn arn:aws:iam:111122223333:aws:policy/cwlogspolicy
```

## Ruolo IAM di Amazon ECS Anywhere

Quando registri un server o una macchina virtuale (VM) locale nel tuo cluster, il server o la macchina virtuale richiede un ruolo IAM con cui comunicare. AWS APIs Devi creare questo ruolo IAM solo una volta per ogni account. AWS Tuttavia, questo ruolo IAM deve essere associato a ogni server o macchina virtuale registrato in un cluster. Questo ruolo è il `ECSAnywhereRole`. Puoi creare questo ruolo manualmente. In alternativa, Amazon ECS può creare il ruolo per tuo conto quando registri un'istanza esterna nella AWS Management Console. Puoi utilizzare la ricerca nella console IAM per cercare `ecsAnywhereRole` e vedere se il tuo account ha già il ruolo. Per ulteriori informazioni, consulta la [ricerca nella console IAM](#) nella guida per l'utente IAM.

AWS fornisce due policy IAM gestite che possono essere utilizzate durante la creazione del ruolo IAM ECS Anywhere, le `AmazonEC2ContainerServiceforEC2Role` policy `AmazonSSMManagedInstanceCore` e. La policy `AmazonEC2ContainerServiceforEC2Role` include autorizzazioni che probabilmente forniscono più accesso del necessario. Pertanto, a seconda del caso d'uso specifico, si consiglia di creare una policy personalizzata aggiungendo solo le autorizzazioni richieste da tale policy. Per ulteriori informazioni, consultare [Ruolo IAM delle istanze di container Amazon ECS](#).

Il ruolo IAM di esecuzione del processo concede all'agente del container di Amazon ECS e agli agenti Fargate l'autorizzazione per effettuare chiamate API AWS per tuo conto. Quando viene utilizzato un ruolo IAM di esecuzione del processo, è necessario specificarlo nella definizione di attività. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Il ruolo di esecuzione del processo è obbligatorio se si applica una delle seguenti condizioni:

- Stai inviando i log dei container a CloudWatch Logs utilizzando il `awslogs` driver di registro.
- La definizione di attività specifica un'immagine del container ospitata in un repository privato Amazon ECR. Tuttavia, se il `ECSAnywhereRole` ruolo associato alla tua istanza esterna include anche le autorizzazioni necessarie per estrarre immagini da Amazon ECR, non è necessario che il ruolo di esecuzione dell'attività le includa.



## Creazione del ruolo Amazon ECS Anywhere

Sostituisci tutto **user input** con le tue informazioni.

1. Crea un file locale denominato `ssm-trust-policy.json` con la seguente politica di attendibilità.

```
{
 "Version": "2012-10-17",
 "Statement": {
 "Effect": "Allow",
 "Principal": {"Service": [
 "ssm.amazonaws.com"
]},
 "Action": "sts:AssumeRole"
 }
}
```

2. Crea il ruolo e allega la politica di fiducia utilizzando il AWS CLI comando seguente.

```
aws iam create-role --role-name ecsAnywhereRole --assume-role-policy-document
file://ssm-trust-policy.json
```

3. Allega le politiche AWS gestite utilizzando il comando seguente.

```
aws iam attach-role-policy --role-name ecsAnywhereRole --policy-arn
arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
aws iam attach-role-policy --role-name ecsAnywhereRole --policy-arn
arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
```

Puoi anche utilizzare il flusso di lavoro delle policy di fiducia personalizzate IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando politiche di fiducia personalizzate \(console\)](#) nella Guida per l'utente IAM.

## Ruolo IAM dell'infrastruttura Amazon ECS

Un ruolo IAM dell'infrastruttura Amazon ECS consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei cluster per tuo conto e viene utilizzato quando:

- Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività.

- Desideri utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect.
- Vuoi creare gruppi target Amazon VPC Lattice.

Quando Amazon ECS assume questo ruolo per intraprendere azioni per tuo conto, gli eventi saranno visibili in AWS CloudTrail. Se Amazon ECS utilizza il ruolo per gestire i volumi Amazon EBS collegati alle tue attività, il CloudTrail log `roleSessionName` sarà `ECSTaskVolumesForEBS`. Se il ruolo viene utilizzato per crittografare il traffico tra i servizi Service Connect, il CloudTrail registro `roleSessionName` sarà `ECSServiceConnectForTLS`. Se il ruolo viene utilizzato per creare gruppi target per VPC Lattice, il CloudTrail registro `roleSessionName` sarà `ECSNetworkingWithVPCLattice`. È possibile utilizzare questo nome per cercare eventi nella CloudTrail console filtrando in base al nome utente.

Amazon ECS fornisce policy gestite che contengono le autorizzazioni necessarie per gli allegati di volume e TLS. Per ulteriori informazioni, consulta [Amazon ECSInfrastructure RolePolicyForVolumes](#), [Amazon ECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity](#), [Amazon ECSInfrastructure RolePolicyForVpcLattice](#) nella AWS Managed Policy Reference Guide.

## Creazione del ruolo dell'infrastruttura Amazon ECS

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `ecs-infrastructure-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AllowAccessToECSForInfrastructureManagement",
 "Effect": "Allow",
 "Principal": {
 "Service": "ecs.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Utilizzate il AWS CLI comando seguente per creare un ruolo denominato `ecsInfrastructureRole` utilizzando la politica di fiducia creata nel passaggio precedente.

```
aws iam create-role \
 --role-name ecsInfrastructureRole \
 --assume-role-policy-document file://ecs-infrastructure-trust-policy.json
```

3. A seconda del caso d'uso, AWS allega il `AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity` gestore `AmazonECSInfrastructureRolePolicyForVolumes` o la `AmazonECSInfrastructureRolePolicyForVpcLattice` policy al `ecsInfrastructureRole` ruolo.

```
aws iam attach-role-policy \
 --role-name ecsInfrastructureRole \
 --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSInfrastructureRolePolicyForVolumes
```

```
aws iam attach-role-policy \
 --role-name ecsInfrastructureRole \
 --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity
```

Puoi anche utilizzare il flusso di lavoro Custom Trust Policy della console IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando politiche di fiducia personalizzate \(console\)](#) nella Guida per l'utente IAM.

#### Important

Se il ruolo dell'infrastruttura ECS viene utilizzato da Amazon ECS per gestire i volumi Amazon EBS collegati alle tue attività, verifica quanto segue prima di interrompere le attività che utilizzano volumi Amazon EBS.

- Il ruolo non viene eliminato.
- La policy di fiducia per il ruolo non viene modificata per rimuovere Amazon ECS access (`ecs.amazonaws.com`).
- La policy gestita `AmazonECSInfrastructureRolePolicyForVolumes` non viene rimossa. Se devi modificare le autorizzazioni del ruolo, conservale almeno

`ec2:DetachVolume` e `ec2:DescribeVolumes` per l'eliminazione di un volume.  
`ec2:DeleteVolume`

L'eliminazione o la modifica del ruolo prima di interrompere le attività con volumi Amazon EBS collegati comporterà il blocco delle attività e la mancata eliminazione DEPROVISIONING dei volumi Amazon EBS associati. Amazon ECS riproverà automaticamente a intervalli regolari per interrompere l'attività ed eliminare il volume fino al ripristino delle autorizzazioni necessarie. Puoi visualizzare lo stato degli allegati al volume di un'attività e il motivo dello stato associato utilizzando l'API. [DescribeTasks](#)

Dopo aver creato il file, devi concedere all'utente l'autorizzazione a passare il ruolo ad Amazon ECS.

Autorizzazione a trasferire il ruolo di infrastruttura ad Amazon ECS

Per utilizzare un ruolo IAM dell'infrastruttura ECS, devi concedere all'utente l'autorizzazione a passare il ruolo ad Amazon ECS. Allega la seguente `iam:PassRole` autorizzazione al tuo utente. Sostituiscilo *`ecsInfrastructureRole`* con il nome del ruolo di infrastruttura che hai creato.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "iam:PassRole",
 "Effect": "Allow",
 "Resource": ["arn:aws:iam::*:role/ecsInfrastructureRole"],
 "Condition": {
 "StringEquals": {"iam:PassedToService": "ecs.amazonaws.com"}
 }
 }
]
}
```

Per ulteriori informazioni `iam:PassRole` e sull'aggiornamento delle autorizzazioni per il tuo utente, consulta [Concessione a un utente delle autorizzazioni per passare un ruolo a un AWS servizio](#) e [Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente](#). AWS Identity and Access Management

## Ruolo CodeDeploy IAM di Amazon ECS

Prima di poter utilizzare il tipo di distribuzione CodeDeploy blu/verde con Amazon ECS, il CodeDeploy servizio necessita delle autorizzazioni per aggiornare il servizio Amazon ECS per tuo conto. Queste autorizzazioni sono fornite dal ruolo IAM (). CodeDeploy `ecsCodeDeployRole`

### Note

Gli utenti richiedono inoltre le autorizzazioni per l'uso CodeDeploy; queste autorizzazioni sono descritte in. [Autorizzazioni IAM richieste](#)

Sono disponibili due policy gestite. Per ulteriori informazioni, consulta una delle seguenti voci nella AWS Managed Policy Reference Guide:

- [AWSCodeDeployRoleForECS](#): concede CodeDeploy l'autorizzazione ad aggiornare qualsiasi risorsa utilizzando l'azione associata.
- [AWSCodeDeployRoleForECSLimited](#)- concede autorizzazioni CodeDeploy più limitate.

### Creare il ruolo CodeDeploy

Puoi utilizzare le seguenti procedure per creare un CodeDeploy ruolo per Amazon ECS

#### AWS Management Console

Per creare il ruolo di servizio per CodeDeploy (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Servizio o caso d'uso, scegli CodeDeploy, quindi scegli il caso d'uso CodeDeploy - ECS.
5. Scegli Next (Successivo).
6. Nella sezione Allega criteri di autorizzazione, assicurati che sia selezionata la politica `AWSCodeDeployRoleForECS`.
7. Scegli Next (Successivo).
8. Per Nome ruolo, inserisci `ecsCodeDeploy Ruolo`.

## 9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

### AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `codedeploy-trust-policy.json` che contenga la policy di fiducia da utilizzare per il ruolo CodeDeploy IAM.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": ["codedeploy.amazonaws.com"]
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Crea un ruolo IAM denominato `ecsCodeDeployRole` utilizzando la policy di attendibilità creata nel passaggio precedente.

```
aws iam create-role \
 --role-name ecsCodeDeployRole \
 --assume-role-policy-document file://codedeploy-trust-policy.json
```

3. Allega la policy `AWSCodeDeployRoleForECS` o la policy `AWSCodeDeployRoleForECSLimited` gestita al `ecsTaskRole` ruolo.

```
aws iam attach-role-policy \
 --role-name ecsCodeDeployRole \
 --policy-arn arn:aws:iam::aws:policy/AWSCodeDeployRoleForECS
```

```
aws iam attach-role-policy \
 --role-name ecsCodeDeployRole \
 --policy-arn arn:aws:iam::aws:policy/AWSCodeDeployRoleForECSLimited
```

Quando le attività del tuo servizio richiedono un ruolo di esecuzione dell'attività, devi aggiungere al ruolo l'`iam:PassRole` autorizzazione per ogni ruolo di esecuzione dell'attività o sostituzione del ruolo dell'attività al CodeDeploy ruolo come politica.

### Autorizzazioni relative al ruolo di esecuzione delle attività

Quando le attività del tuo servizio richiedono un ruolo di esecuzione dell'attività, devi aggiungere al ruolo l'`iam:PassRole` autorizzazione per ogni ruolo di esecuzione dell'attività o sovrascrittura del ruolo dell'attività al CodeDeploy ruolo come politica. Per ulteriori informazioni, consultare [Ruolo IAM di esecuzione di attività Amazon ECS](#) e [Ruolo IAM dell'attività Amazon ECS](#). Quindi, si allega tale politica al ruolo CodeDeploy

### Creare la policy

#### AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsCodeDeployRole>"]
 }
]
}
```

6. Scegli Next (Successivo).

 Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al CodeDeploy ruolo. Per informazioni su come allegare la politica al ruolo, consulta [Aggiornare le autorizzazioni per un ruolo](#) nella Guida per l'AWS Identity and Access Management utente.

## AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `blue-green-iam-passrole.json`, con il seguente contenuto:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsCodeDeployRole>"]
 }
]
}
```

2. Usa il seguente comando per creare la policy IAM utilizzando il file di documento della policy JSON.

```
aws iam create-policy \
```



```
--policy-name cdTaskExecutionPolicy \
--policy-document file://blue-green-iam-passrole.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cdTaskExecutionPolicy`].Arn'
```

4. Usa il comando seguente per allegare la policy al ruolo CodeDeploy IAM.

```
aws iam attach-role-policy \
--role-name ecsCodedeployRole \
--policy-arn arn:aws:iam:111122223333:aws:policy/cdTaskExecutionPolicy
```

## Ruolo EventBridge IAM di Amazon ECS

Prima di poter utilizzare le attività pianificate di Amazon ECS con EventBridge regole e obiettivi, il EventBridge servizio necessita delle autorizzazioni per eseguire le attività di Amazon ECS per tuo conto. Queste autorizzazioni vengono fornite dal ruolo IAM di EventBridge (`ecsEventsRole`).

Di seguito viene mostrata la policy `AmazonEC2ContainerServiceEventsRole`.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": ["ecs:RunTask"],
 "Resource": ["*"]
 },
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["*"],
 "Condition": {
 "StringLike": {"iam:PassedToService": "ecs-tasks.amazonaws.com"}
 }
 },
 {
 "Effect": "Allow",
 "Action": "ecs:TagResource",
 "Resource": "*",
 }
]
}
```

```

 "Condition": {
 "StringEquals": {
 "ecs:CreateAction": ["RunTask"]
 }
 }
 }
}
]
}

```

Se le tue attività pianificate richiedono l'uso del ruolo di esecuzione dell'attività, di un ruolo di attività o dell'override di un ruolo dell'attività, devi aggiungere `iam:PassRole` le autorizzazioni per ogni ruolo di esecuzione dell'attività, ruolo dell'attività o sostituzione del ruolo dell'attività al ruolo IAM. EventBridge Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

#### Note

Specifica l'ARN completo del ruolo di esecuzione delle attività o della sostituzione del ruolo attività.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
 }
]
}

```

Puoi scegliere di lasciare che AWS Management Console crei il EventBridge ruolo automaticamente quando configuri un'attività pianificata. Per ulteriori informazioni, consulta [Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS](#).

## Creazione del EventBridge ruolo

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `eventbridge-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "events.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Utilizza il comando seguente per creare un ruolo IAM denominato `ecsEventsRole` utilizzando la policy di fiducia creata nel passaggio precedente.

```
aws iam create-role \
 --role-name ecsEventsRole \
 --assume-role-policy-document file://eventbridge-trust-policy.json
```

3. Associa il AWS gestore `AmazonEC2ContainerServiceEventsRole` al `ecsEventsRole` ruolo utilizzando il comando seguente.

```
aws iam attach-role-policy \
 --role-name ecsEventsRole \
 --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceEventsRole
```

Puoi anche utilizzare la Custom Trust Policy workflow (<https://console.aws.amazon.com/iam/>) della console IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando politiche di fiducia personalizzate \(console\)](#) nella Guida per l'utente IAM.

## Collegamento di una policy al ruolo `ecsEventsRole`

È possibile utilizzare le seguenti procedure per aggiungere le autorizzazioni per il ruolo di esecuzione dell'attività al ruolo EventBridge IAM.

## AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
 }
]
}
```

6. Scegli Next (Successivo).

### Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.

8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al EventBridge ruolo. Per informazioni su come allegare la politica al ruolo, consulta [Aggiornare le autorizzazioni per un ruolo](#) nella Guida per l'AWS Identity and Access Management utente.

## AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `ev-iam-passrole.json`, con il seguente contenuto:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
 }
]
}
```

2. Usa il seguente AWS CLI comando per creare la policy IAM utilizzando il file di documento della policy JSON.

```
aws iam create-policy \
 --policy-name eventsTaskExecutionPolicy \
 --policy-document file://ev-iam-passrole.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`eventsTaskExecutionPolicy`].Arn'
```

4. Usa il comando seguente per collegare la policy al ruolo EventBridge IAM utilizzando la policy ARN.

```
aws iam attach-role-policy \
 --role-name ecsEventsRole \
```

```
--policy-arn arn:aws:iam:111122223333:aws:policy/eventsTaskExecutionPolicy
```

## Autorizzazioni necessarie per la console Amazon ECS

Seguendo la best practice per concedere il privilegio minimo, è possibile utilizzare la policy gestita di AmazonECS\_FullAccess come modello per la creazione di policy personalizzate. In questo modo, è possibile rimuovere o aggiungere autorizzazioni da e verso le policy gestite in base ai requisiti specifici. Per ulteriori informazioni, consulta [AmazonECS\\_FullAccess](#) nel Managed Policy Reference.AWS

### Autorizzazioni per la creazione di ruoli IAM

Le operazioni seguenti richiedono autorizzazioni aggiuntive per completare l'operazione:

- Registrazione di un'istanza esterna: per ulteriori informazioni, consulta [Ruolo IAM di Amazon ECS Anywhere](#)
- Registrazione di una definizione di attività: per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#)
- Creazione di una EventBridge regola da utilizzare per la pianificazione delle attività: per ulteriori informazioni, consulta [Ruolo EventBridge IAM di Amazon ECS](#)

Puoi aggiungere queste autorizzazioni creando un ruolo in IAM prima di utilizzarle nella console Amazon ECS. Se non crei i ruoli, la console Amazon ECS li crea per tuo conto.

### Autorizzazioni necessarie per registrare un'istanza esterna in un cluster

Hai bisogno di autorizzazioni aggiuntive quando registri un'istanza esterna in un cluster e desideri creare un nuovo ruolo di istanza esterna (ecsExternalInstanceRole).

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- iam: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate.
- ssm: consente ai principali di registrare l'istanza esterna con Systems Manager.

 Note

Per scegliere un ruolo `ecsExternalInstanceRole` esistente, devi disporre delle autorizzazioni `iam:GetRole` e `iam:PassRole`.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `ecsExternalInstanceRole`.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:CreateInstanceProfile",
 "iam:AddRoleToInstanceProfile",
 "iam:ListInstanceProfilesForRole",
 "iam:GetRole"
],
 "Resource": "arn:aws:iam::*:role/ecsExternalInstanceRole"
 },
 {
 "Effect": "Allow",
 "Action": ["iam:PassRole", "ssm:CreateActivation"],
 "Resource": "arn:aws:iam::*:role/ecsExternalInstanceRole"
 }
]
}
```

## Autorizzazioni necessarie per la registrazione di una definizione di attività

Hai bisogno di autorizzazioni aggiuntive quando registri una definizione di attività e desideri creare un nuovo ruolo di esecuzione di attività (`ecsTaskExecutionRole`).

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `iam`: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate.

 Note

Per scegliere un ruolo `ecsTaskExecutionRole` esistente, devi disporre dell'autorizzazione `iam:GetRole`.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `ecsTaskExecutionRole`.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:GetRole"
],
 "Resource": "arn:aws:iam::*:role/ecsTaskExecutionRole"
 }
]
}
```

## Autorizzazioni necessarie per creare una EventBridge regola per le attività pianificate

Hai bisogno di autorizzazioni aggiuntive quando pianifichi un'attività e desideri creare un nuovo ruolo CloudWatch Events role (). `ecsEventsRole`

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `iam`: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate e di permettere ad Amazon ECS di trasmettere il ruolo ad altri servizi affinché lo assumano.

 Note

Per scegliere un ruolo `ecsEventsRole` esistente, devi disporre delle autorizzazioni `iam:GetRole` e `iam:PassRole`.



La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `ecsEventsRole`.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:GetRole",
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::*:role/ecsEventsRole"
 }
]
}
```

## Autorizzazioni necessarie per visualizzare le distribuzioni dei servizi

Se si segue la best practice di concessione del privilegio minimo, è necessario aggiungere autorizzazioni aggiuntive per visualizzare le distribuzioni dei servizi nella console.

È necessario accedere alle seguenti azioni:

- `ListServiceDeployments`
- `DescribeServiceDeployments`
- `DescribeServiceRevisions`

È necessario accedere alle seguenti risorse:

- Servizio
- Distribuzione del servizio
- Revisione del servizio

La seguente politica di esempio contiene le autorizzazioni richieste e limita le azioni a un servizio specificato.

Sostituisci `accountcluster-name`, e `service-name` con i tuoi valori.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:ListServiceDeployments",
 "ecs:DescribeServiceDeployments",
 "ecs:DescribeServiceRevisions"
],
 "Resource": [
 "arn:aws:ecs:us-east-1:123456789012:service/cluster-name/service-name",
 "arn:aws:ecs:us-east-1:123456789012:service-deployment/cluster-name/
service-name/*",
 "arn:aws:ecs:us-east-1:123456789012:service-revision/cluster-name/service-
name/*"
]
 }
]
}
```

## Autorizzazioni richieste per la console Amazon ECS con AWS CloudFormation

Prima di AWS Management Console utilizzarlo per creare le tue risorse, dovrai assicurarti di disporre delle autorizzazioni IAM corrette. Per informazioni su come configurare le autorizzazioni per la console Amazon ECS in generale, consulta [Autorizzazioni richieste per la console Amazon ECS](#).

La console Amazon ECS è alimentata da AWS CloudFormation e richiede autorizzazioni IAM aggiuntive nei seguenti casi:

- Creazione di un cluster
- Creazione di un servizio
- Creazione di un provider di capacità

Puoi creare una policy per le autorizzazioni aggiuntive e quindi collegarle al ruolo IAM che utilizzi per accedere alla console. Per ulteriori informazioni, consulta [Creazione di policy IAM](#) nella Guida per l'utente di IAM.

## Autorizzazioni necessarie per creare un cluster

Quando crei un cluster nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. AWS CloudFormation

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack AWS CloudFormation . Ciò è necessario quando si creano cluster Amazon ECS tramite la AWS Management Console e la successiva gestione di tali cluster.
- `ssm`— Consente di AWS CloudFormation fare riferimento alle più recenti AMI ottimizzate per Amazon ECS. Ciò è necessario quando si creano cluster Amazon ECS utilizzando. AWS Management Console

La seguente policy contiene le AWS CloudFormation autorizzazioni richieste e limita le azioni alle risorse create nella console Amazon ECS.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cloudformation:CreateStack",
 "cloudformation>DeleteStack",
 "cloudformation:DescribeStack*",
 "cloudformation:UpdateStack"
],
 "Resource": [
 "arn:*:cloudformation:*:*:stack/Infra-ECS-Cluster-*"
]
 },
 {
 "Effect": "Allow",
 "Action": "ssm:GetParameters",
 "Resource": [
 "arn:aws:ssm:*:*:parameter/aws/service/ecs/optimized-ami/amazon-linux-2*/*",
 "arn:aws:ssm:*:*:parameter/aws/service/ecs/optimized-ami/amazon-linux-2023*/*"
]
 }
]
}
```

```
]
}
```

Se non hai creato il ruolo dell'istanza del contenitore Amazon ECS (`ecsInstanceRole`) e stai creando un cluster che utilizza EC2 istanze Amazon, la console creerà il ruolo per tuo conto.

Inoltre, se si utilizzano i gruppi Auto Scaling, sono necessarie autorizzazioni aggiuntive in modo che la console possa aggiungere tag ai gruppi di auto scaling quando si utilizza la funzionalità di auto scaling del cluster.

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- **autoscaling**— Consente alla console di etichettare il gruppo Amazon EC2 Auto Scaling. Ciò è necessario per gestire i gruppi di Amazon EC2 auto scaling quando si utilizza la funzionalità di scalabilità automatica del cluster. Il tag è il tag gestito da ECS che la console aggiunge in automatico al gruppo per indicare che è stato creato nella console.
- **iam**: consente ai principali di elencare i ruoli IAM e le relative policy associate. I responsabili possono anche elencare i profili di istanza disponibili per le tue EC2 istanze Amazon.

La policy seguente contiene le autorizzazioni IAM necessarie e limita le operazioni al ruolo `ecsInstanceRole`.

Le autorizzazioni di dimensionamento automatico non sono limitate.

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:CreateInstanceProfile",
 "iam:AddRoleToInstanceProfile",
 "iam:ListInstanceProfilesForRole",
 "iam:GetRole"
],
 "Resource": "arn:aws:iam::*:role/ecsInstanceRole"
 },
 {
 "Effect": "Allow",
```

```

 "Action": "autoscaling:CreateOrUpdateTags",
 "Resource": "*"
 }
]
}

```

## Autorizzazioni necessarie per creare un servizio

Quando crei un servizio nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. AWS CloudFormation Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack AWS CloudFormation . Ciò è necessario quando si creano servizi Amazon ECS utilizzando la AWS Management Console e la successiva gestione di tali cluster.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni alle risorse create nella console Amazon ECS.

```

{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cloudformation:CreateStack",
 "cloudformation>DeleteStack",
 "cloudformation:DescribeStack*",
 "cloudformation:UpdateStack"
],
 "Resource": [
 "arn:*:cloudformation:*:*:stack/ECS-Console-V2-Service-*"
]
 }
]
}

```

## Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS

Service Auto Scaling è reso possibile da una combinazione di Amazon ECS e Application Auto Scaling. CloudWatch APIs I servizi vengono creati e aggiornati con Amazon ECS, gli allarmi vengono creati e le politiche di scalabilità vengono create con CloudWatch Application Auto Scaling.

Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie le seguenti autorizzazioni per interagire con le impostazioni di Service Auto Scaling, come illustrato nella seguente politica di esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "application-autoscaling:*",
 "ecs:DescribeServices",
 "ecs:UpdateService",
 "cloudwatch:DescribeAlarms",
 "cloudwatch:PutMetricAlarm",
 "cloudwatch>DeleteAlarms",
 "cloudwatch:DescribeAlarmHistory",
 "cloudwatch:DescribeAlarmsForMetric",
 "cloudwatch:GetMetricStatistics",
 "cloudwatch:ListMetrics",
 "cloudwatch:DisableAlarmActions",
 "cloudwatch:EnableAlarmActions",
 "iam:CreateServiceLinkedRole",
 "sns:CreateTopic",
 "sns:Subscribe",
 "sns:Get*",
 "sns:List*"
],
 "Resource": ["*"]
 }
]
}
```

Gli esempi di policy IAM [Crea un esempio di servizio Amazon ECS](#) e [Esempio di aggiornamento del servizio Amazon ECS](#) riportano le autorizzazioni necessarie per utilizzare Dimensionamento automatico nella AWS Management Console.

Il servizio Application Auto Scaling necessita inoltre dell'autorizzazione per descrivere i servizi e gli CloudWatch allarmi Amazon ECS e delle autorizzazioni per modificare il conteggio desiderato del servizio per tuo conto. Le `sns:` autorizzazioni sono per le notifiche CloudWatch inviate a un argomento di Amazon SNS quando viene superata una soglia. Se usi il dimensionamento automatico per i servizi Amazon ECS, viene creato un ruolo collegato ai servizi denominato `AWSServiceRoleForApplicationAutoScaling_ECSService`. Questo ruolo collegato al servizio concede ad Application Auto Scaling l'autorizzazione per descrivere gli allarmi per le policy, per monitorare il conteggio corrente dei processi in esecuzione del servizio e per modificare il conteggio desiderato del servizio. Il ruolo originale di Amazon ECS gestito per Application Auto Scaling era `ecsAutoscaleRole`, ma non è più richiesto. Il ruolo collegato al servizio è il ruolo predefinito per Application Auto Scaling. Per ulteriori informazioni, consulta [Ruoli collegati ai servizi per Application Auto Scaling](#) nella Guida per l'utente di Application Auto Scaling.

Se hai creato il ruolo dell'istanza del contenitore Amazon ECS prima che i CloudWatch parametri fossero disponibili per Amazon ECS, potresti dover aggiungere l'autorizzazione. `ecs:StartTelemetrySession` Per ulteriori informazioni, consulta [Considerazioni](#).

## Concessione dell'autorizzazione all'assegnazione di tag alle risorse al momento della creazione

Le seguenti operazioni API di Amazon ECS per l'assegnazione di tag al momento della creazione consentono di specificare tag quando crei la risorsa. Se i tag sono specificati nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vengano assegnate le autorizzazioni corrette per creare tag.

- `CreateCapacityProvider`
- `CreateCluster`
- `CreateService`
- `CreateTaskSet`
- `RegisterContainerInstance`
- `RegisterTaskDefinition`
- `RunTask`

- StartTask

È possibile utilizzare i tag delle risorse per implementare il controllo basato sugli attributi (ABAC). Per ulteriori informazioni, consultare [the section called “Controllo dell'accesso alle risorse Amazon ECS mediante i tag delle risorse”](#) e [Applicazione di tag alle risorse](#).

Per consentire l'assegnazione di tag al momento della creazione, crea o modifica una policy che includa sia le autorizzazioni per utilizzare l'operazione che crea la risorsa, ad esempio `ecs:CreateCluster` o `ecs:RunTask`, sia l'operazione `ecs:TagResource`.

L'esempio seguente illustra una politica che consente agli utenti di creare cluster e aggiungere tag durante la creazione del cluster. Gli utenti non sono autorizzati ad applicare tag alle risorse esistenti (non possono chiamare l'operazione `ecs:TagResource` direttamente).

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:CreateCluster"
],
 "Resource": "*"
 },
 {
 "Effect": "Allow",
 "Action": [
 "ecs:TagResource"
],
 "Resource": "*",
 "Condition": {
 "StringEquals": {
 "ecs:CreateAction": [
 "CreateCluster",
 "CreateCapacityProvider",
 "CreateService",
 "CreateTaskSet",
 "RegisterContainerInstance",
 "RegisterTaskDefinition",
 "RunTask",
 "StartTask"
]
 }
 }
 }
]
}
```



```
}
 }
]
}
```

L'operazione `ecs:TagResource` viene valutata solo se i tag vengono applicati durante l'operazione di creazione di risorse. Pertanto, un utente con le autorizzazioni per la creazione di una risorsa (presupponendo che non siano presenti condizioni di assegnazione di tag) non necessita delle autorizzazioni per utilizzare l'operazione `ecs:TagResource` se nella richiesta non viene specificato alcun tag. Tuttavia, se l'utente tenta di creare una risorsa con tag, la richiesta ha esito negativo se non dispone delle autorizzazioni per utilizzare l'operazione `ecs:TagResource`.

## Amazon ECS controlla l'accesso a tag specifici

È possibile utilizzare condizioni aggiuntive nell'elemento `Condition` delle policy IAM per controllare le chiavi dei tag e i valori che possono essere applicati alle risorse.

Le seguenti chiavi di condizione possono essere utilizzate con gli esempi nella sezione precedente:

- `aws:RequestTag`: indica che una chiave di tag o una chiave e un valore di tag sono presenti in una richiesta. Anche gli altri tag devono essere specificati nella richiesta.
- Da utilizzare assieme all'operatore di condizione `StringEquals` per applicare una combinazione specifica di chiave e valore di tag, ad esempio per applicare il tag `cost-center=cc123`:

```
"StringEquals": { "aws:RequestTag/cost-center": "cc123" }
```

- Da utilizzare assieme all'operatore di condizione `StringLike` per applicare una chiave di tag specifica nella richiesta, ad esempio per applicare la chiave di tag `purpose`:

```
"StringLike": { "aws:RequestTag/purpose": "*" }
```

- `aws:TagKeys`: applica le chiavi di tag utilizzate nella richiesta.
- Da utilizzare assieme al modificatore `ForAllValues` per applicare chiavi di tag specifiche se vengono fornite nella richiesta (se i tag vengono specificati nella richiesta, solo le chiavi di tag specifiche sono consentite; non sono consentiti altri tag). Ad esempio, la chiave di tag `environment` o `cost-center` è consentita:

```
"ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
```

- Da utilizzare assieme al modificatore `ForAnyValue` per implementare la presenza di almeno una delle chiavi di tag specificate nella richiesta. Ad esempio, nella richiesta deve essere presente almeno una delle chiavi di tag `environment` o `webserver`:

```
"ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }
```

Queste chiavi di condizione possono essere applicate alle operazioni di creazione di risorse che supportano l'assegnazione di tag, nonché alle operazioni `ecs:TagResource`. Per sapere se un'operazione API di Amazon ECS supporta l'assegnazione di tag, consulta [Operazioni, risorse e chiavi di condizione per Amazon ECS](#).

Per obbligare gli utenti a specificare i tag quando creano una risorsa, devi utilizzare la chiave di condizione `aws:RequestTag` o `aws:TagKeys` con il modificatore `ForAnyValue` nell'operazione di creazione delle risorse. L'operazione `ecs:TagResource` non viene valutata se un utente non specifica i tag per l'operazione di creazione delle risorse.

Per le condizioni, la chiave di condizione non fa distinzione tra maiuscole e minuscole, mentre il valore della condizione fa distinzione tra maiuscole e minuscole. Pertanto, per applicare la distinzione tra maiuscole e minuscole per una chiave di tag, utilizza la chiave di condizione `aws:TagKeys`, specificando la chiave di tag come valore nella condizione.

Per ulteriori informazioni sulle condizioni multivalore, consulta [Conditions with multiple context key o value](#) nella IAM User Guide.

## Controllo dell'accesso alle risorse Amazon ECS mediante i tag delle risorse

Quando crei una policy IAM che concede agli utenti l'autorizzazione a utilizzare le risorse Amazon ECS, puoi includere informazioni sui tag nell'elemento `Condition` della policy per controllare l'accesso in base ai tag. Questo è noto come controllo degli accessi basato su attributi (ABAC). Il controllo ABAC fornisce un miglior controllo su quali risorse possono essere modificate, utilizzate o eliminate da un utente. Per ulteriori informazioni, consulta [Che cos'è ABAC per AWS?](#)

Ad esempio, è possibile creare una policy che consenta agli utenti di eliminare un cluster ma che neghi l'operazione se il cluster ha il tag `environment=production`. A tale scopo, è possibile utilizzare la chiave di condizione `aws:ResourceTag` per consentire o negare l'accesso alla risorsa in base ai tag collegati alla risorsa.

```
"StringEquals": { "aws:ResourceTag/environment": "production" }
```

Per sapere se un'operazione API di Amazon ECS supporta il controllo degli accessi utilizzando la chiave di condizione `aws:ResourceTag`, consulta [Operazioni, risorse e chiavi di condizione per Amazon ECS](#). Tieni a mente che le operazioni `Describe` non supportano le autorizzazioni a livello di risorsa, pertanto è necessario specificarle in una dichiarazione separata senza condizioni.

Per esempi di policy IAM, consulta [Esempi di politiche di Amazon ECS](#).

Se consenti o neghi a un utente l'accesso a risorse in base ai tag, devi considerare esplicitamente di negare agli utenti la possibilità di aggiungere o rimuovere tali tag dalle stesse risorse. In caso contrario, un utente può eludere le restrizioni e ottenere l'accesso a una risorsa modificandone i tag.

## Esempi di politiche di Amazon ECS

Puoi utilizzare le policy IAM per concedere agli utenti le autorizzazioni per visualizzare e utilizzare risorse specifiche nella console Amazon ECS. È possibile utilizzare le politiche di esempio nella sezione precedente; tuttavia, sono progettate per le richieste effettuate con AWS CLI o un AWS SDK.

Esempio: consenti agli utenti di eliminare un cluster Amazon ECS in base ai tag

La seguente politica consente agli utenti di eliminare i cluster quando il tag ha unkey/value pair of "Purpose/Testing».

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
 "ecs:DeleteCluster"
],
 "Effect": "Allow",
 "Resource": "arn:aws:ecs:region:account-id:cluster/*",
 "Condition": {
 "StringEquals": {
 "aws:ResourceTag/Purpose": "Testing"
 }
 }
 }
]
}
```

# Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service

Utilizza le informazioni seguenti per eseguire la diagnosi e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon ECS e IAM.

## Argomenti

- [Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon ECS](#)
- [Non sono autorizzato a eseguire iam: PassRole](#)
- [Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon ECS](#)
- [Altre risorse per la risoluzione dei problemi](#)

## Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon ECS

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa `my-example-widget` fittizia ma non dispone di autorizzazioni `ecs:GetWidget` fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
ecs:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa `my-example-widget` utilizzando l'azione `ecs:GetWidget`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Non sono autorizzato a eseguire iam: PassRole

Se si riceve un errore che indica che non si è autorizzati a eseguire l'operazione `iam:PassRole`, è necessario aggiornare le policy per poter passare un ruolo ad Amazon ECS.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in Amazon ECS. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

## Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon ECS

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon ECS supporta queste funzionalità, consulta [Come funziona Amazon Elastic Container Service con IAM](#).
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in Account AWS un altro Account AWS di tua proprietà nella](#) IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta [Fornire l'accesso a soggetti Account AWS di proprietà di terze parti](#) nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta [Fornire l'accesso a utenti autenticati esternamente \(Federazione delle identità\)](#) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

## Altre risorse per la risoluzione dei problemi

Le pagine seguenti forniscono informazioni sui codici di errore:

- [Messaggi di errore delle attività interrotte da Amazon ECS](#)
- [Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS](#)

## Best practice IAM per Amazon ECS

È possibile utilizzare AWS Identity and Access Management (IAM) per gestire e controllare l'accesso ai AWS servizi e alle risorse tramite politiche basate su regole per scopi di autenticazione e autorizzazione. Più specificamente, tramite questo servizio, puoi controllare l'accesso alle tue AWS risorse utilizzando policy applicate a utenti, gruppi o ruoli. Tra questi tre, gli utenti sono account che possono avere accesso alle tue risorse. Inoltre, un ruolo IAM è un insieme di autorizzazioni che possono essere assunte da un'identità autenticata, che non è associata a una particolare identità esterna a IAM. Per ulteriori informazioni, consulta la [panoramica di Amazon ECS sulla gestione degli accessi: autorizzazioni e politiche](#).

### Rispetto della policy di accesso con privilegio minimo

Crea policy mirate a consentire agli utenti di eseguire i processi prescritti. Ad esempio, se uno sviluppatore deve interrompere periodicamente un'attività, crea una policy che consenta solo quella particolare azione. L'esempio seguente consente solo a un utente di interrompere un'attività che appartiene a una particolare `task_family` in un cluster con un nome della risorsa Amazon (ARN) specifico. Il riferimento a un ARN in una condizione è anche un esempio di uso delle autorizzazioni a livello di risorsa. Puoi utilizzare le autorizzazioni a livello di risorsa per specificare la risorsa a cui desideri applicare un'azione.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:StopTask"
],
 "Condition": {
 "ArnEquals": {
 "ecs:cluster": "arn:aws:ecs:region:account_id:cluster/cluster_name"
 }
 }
 }
]
}
```

```
 },
 "Resource": [
 "arn:aws:ecs:region:account_id:task-definition/task_family:*"
]
}
]
```

## Fai in modo che le risorse del cluster fungano da limite amministrativo

Le policy con un ambito troppo ristretto possono causare una proliferazione di ruoli e aumentare il sovraccarico amministrativo. Aniché creare ruoli limitati solo ad attività o servizi particolari, crea ruoli limitati ai cluster e utilizza il cluster come limite amministrativo principale.

## Crea pipeline automatizzate per isolare gli utenti finali dall'API

Puoi limitare le azioni che gli utenti possono utilizzare creando pipeline che impacchettano e implementano in automatico le applicazioni sui cluster Amazon ECS. Ciò delega in modo efficace il processo di creazione, aggiornamento ed eliminazione delle attività alla pipeline. Per ulteriori informazioni, consulta [Tutorial: distribuzione standard di Amazon ECS con CodePipeline](#) nella Guida per l'AWS CodePipeline utente.

## Uso di condizioni di policy per un ulteriore livello di sicurezza

Quando necessiti di un ulteriore livello di sicurezza, aggiungi una condizione alla tua policy. Ciò può essere utile se stai eseguendo un'operazione privilegiata o quando devi limitare l'insieme di azioni che possono essere eseguite su particolari risorse. La policy di esempio seguente richiede l'autorizzazione a più fattori quando si elimina un cluster.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:DeleteCluster"
],
 "Condition": {
 "Bool": {
 "aws:MultiFactorAuthPresent": "true"
 }
 }
 }
],
}
```

```
 "Resource": ["*"]
 }
]
```

I tag applicati a un servizio vengono propagati a tutte le attività che fanno parte di tale servizio. Per questo motivo, puoi creare ruoli mirati alle risorse Amazon ECS con tag specifici. Nella seguente politica, un principale IAM avvia e interrompe tutte le attività con una chiave di tag `Department` e un valore di tag `Accounting`.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:StartTask",
 "ecs:StopTask",
 "ecs:RunTask"
],
 "Resource": "arn:aws:ecs:*",
 "Condition": {
 "StringEquals": {"ecs:ResourceTag/Department": "Accounting"}
 }
 }
]
}
```

## Verifica periodicamente l'accesso a APIs

Un utente potrebbe cambiare ruolo. In seguito al cambio di ruolo, le autorizzazioni che erano state precedentemente concesse all'utente potrebbero non essere più valide. Assicurati di verificare chi ha accesso ad Amazon ECS APIs e se tale accesso è ancora garantito. Valuta l'integrazione di IAM con una soluzione di gestione del ciclo di vita degli utenti che revochi in automatico l'accesso quando un utente lascia l'organizzazione. Per ulteriori informazioni, consulta le [linee guida per i controlli AWS di sicurezza nella Guida](#) per l'AWS Identity and Access Management utente.

## Registrazione e monitoraggio in Amazon Elastic Container Service

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni di Amazon Elastic Container Service e delle tue AWS soluzioni. È necessario raccogliere i dati



di monitoraggio da tutte le parti della AWS soluzione in modo da poter eseguire più facilmente il debug di un errore multipunto, se si verifica. AWS fornisce diversi strumenti per monitorare le risorse Amazon ECS e rispondere a potenziali incidenti:

### CloudWatch Allarmi Amazon

Controllano un singolo parametro per un periodo di tempo specificato ed eseguono una o più operazioni in base alla relazione tra il valore del parametro e una determinata soglia per più periodi di tempo. L'azione è una notifica inviata a un argomento di Amazon Simple Notification Service (Amazon SNS) o a una politica di Amazon Auto EC2 Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta [Monitora Amazon ECS utilizzando CloudWatch](#).

Per i servizi con attività che utilizzano il tipo di avvio Fargate, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le attività del servizio in base a CloudWatch metriche, come l'utilizzo della CPU e della memoria. Per ulteriori informazioni, consulta [Ridimensiona automaticamente il tuo servizio Amazon ECS](#).

Per i cluster con attività o servizi che utilizzano il tipo di EC2 avvio, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le istanze del contenitore in base a CloudWatch metriche, come la prenotazione della memoria del cluster.

### CloudWatch Registri Amazon

Monitora, archivia e accedi ai file di log dei container nei processi di Amazon ECS specificando il driver di log `awslogs` nelle definizioni di attività. Per ulteriori informazioni, consulta [Uso del driver awslogs](#).

Puoi anche monitorare, archiviare e accedere al sistema operativo e ai file di log dell'agente del container Amazon ECS dalle istanze di container Amazon ECS. Questo metodo per accedere ai log può essere utilizzato per i contenitori che utilizzano il tipo di EC2 avvio..

### CloudWatch Eventi Amazon

Abbina gli eventi e li indirizza a una o più funzioni o flussi target per apportare modifiche, acquisire informazioni di stato ed effettuare azioni correttive. Per ulteriori informazioni, [Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge](#) consulta questa guida ed [EventBridge è l'evoluzione di Amazon CloudWatch Events](#) nella Amazon EventBridge User Guide.

## AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon ECS. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon ECS, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta [Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail](#).

## AWS Trusted Advisor

Trusted Advisor attinge alle best practice apprese servendo centinaia di migliaia di AWS clienti. Trusted Advisor ispeziona l'AWS ambiente e quindi formula raccomandazioni quando esistono opportunità per risparmiare denaro, migliorare la disponibilità e le prestazioni del sistema o contribuire a colmare le lacune di sicurezza. Tutti i AWS clienti hanno accesso a cinque Trusted Advisor controlli. I clienti con un piano di supporto Business o Enterprise possono visualizzare tutti i Trusted Advisor controlli.

Per ulteriori informazioni, consulta [AWS Trusted Advisor](#) nella Guida per l'utente di Supporto .

## AWS Compute Optimizer

AWS Compute Optimizer è un servizio che analizza le metriche di configurazione e utilizzo delle risorse. AWS Il servizio segnala se le risorse sono ideali e genera suggerimenti di ottimizzazione per ridurre i costi e migliorare le prestazioni dei carichi di lavoro.

Per ulteriori informazioni, consulta [AWS Compute Optimizer consigli per Amazon ECS](#).

Un'altra parte importante del monitoraggio di Amazon ECS consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Le dashboard della AWS console CloudWatch Trusted Advisor, e altre, forniscono una at-a-glance panoramica dello stato del tuo ambiente. AWS Ti consigliamo inoltre di controllare i file di log sulle tue istanze di container e i container nei tuoi processi.

# Convalida della conformità per Amazon Elastic Container Service

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione [Scope by Compliance Program Servizi AWS](#) e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di [AWS conformità Programmi](#) di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Scaricamento dei report in AWS Artifact](#).

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Governance e conformità per la sicurezza](#): queste guide all'implementazione di soluzioni illustrano considerazioni relative all'architettura e i passaggi per implementare le funzionalità di sicurezza e conformità.
- [Riferimenti sui servizi conformi ai requisiti HIPAA](#): elenca i servizi HIPAA idonei. Non tutti Servizi AWS sono idonei alla normativa HIPAA.
- [AWS Risorse per la conformità](#): questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- [AWS Guide alla conformità dei clienti](#): comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- [Valutazione delle risorse con regole](#) nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- [AWS Security Hub](#)— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).
- [Amazon GuardDuty](#): Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- [AWS Audit Manager](#)— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

## Best practice di conformità e sicurezza per Amazon ECS

La tua responsabilità di conformità quando utilizzi Amazon ECS è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili.

### Payment Card Industry Data Security Standard (PCI DSS)

È importante comprendere l'intero flusso di dati dei titolari di carta (cardholder data, CHD) all'interno dell'ambiente quando si aderisce allo standard PCI DSS. Il flusso CHD determina l'applicabilità del PCI DSS, definisce i confini e i componenti di un ambiente di dati dei titolari di carta (cardholder data environment, CDE) e quindi l'ambito di una valutazione PCI DSS. La determinazione accurata dell'ambito del PCI DSS è fondamentale per definire il livello di sicurezza e, in ultima analisi, per una valutazione efficace. I clienti devono disporre di una procedura per la determinazione dell'ambito che ne garantisca la completezza e rilevi modifiche o deviazioni dall'ambito.

La natura temporanea delle applicazioni containerizzate comporta ulteriori complessità quando si effettua l'audit delle configurazioni. Di conseguenza, i clienti devono mantenere la consapevolezza di tutti i parametri di configurazione di container per garantire che i requisiti di conformità siano soddisfatti in tutte le fasi del ciclo di vita di un container.

Per ulteriori informazioni su come ottenere la conformità allo standard PCI DSS su Amazon ECS, fai riferimento ai seguenti whitepaper.

- [Progettazione su Amazon ECS per la conformità allo standard PCI DSS](#)

### HIPAA (Health Insurance Portability and Accountability Act degli Stati Uniti d'America)

L'uso di Amazon ECS con carichi di lavoro che elaborano informazioni sanitarie protette (protected health information, PHI) non richiede alcuna configurazione aggiuntiva. Amazon ECS funge da servizio di orchestrazione che coordina il lancio di container su Amazon EC2. Non funziona con o su dati all'interno del carico di lavoro orchestrato. In linea con le normative HIPAA e il Business Associate Addendum per AWS, le PHI devono essere crittografate in transito e a riposo quando vi si accede dai container lanciati con Amazon ECS.

Con ogni opzione di AWS storage sono disponibili diversi meccanismi di crittografia a riposo, ad esempio Amazon S3, Amazon EBS e AWS KMS. Puoi implementare una rete overlay (come Weave Net) per garantire la crittografia completa dei PHI trasferiti tra i contenitori VNS3 o per fornire un livello di crittografia ridondante. Dovresti anche utilizzare la registrazione completa e indirizzare tutti

i log dei container ad Amazon. CloudWatch Per informazioni sull'utilizzo delle migliori pratiche per la sicurezza dell'infrastruttura, consulta [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

## AWS Security Hub

Uso AWS Security Hub. Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno AWS. La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina [Documentazione di riferimento sui controlli della Centrale di sicurezza](#).

## Amazon GuardDuty con monitoraggio del runtime di Amazon ECS

Amazon GuardDuty è un servizio di rilevamento delle minacce che aiuta a proteggere account, contenitori, carichi di lavoro e dati all'interno del tuo AWS ambiente. Utilizzando modelli di machine learning (ML) e funzionalità di rilevamento di anomalie e minacce, monitora GuardDuty continuamente diverse fonti di log e attività di runtime per identificare e dare priorità ai potenziali rischi per la sicurezza e alle attività dannose nel tuo ambiente.

Utilizza Runtime Monitoring GuardDuty per identificare comportamenti dannosi o non autorizzati. Il monitoraggio del runtime protegge i carichi di lavoro in esecuzione su Fargate EC2 e AWS monitorando continuamente i log e le attività di rete per identificare comportamenti dannosi o non autorizzati. Runtime Monitoring utilizza un agente di GuardDuty sicurezza leggero e completamente gestito che analizza il comportamento sull'host, come l'accesso ai file, l'esecuzione dei processi e le connessioni di rete. Ciò riguarda questioni quali l'aumento dei privilegi, l'uso di credenziali esposte o la comunicazione con indirizzi IP e domini dannosi e la presenza di malware sulle istanze Amazon EC2 e sui carichi di lavoro dei container. [Per ulteriori informazioni, consulta GuardDuty Runtime Monitoring nella Guida per l'utente. GuardDuty](#)

## Consigli sulla conformità

Ti consigliamo di coinvolgere tempestivamente i responsabili dei programmi di conformità all'interno della tua azienda e di utilizzare il modello di responsabilità AWS condivisa per identificare la titolarità del controllo della conformità e il successo dei programmi di conformità pertinenti. Per ulteriori informazioni, consulta [AWS modello di responsabilità condivisa per Amazon ECS](#).

# AWS Fargate Standard federale per l'elaborazione delle informazioni (FIPS-140)

Federal Information Processing Standard (FIPS). FIPS-140 è uno standard di sicurezza del governo degli Stati Uniti e del Canada che specifica i requisiti di sicurezza previsti per i moduli crittografici che proteggono informazioni sensibili. FIPS-140 definisce una serie di funzioni di crittografia convalidate che si possono utilizzare per la crittografia dei dati in transito e dei dati a riposo.

Quando attivi la conformità a FIPS-140, puoi eseguire carichi di lavoro su Fargate in maniera conforme a FIPS-140. Per ulteriori informazioni sulla conformità FIPS-140, vedere [Federal Information Processing Standard \(FIPS\) 140-3](#).

## AWS Fargate Considerazioni sullo standard FIPS-140

Quando utilizzi la conformità a FIPS-140 su Fargate, tieni in considerazione i fattori seguenti:

- La conformità a FIPS-140 è disponibile solo nelle Regioni AWS GovCloud (US) .
- Fargate supporta la versione FIPS-140 140.3
- La conformità a FIPS-140 è disattivata per impostazione predefinita. Devi attivarla.
- Per la conformità a FIPS-140, le tue attività devono utilizzare la configurazione seguente:
  - `operatingSystemFamily` deve essere LINUX.
  - `cpuArchitecture` deve essere X86\_64.
  - La versione della piattaforma Fargate deve essere 1.4.0 o successiva.

## Uso di FIPS su Fargate

Utilizza la procedura seguente per adottare la conformità a FIPS-140 su Fargate.

1. Attiva la conformità a FIPS-140. Per ulteriori informazioni, consulta [the section called “AWS Fargate Conformità al Federal Information Processing Standard \(FIPS-140\)”](#).
2. Facoltativamente, puoi utilizzare ECS Exec per eseguire il comando seguente allo scopo di verificare lo stato di conformità a FIPS-140 di un cluster.

Sostituisci *cluster-name* con il nome del cluster, *task-id* con l'ID o l'ARN dell'attività e *container-name* con il nome del contenitore dell'attività su cui desideri eseguire il comando.

Un valore restituito corrispondente a "1" indica che stai utilizzando FIPS.

```
aws ecs execute-command \
 --cluster cluster-name \
 --task task-id \
 --container container-name \
 --interactive \
 --command "cat /proc/sys/crypto/fips_enabled"
```

## Uso CloudTrail per il controllo FIPS-140 di Fargate

CloudTrail è attivata nel tuo AWS account quando crei l'account. Quando si verifica un'attività di API e console in Amazon ECS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta [Visualizzazione degli eventi con la cronologia degli CloudTrail eventi](#).

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon ECS, crea un percorso da CloudTrail utilizzare per distribuire i file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta [the section called "Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail"](#).

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'azione dell'PutAccountSettingDefaultAPI:

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAIV5AJI5LXF5EXAMPLE",
 "arn": "arn:aws:iam::123456789012:user/jdoe",
 "accountId": "123456789012",
 "accessKeyId": "AKIAIPWIOFC3EXAMPLE",
 },
 "eventTime": "2023-03-01T21:45:18Z",
 "eventSource": "ecs.amazonaws.com",
```

```
"eventName": "PutAccountSettingDefault",
"awsRegion": "us-gov-east-1",
"sourceIPAddress": "52.94.133.131",
"userAgent": "aws-cli/2.9.8 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
ecs.put-account-setting",
"requestParameters": {
 "name": "fargateFIPSMODE",
 "value": "enabled"
},
"responseElements": {
 "setting": {
 "name": "fargateFIPSMODE",
 "value": "enabled",
 "principalArn": "arn:aws:iam::123456789012:user/jdoe"
 }
},
"requestID": "acdc731e-e506-447c-965d-f5f75EXAMPLE",
"eventID": "6afced68-75cd-4d44-8076-0beEXAMPLE",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "123456789012",
"eventCategory": "Management",
"tlsDetails": {
 "tlsVersion": "TLSv1.2",
 "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
 "clientProvidedHostHeader": "ecs-fips.us-gov-east-1.amazonaws.com"
}
}
```

## Sicurezza dell'infrastruttura in Amazon Elastic Container Service

In quanto servizio gestito, Amazon Elastic Container Service è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta [AWS Cloud Security](#). Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi [Infrastructure Protection](#) in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon ECS attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.



- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare [AWS Security Token Service](#) (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi chiamare queste operazioni API da una qualsiasi posizione di rete. Amazon ECS supporta le policy di accesso basate sulle risorse, che possono includere limitazioni in base all'indirizzo IP di origine, quindi assicurati che le policy contengano l'indirizzo IP della posizione di rete. Puoi anche utilizzare le policy di Amazon ECS per controllare l'accesso da endpoint Amazon Virtual Private Cloud specifici o specifici VPCs. In effetti, questo isola l'accesso alla rete a una determinata risorsa Amazon ECS solo dal VPC specifico all'interno della rete. AWS Per ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia di Amazon ECS \(AWS PrivateLink\)](#).

## Endpoint VPC dell'interfaccia di Amazon ECS (AWS PrivateLink)

Puoi migliorare la posizione di sicurezza del VPC configurando Amazon ECS in modo che utilizzi un endpoint VPC di interfaccia. Gli endpoint di interfaccia sono alimentati da AWS PrivateLink, una tecnologia che consente di accedere in modo privato ad Amazon ECS APIs utilizzando indirizzi IP privati. AWS PrivateLink limita tutto il traffico di rete tra il tuo VPC e Amazon ECS alla rete Amazon. Non è richiesto un gateway Internet, un dispositivo NAT o un gateway privato virtuale.

Per ulteriori informazioni sugli AWS PrivateLink endpoint VPC, consulta la sezione Endpoint VPC [nella Amazon VPC User Guide](#).

## Considerazioni

Le considerazioni relative agli endpoint nelle regioni sono state introdotte a partire dal 23 dicembre 2023

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- È necessario disporre dei seguenti endpoint VPC specifici della regione:

 **Note**

Se non configuri tutti gli endpoint, il traffico passerà agli endpoint pubblici, non all'endpoint VPC.

- `com.amazonaws.region.ecs-agent`
  - `com.amazonaws.region.ecs-telemetry`
  - `com.amazonaws.region.ecs`
- Ad esempio, la regione Canada occidentale (Calgary) (ca-west-1) necessita dei seguenti endpoint VPC:
- `com.amazonaws.ca-west-1.ecs-agent`
  - `com.amazonaws.ca-west-1.ecs-telemetry`
  - `com.amazonaws.ca-west-1.ecs`
- Se utilizzi un modello per creare AWS risorse nella nuova regione e il modello è stato copiato da una regione introdotta prima del 23 dicembre 2023, a seconda della regione di origine, esegui una delle seguenti operazioni.

Ad esempio, la regione del mittente è Stati Uniti orientali (Virginia settentrionale) (us-east-1). La regione da copiare è Canada occidentale (Calgary) (ca-west-1).

| Configurazione                                                       | Azione                                                                                                                  |  |
|----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|--|
| La regione da cui è stata copiata non è presente alcun endpoint VPC. | Crea tutti e tre gli endpoint VPC per la nuova regione (ad esempio,). <code>com.amazonaws.ca-west-1.ecs-agent</code>    |  |
| La regione copiata contiene endpoint VPC specifici della regione.    | a. Crea tutti e tre gli endpoint VPC per la nuova regione (ad esempio,). <code>com.amazonaws.ca-west-1.ecs-agent</code> |  |

| Configurazione | Azione                                                                                                                                         |  |
|----------------|------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                | b. Elimina tutti e tre gli endpoint VPC per la regione di origine della copia (ad esempio,).<br><code>com.amazonaws.us-east-1.ecs-agent</code> |  |

### Considerazioni sugli endpoint VPC di Amazon ECS per il tipo di avvio Fargate

Quando è presente un endpoint VPC per `ecr.dkr` e nello stesso VPC `ecr.api` in cui viene distribuita un'attività Fargate, utilizzerà l'endpoint VPC. Se non è presente un endpoint VPC, utilizzerà l'interfaccia Fargate.

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- Le attività che utilizzano il tipo di avvio Fargate non richiedono l'interfaccia VPC endpoint per Amazon ECS, ma potrebbero essere necessari endpoint VPC di interfaccia per Amazon ECR, Secrets Manager o Amazon Logs descritti nei punti seguenti. CloudWatch
- Per consentire ai processi di estrarre immagini private da Amazon ECR, è necessario creare gli endpoint VPC di interfaccia per Amazon ECR. Per ulteriori informazioni, consulta [Endpoint VPC dell'interfaccia \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon Elastic Container Registry.

#### Important

Se configuri Amazon ECR per utilizzare un endpoint VPC dell'interfaccia, puoi creare un ruolo di esecuzione delle attività che include chiavi di condizione per limitare l'accesso a un VPC o endpoint VPC specifico. Per ulteriori informazioni, consulta [Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia](#).

- Per consentire ai processi di estrarre dati sensibili da Secrets Manager, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta [Utilizzo di Secrets Manager con endpoint VPC](#) nella Guida per l'utente di AWS Secrets Manager.

- Se il tuo VPC non dispone di un gateway Internet e le tue attività utilizzano il driver di `awslogs` registro per inviare le informazioni di registro ai CloudWatch registri, devi creare un endpoint VPC di interfaccia per i registri. CloudWatch Per ulteriori informazioni, consulta [Using CloudWatch Logs with Interface VPC Endpoints](#) nella CloudWatch Amazon Logs User Guide.
- Gli endpoint VPC attualmente non supportano le richieste inter-Regionali. Assicurati di creare l'endpoint nella stessa regione in cui prevedi di inviare le chiamate API ad Amazon ECS. Supponiamo ad esempio di voler eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale). Dovrai ovviamente creare l'endpoint VPC di Amazon ECS nella Regione Stati Uniti orientali (Virginia settentrionale). Un endpoint VPC di Amazon ECS creato in qualsiasi altra Regione non può eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale).
- Gli endpoint VPC supportano solo il DNS fornito da Amazon tramite Amazon Route 53. Se si desidera utilizzare il proprio DNS, è possibile usare l'inoltro condizionale sul DNS. Per ulteriori informazioni, consulta [Set opzioni DHCP](#) nella Guida per l'utente di Amazon VPC.
- Il gruppo di sicurezza collegato all'endpoint VPC deve consentire le connessioni in entrata sulla porta TCP 443 dalla sottorete privata del VPC.
- La gestione Service Connect del proxy Envoy utilizza l'endpoint VPC `com.amazonaws.region.ecs-agent`. Quando non utilizzi gli endpoint VPC, la gestione Service Connect del proxy Envoy utilizza l'endpoint `ecs-sc` in quella determinata Regione. Per un elenco degli endpoint Amazon ECS in ciascuna Regione, consulta [Endpoint e quote Amazon ECS](#).

## Considerazioni sugli endpoint VPC di Amazon ECS per il tipo di lancio EC2

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- Le attività che utilizzano il tipo di EC2 avvio richiedono che le istanze del contenitore su cui vengono avviate eseguano una versione 1.25.1 o successiva dell'agente container Amazon ECS. Per ulteriori informazioni, consulta [Gestione delle istanze di container Amazon ECS Linux](#).
- Per consentire ai processi di estrarre dati sensibili da Secrets Manager, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta [Utilizzo di Secrets Manager con endpoint VPC](#) nella Guida per l'utente di AWS Secrets Manager .
- Se il tuo VPC non dispone di un gateway Internet e le tue attività utilizzano il driver di `awslogs` registro per inviare le informazioni di registro ai CloudWatch registri, devi creare un endpoint VPC di interfaccia per i registri. CloudWatch Per ulteriori informazioni, consulta [Using CloudWatch Logs with Interface VPC Endpoints](#) nella CloudWatch Amazon Logs User Guide.

- Gli endpoint VPC attualmente non supportano le richieste inter-Regionali. Assicurati di creare l'endpoint nella stessa regione in cui prevedi di inviare le chiamate API ad Amazon ECS. Supponiamo ad esempio di voler eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale). Dovrai ovviamente creare l'endpoint VPC di Amazon ECS nella Regione Stati Uniti orientali (Virginia settentrionale). Un endpoint VPC Amazon ECS creato in qualsiasi altra regione non può eseguire attività negli Stati Uniti orientali (Virginia settentrionale).
- Gli endpoint VPC supportano solo il DNS fornito da Amazon tramite Amazon Route 53. Se si desidera utilizzare il proprio DNS, è possibile usare l'inoltro condizionale sul DNS. Per ulteriori informazioni, consulta [Set opzioni DHCP](#) nella Guida per l'utente di Amazon VPC.
- Il gruppo di sicurezza collegato all'endpoint VPC deve consentire le connessioni in entrata sulla porta TCP 443 dalla sottorete privata del VPC.

## Creazione di endpoint VPC per Amazon ECS

Per creare l'endpoint VPC per il servizio Amazon ECS, utilizza la [procedura Access an AWS service using an interface VPC endpoint nella Amazon VPC](#) User Guide per creare i seguenti endpoint. Se sono presenti istanze di container all'interno del VPC, è necessario creare gli endpoint nell'ordine in cui sono elencati. Se intendi creare le istanze di container dopo la creazione dell'endpoint VPC, l'ordine non ha importanza.

### Note

Se non configuri tutti gli endpoint, il traffico passerà agli endpoint pubblici, non all'endpoint VPC.

Quando crei endpoint, Amazon ECS crea anche un nome DNS privato per l'endpoint. Ad esempio, per ecs-agent e `ecs-a.region.amazonaws.com` per ecs-telemetry. `ecs-t.region.amazonaws.com`

- `com.amazonaws.region.ecs-agent`
- `com.amazonaws.region.ecs-telemetry`
- `com.amazonaws.region.ecs`

 Note

*region* rappresenta l'identificatore della regione per una AWS regione supportata da Amazon ECS, ad esempio us-east-2 per la regione Stati Uniti orientali (Ohio).

L'ecs-agent endpoint utilizza l'ecs:pollAPI e l'ecs-telemetry endpoint utilizza l'API and. ecs:poll ecs:StartTelemetrySession

Se hai attività esistenti che utilizzano il tipo di EC2 avvio, dopo aver creato gli endpoint VPC, ogni istanza del contenitore deve acquisire la nuova configurazione. Perché ciò accada, è necessario riavviare ogni istanza di container o riavviare l'agente del container Amazon ECS in ogni istanza di container. Per riavviare l'agente container, effettua le seguenti operazioni.

Come riavviare l'agente del container di Amazon ECS

1. Accedi alla tua istanza di container con SSH.
2. Arresta l'agente del container di .

```
sudo docker stop ecs-agent
```

3. Avvia l'agente container.

```
sudo docker start ecs-agent
```

Dopo aver creato gli endpoint VPC e riavviato l'agente del container di Amazon ECS in ogni istanza di container, tutte le attività appena avviate ottengono la nuova configurazione.

## Creazione di una policy per l'endpoint VPC per Amazon ECS

Puoi allegare una policy di endpoint all'endpoint VPC che controlla l'accesso ad Amazon ECS. La policy specifica le informazioni riportate di seguito:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite.
- Le risorse sui cui si possono eseguire operazioni.

Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) in Guida per l'utente di Amazon VPC.

Esempio: policy di endpoint VPC per le operazioni Amazon ECS

Di seguito è riportato un esempio di una policy di endpoint per Amazon ECS. Se collegata a un endpoint, questa policy concede l'accesso all'autorizzazione per creare ed elencare i cluster. Le operazioni `CreateCluster` e `ListClusters` non accettano risorse, pertanto la definizione delle risorse è impostata su `*` per tutte le risorse.

```
{
 "Statement": [
 {
 "Principal": "*",
 "Effect": "Allow",
 "Action": [
 "ecs:CreateCluster",
 "ecs:ListClusters"
],
 "Resource": [
 "*"
]
 }
]
}
```

## AWS modello di responsabilità condivisa per Amazon ECS

La sicurezza e la conformità sono una responsabilità condivisa tra AWS e il cliente. Questo modello condiviso può contribuire ad alleggerire l'onere operativo del cliente in quanto AWS opera, gestisce e controlla i componenti, dal sistema operativo host e dal livello di virtualizzazione fino alla sicurezza fisica delle strutture in cui opera il servizio. Il cliente si assume la responsabilità e la gestione del sistema operativo guest (inclusi gli aggiornamenti e le patch di sicurezza), degli altri software applicativi associati e della configurazione del firewall del gruppo di sicurezza AWS fornito. I clienti devono valutare con attenzione i servizi scelti, dato che le loro responsabilità variano in base a servizi utilizzati, integrazione di tali servizi nel loro ambiente IT, leggi e regolamenti applicabili. La natura di questa responsabilità condivisa fornisce inoltre la flessibilità e il controllo dei clienti che consentono l'implementazione.

## Tipo di avvio di Fargate

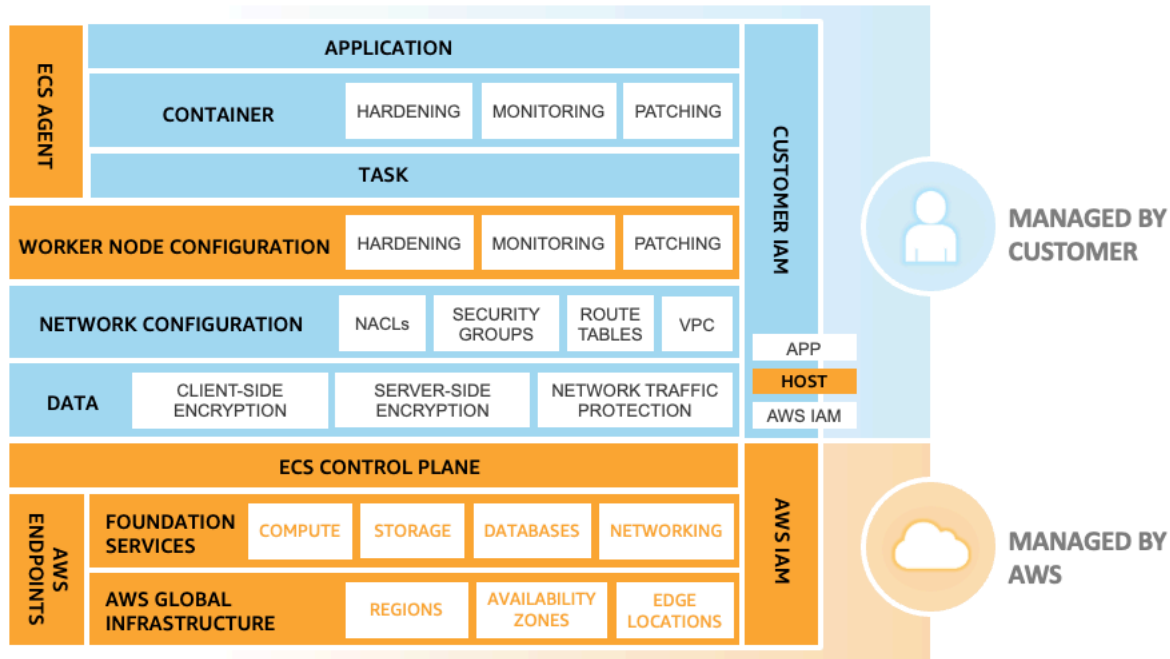
La figura seguente mostra il modello di responsabilità condivisa per il tipo di lancio di Fargate. Fargate esegue ogni carico di lavoro in un ambiente hardware virtualizzato isolato. Di conseguenza, ogni attività ottiene una capacità di infrastruttura dedicata. I carichi di lavoro containerizzati in esecuzione su Fargate non condividono un sistema operativo, un kernel Linux, un'interfaccia di rete, uno storage temporaneo, una CPU o una memoria con altre attività. Quando utilizzano Fargate, i clienti non sono responsabili della protezione dell'infrastruttura di elaborazione che gestisce i propri container. Fargate provvederà al provisioning e alle patch dell'infrastruttura su cui vengono eseguiti i carichi di lavoro dei clienti. Per ulteriori informazioni, consulta [Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS](#).

L'utente è responsabile della gestione delle seguenti risorse:

- Configurazione di rete tra cui VPC NACLs, gruppi di sicurezza e tabelle di routing
- Crittografia dello storage di client e servizi. Per ulteriori informazioni, consulta [Opzioni di storage per le attività di Amazon ECS](#).
- Immagini dei contenitori. Per ulteriori informazioni, consulta [Best practice per la sicurezza di attività e container di Amazon ECS](#).
- Autorizzazioni IAM per le applicazioni utilizzando il ruolo task. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).



## AWS Shared Responsibility Model for Amazon ECS with Fargate

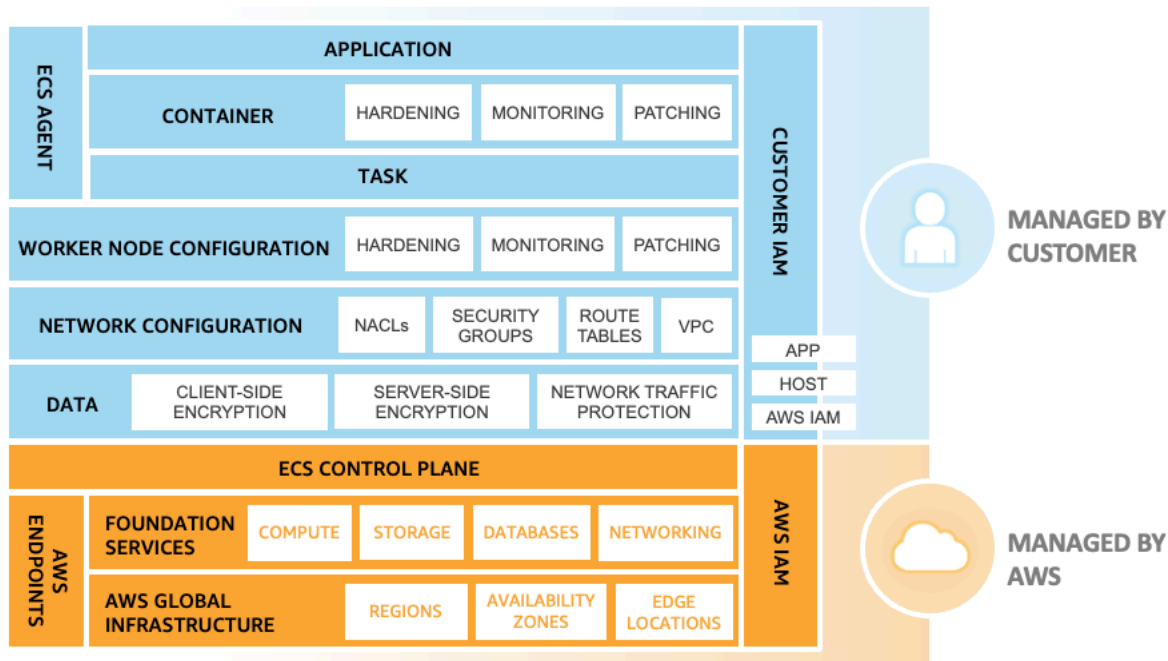


## EC2 tipo di lancio

La figura seguente mostra la responsabilità condivisa per il tipo di EC2 lancio. Quando esegui attività su EC2 istanze, sei responsabile della manutenzione delle EC2 istanze in aggiunta alle seguenti risorse:

- L'agente Amazon ECS.
- L'AMI dell' EC2 istanza, inclusi patch e rafforzamento.
- Configurazione di rete che include VPC NACLs, gruppi di sicurezza e tabelle di routing.
- Crittografia dello storage di client e servizi. Per ulteriori informazioni, consulta [Opzioni di storage per le attività di Amazon ECS](#).
- Immagini dei contenitori. Per ulteriori informazioni, consulta [Best practice per la sicurezza di attività e container di Amazon ECS](#).
- Autorizzazioni IAM per le applicazioni utilizzando il ruolo task. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

## AWS Shared Responsibility Model for Amazon ECS



## Best practice di sicurezza di rete per Amazon ECS

La sicurezza di rete è un argomento ampio che comprende diversi argomenti secondari. Queste includono segmentazione e isolamento della rete encryption-in-transit, firewall, routing del traffico e osservabilità.

### Crittografia in transito

La crittografia del traffico di rete impedisce agli utenti non autorizzati di intercettare e leggere i dati trasmessi attraverso una rete. Con Amazon ECS, la crittografia di rete può essere implementata in uno qualsiasi dei modi descritti di seguito.

- Uso delle istanze Nitro:

Per impostazione predefinita, il traffico viene crittografato in automatico tra i seguenti tipi di istanza Nitro: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn e R5n. Il traffico non viene crittografato se instradato attraverso un gateway di transito, un sistema di bilanciamento del carico o un intermediario simile.

- [Crittografia in transito](#)
- [Annuncio di novità del 2019](#)
- [Questo intervento di RE:InForce 2019](#)

- Uso dell'estensione Server Name Indication (SNI) con Application Load Balancer:

Application Load Balancer (ALB) e Network Load Balancer (NLB) supportano l'estensione Server Name Indication (SNI). Utilizzando SNI, puoi posizionare più applicazioni sicure dietro un unico ascoltatore. A tale scopo, ciascuna dispone di un suo certificato TLS. Consigliamo di effettuare il provisioning dei certificati per il sistema di bilanciamento del carico utilizzando AWS Certificate Manager (ACM) e di aggiungerli all'elenco di certificati dell'ascoltatore. Il AWS load balancer utilizza un algoritmo di selezione intelligente dei certificati con SNI. Se l'hostname fornito da un client corrisponde a un singolo certificato nell'elenco di certificati, il sistema di bilanciamento del carico seleziona tale certificato. Se un hostname fornito da un client corrisponde a più certificati nell'elenco di certificati, il sistema di bilanciamento del carico seleziona un certificato che il client è in grado di supportare. Alcuni esempi sono un certificato autofirmato o un certificato generato attraverso ACM.

- [SNI con Application Load Balancer](#)
- [SNI con Network Load Balancer](#)
- End-to-end crittografia con certificati TLS:

Questa opzione comporta l'implementazione di un certificato TLS insieme all'attività. Può trattarsi o di un certificato autofirmato o di un certificato rilasciato da un'autorità di certificazione attendibile. Puoi ottenere il certificato facendo riferimento a un segreto per il certificato. Altrimenti, puoi scegliere di eseguire un contenitore che emette una richiesta di firma del certificato (CSR) ad ACM e poi monta il segreto risultante su un volume condiviso.

- [Mantenimento di Transport Layer Security \(TLS\) fino ai container attraverso l'uso di Network Load Balancer con Amazon ECS parte 1](#)
- [Mantenimento del Transport Layer Security \(TLS\) fino al container. Parte 2: Utilizzo AWS Private Certificate Authority](#)

## Reti di attività

I seguenti suggerimenti prendono in considerazione il funzionamento di Amazon ECS. Amazon ECS non utilizza una rete di sovrapposizione. Le attività sono invece configurate per funzionare in diverse modalità di rete. Ad esempio, le attività configurate per l'uso della modalità `bridge` acquisiscono un indirizzo IP non instradabile da una rete Docker in esecuzione su ciascun host. Le attività configurate per l'uso della modalità di rete `awsvpc` acquisiscono un indirizzo IP dalla sottorete dell'host. Le attività configurate con reti `host` utilizzano l'interfaccia di rete dell'host. `awsvpc` è la modalità di rete

preferita. Questo perché è l'unica modalità che puoi utilizzare per assegnare gruppi di sicurezza alle attività. È anche l'unica modalità disponibile per AWS Fargate le attività su Amazon ECS.

## Gruppi di sicurezza per le attività

Consigliamo di configurare le attività per l'uso della modalità di rete `awsvpc`. Dopo aver configurato l'attività per l'uso di tale modalità, l'agente Amazon ECS effettua in automatico il provisioning e il collegamento di un'interfaccia di rete elastica (ENI) all'attività. Quando viene fornito l'ENI, l'attività viene registrata in un AWS gruppo di sicurezza. Il gruppo di sicurezza agisce da firewall virtuale che puoi utilizzare per il controllo del traffico in entrata e in uscita.

Se utilizzi un firewall personalizzato con attività o servizi, aggiungi una regola in uscita per consentire il traffico per gli endpoint di gestione degli agenti Amazon ECS (`ecs-a-*.region.amazonaws.com`), gli endpoint di telemetria (`ecs-t-*.region.amazonaws.com`) e gli endpoint di gestione di Service Connect Envoy (`ecs-sc.region.amazonaws.com`).

## AWS PrivateLink e Amazon ECS

AWS PrivateLink è una tecnologia di rete che consente di creare endpoint privati per diversi AWS servizi, tra cui Amazon ECS. Gli endpoint sono necessari in ambienti di sperimentazione (sandbox) in cui non è presente alcun gateway Internet (IGW) collegato ad Amazon VPC e non sono presenti instradamenti alternativi verso Internet. L'utilizzo AWS PrivateLink garantisce che le chiamate al servizio Amazon ECS rimangano all'interno di Amazon VPC e non attraversino Internet. Per istruzioni su come creare AWS PrivateLink endpoint per Amazon ECS e altri servizi correlati, consulta [Endpoint Amazon VPC con interfaccia Amazon ECS](#).

### Important

AWS Fargate le attività non richiedono un AWS PrivateLink endpoint per Amazon ECS.

Amazon ECR e Amazon ECS supportano entrambi le policy di endpoint. Queste politiche consentono di affinare l'accesso a un servizio. APIs Ad esempio, puoi creare una policy di endpoint per Amazon ECR che consenta solo di inviare immagini a registri in determinati account AWS. Una policy come questa potrebbe essere utilizzata per impedire che i dati vengano esfiltrati attraverso le immagini di container, pur consentendo agli utenti di effettuare l'invio ai registri Amazon ECR autorizzati. Per ulteriori informazioni, consulta [Uso delle policy di endpoint VPC](#).

La seguente politica consente a tutti AWS i principali del tuo account di eseguire tutte le azioni solo sui tuoi repository Amazon ECR:

```
{
 "Statement": [
 {
 "Sid": "LimitECRAccess",
 "Principal": "*",
 "Action": "*",
 "Effect": "Allow",
 "Resource": "arn:aws:ecr:region:account_id:repository/*"
 },
],
}
```

Puoi migliorare ulteriormente quest'opzione impostando una condizione che utilizzi la nuova proprietà `PrincipalOrgID`. Ciò impedisce l'invio e l'estrazione di immagini da parte di un principale IAM che non fa parte del tuo. AWS Organizations Per ulteriori informazioni, consulta [aws: PrincipalOrg ID](#).

Consigliamo di applicare la medesima policy sia agli endpoint com.amazonaws.*region*.ecr.dkr che agli endpoint com.amazonaws.*region*.ecr.api.

## Impostazioni dell'agente Container

Il file di configurazione dell'agente container Amazon ECS include diverse variabili di ambiente relative alla sicurezza della rete. `ECS_AWSVPC_BLOCK_IMDSe` `ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST` vengono utilizzati per bloccare l'accesso di un'attività ai EC2 metadati di Amazon. `HTTP_PROXY` viene utilizzato per configurare l'agente in modo che effettui il routing attraverso un proxy HTTP per la connessione a Internet. Per ricevere istruzioni sulla configurazione dell'agente e del runtime Docker per l'instradamento attraverso un proxy, consulta [Configurazione del proxy HTTP](#).

### Important

Queste impostazioni non sono disponibili quando utilizzi AWS Fargate.

## Consigli sulla sicurezza della rete

Consigliamo di completare la procedura seguente quando configuri Amazon VPC, i sistemi di bilanciamento del carico e la rete.

### Usa la crittografia di rete, ove applicabile, con Amazon ECS

È consigliabile utilizzare la crittografia di rete laddove applicabile. Alcuni programmi di conformità, come PCI DSS, richiedono la crittografia dei dati in transito se i dati contengono dati del titolare della carta. Se il tuo carico di lavoro presenta requisiti simili, configura la crittografia di rete.

I browser moderni avvisano gli utenti quando si connettono a siti non sicuri. Se il servizio è gestito da un sistema di bilanciamento del carico rivolto al pubblico, utilizza TLS/SSL per crittografare il traffico dal browser del client al sistema di bilanciamento del carico e, se necessario, crittografarlo nuovamente nel backend.

### Usa la modalità di **awsvpc** rete e i gruppi di sicurezza per controllare il traffico tra attività e altre risorse in Amazon ECS

È consigliabile utilizzare la modalità di rete `awsvpc` e i gruppi di sicurezza quando devi controllare il traffico tra le attività e tra le attività e altre risorse di rete. Se il tuo servizio è protetto da un ALB, utilizza i gruppi di sicurezza per consentire solo il traffico in entrata da altre risorse di rete che utilizzano lo stesso gruppo di sicurezza del tuo ALB. Se la tua applicazione è protetta da un NLB, configura il gruppo di sicurezza dell'attività per consentire solo il traffico in entrata dall'intervallo CIDR di Amazon VPC e gli indirizzi IP statici assegnati al NLB.

È consigliabile utilizzare i gruppi di sicurezza anche per controllare il traffico tra le attività e altre risorse all'interno di Amazon VPC, come i database Amazon RDS.

### Crea cluster Amazon ECS in Amazon separati VPCs quando il traffico di rete deve essere strettamente isolato

È necessario creare cluster in Amazon separati VPCs quando il traffico di rete deve essere strettamente isolato. Evita di eseguire carichi di lavoro con requisiti di sicurezza rigorosi su cluster con carichi di lavoro che non devono rispettare tali requisiti. Quando è obbligatorio un rigoroso isolamento della rete, crea cluster in Amazon separati VPCs ed esponi selettivamente i servizi ad altri Amazon utilizzando gli endpoint VPCs Amazon VPC. Per ulteriori informazioni, consultare [Endpoint VPC](#).

## Configura AWS PrivateLink gli endpoint quando richiesto per Amazon ECS

È necessario configurare AWS PrivateLink gli endpoint quando necessario. Se la tua politica di sicurezza ti impedisce di collegare un Internet Gateway (IGW) ad Amazon VPCs, configura gli AWS PrivateLink endpoint per Amazon ECS e altri servizi come Amazon ECR e Amazon. AWS Secrets Manager CloudWatch

## Usa Amazon VPC Flow Logs per analizzare il traffico da e verso le attività a esecuzione prolungata in Amazon ECS

È consigliabile utilizzare Amazon VPC Flow Logs per analizzare il traffico da e verso le attività a esecuzione prolungata. Le attività che utilizzano la modalità di rete awsvpc ottengono la propria ENI. In tal modo, puoi monitorare il traffico da e verso le singole attività utilizzando Amazon VPC Flow Logs. Un recente aggiornamento di Amazon VPC Flow Logs (v3) arricchisce i log con metadati sul traffico tra cui l'ID di vpc, l'ID di sottorete e l'ID dell'istanza. Questi metadati possono essere utilizzati per restringere un'indagine. Per ulteriori informazioni, consulta [Amazon VPC Flow Logs](#).

### Note

A causa della natura temporanea dei container, i log di flusso potrebbero non essere sempre un modo efficace per analizzare i pattern di traffico tra diversi container o tra container e altre risorse di rete.

## Best practice per la sicurezza di attività e container di Amazon ECS

L'immagine di container deve essere considerata la prima linea di difesa contro un attacco.

Un'immagine non sicura e mal costruita può consentire a un utente malintenzionato di sfuggire ai limiti del container e accedere all'host. Per ridurre questo rischio, effettua le seguenti operazioni.

Quando configuri le attività e i container, tieni in considerazione i suggerimenti elencati di seguito.

### Creazione di immagini di base o uso di immagini distroless

Inizia rimuovendo tutti i file binari estranei dall'immagine di container. Se stai utilizzando un'immagine sconosciuta tratta da Amazon ECR Public Gallery, ispezionala per fare riferimento al contenuto di ciascuno dei livelli del container. A tale scopo, puoi utilizzare un'applicazione come [Dive](#).

In alternativa, puoi utilizzare immagini distroless che includono solo l'applicazione e le relative dipendenze di runtime. Non contengono gestori di pacchetti o shell (interprete di comandi). Le immagini distroless migliorano il "rapporto segnale/rumore degli scanner e riducono l'onere di stabilire la provenienza in base alle proprie esigenze". Per ulteriori informazioni, consulta la GitHub documentazione su [distroless](#).

Docker dispone di un meccanismo per creare immagini a partire da un'immagine di base e riservata, nota come scratch. Per ulteriori informazioni, consulta [Creazione di una semplice immagine genitore con scratch](#) nella documentazione di Docker. Con linguaggi come Go, puoi creare un file binario collegato statico e farvi riferimento nel Dockerfile. L'esempio seguente mostra come si può ottenere questo risultato.

```
#####
STEP 1 build executable binary
#####
FROM golang:alpine AS builder
Install git.
Git is required for fetching the dependencies.
RUN apk update && apk add --no-cache git
WORKDIR $GOPATH/src/mypackage/myapp/
COPY . .
Fetch dependencies.
Using go get.
RUN go get -d -v
Build the binary.
RUN go build -o /go/bin/hello
#####
STEP 2 build a small image
#####
FROM scratch
Copy our static executable.
COPY --from=builder /go/bin/hello /go/bin/hello
Run the hello binary.
ENTRYPOINT ["/go/bin/hello"]
This creates a container image that consists of your application and nothing else,
making it extremely secure.
```

L'esempio precedente illustra inoltre una build a più fasi. Questi tipi di build sono interessanti dal punto di vista della sicurezza, perché puoi utilizzarli per ridurre al minimo le dimensioni dell'immagine finale inviata al registro di container. Le immagini di container prive di strumenti di compilazione e altri file binari estranei migliorano il livello di sicurezza, riducendo la superficie di attacco dell'immagine.



[Per ulteriori informazioni sulle build in più fasi, consulta la sezione Compilazioni in più fasi nel Docker documentazione.](#)

## Scansione delle immagini per individuare eventuali vulnerabilità

Analogamente alle loro controparti per macchine virtuali, le immagini di container possono contenere file binari e librerie di applicazioni con vulnerabilità oppure sviluppare vulnerabilità nel corso del tempo. Il modo migliore per proteggersi dagli attacchi è quello di analizzare regolarmente le immagini con uno scanner di immagini.

Amazon ECR offre due versioni di scansione di base che utilizzano il database Common Vulnerabilities and Exposures (): CVEs

- AWS scansione nativa di base: utilizza la tecnologia AWS nativa, che ora è GA e consigliata. Questa scansione di base migliorata è progettata per fornire ai clienti risultati di scansione migliori e il rilevamento delle vulnerabilità su un'ampia gamma di sistemi operativi più diffusi. Ciò consente ai clienti di rafforzare ulteriormente la sicurezza delle immagini dei container. Per impostazione predefinita, tutti i nuovi registri clienti vengono attivati in questa versione migliorata.
- Scansione di base di Clair: la precedente versione di scansione di base che utilizza il progetto Clair open source ed è obsoleta. [Per ulteriori informazioni su Clair, vedere Clair on.](#) GitHub

Sia la scansione AWS nativa che quella di base Clair sono supportate in tutte le regioni elencate in [AWS Servizi per regione](#), ad eccezione di quelle aggiunte dopo settembre 2024. Poiché il supporto Clair è obsoleto, Clair non sarà supportato in nuove regioni man mano che verranno aggiunte e non sarà più supportato in tutte le regioni a partire dal 1° ottobre 2025.

Amazon ECR utilizza la severità di un CVE proveniente dalla fonte di distribuzione upstream, se disponibile. Altrimenti, viene utilizzato il punteggio Common Vulnerability Scoring System (CVSS). Il punteggio CVSS può essere utilizzato per ottenere il livello di gravità della vulnerabilità NVD. Per ulteriori informazioni, consulta [NVD Vulnerability Severity Ratings](#).

Puoi anche visualizzare i risultati di una scansione dalla console Amazon ECR o chiamando l'[DescribeImageScanFindings](#) API. Le immagini con una vulnerabilità HIGH o CRITICAL devono essere eliminate o ricreate. Un'immagine implementata che sviluppa una vulnerabilità deve essere sostituita il prima possibile.

[Docker Desktop Edge versione 2.3.6.0](#) o successiva può effettuare la [scansione](#) delle immagini locali. Le scansioni sono gestite da [Snyk](#), un servizio di sicurezza per applicazioni. Quando vengono individuate vulnerabilità, Snyk identifica i livelli e le dipendenze con la vulnerabilità nel Dockerfile.

Suggerisce inoltre alternative sicure, come l'uso di un'immagine di base più snella con meno vulnerabilità o l'aggiornamento di un particolare pacchetto a una versione più recente. Utilizzando la scansione di Docker, gli sviluppatori possono risolvere potenziali problemi di sicurezza prima di inviare le immagini al registro.

- L'articolo [Automatizzazione della conformità delle immagini con Amazon ECR e AWS Security Hub](#) spiega come far emergere le informazioni sulle vulnerabilità da Amazon ECR in AWS Security Hub e automatizzare la correzione bloccando l'accesso alle immagini vulnerabili.

## Rimozione delle autorizzazioni speciali dalle immagini

I flag dei diritti di accesso `setuid` e `setgid` consentono l'esecuzione di un file eseguibile con le autorizzazioni del proprietario o del gruppo del file eseguibile. Rimuovi tutti i file binari con tali diritti di accesso dall'immagine, in quanto possono essere utilizzati per eseguire l'escalation dei privilegi. Valuta la possibilità di rimuovere tutti gli shell (interpreti di comandi) e le utilità come `nc` e `curl` che possono essere utilizzati per scopi nocivi. Puoi trovare i file con diritti di accesso `setuid` e `setgid` utilizzando il comando seguente.

```
find / -perm /6000 -type f -exec ls -ld {} \;
```

Per rimuovere le autorizzazioni speciali da tali file, aggiungi la direttiva seguente all'immagine di container.

```
RUN find / -xdev -perm /6000 -type f -exec chmod a-s {} \; || true
```

## Creazione di un set di immagini curate

Invece di consentire agli sviluppatori di creare le proprie immagini, crea un set di immagini controllate per i diversi stack di applicazioni della tua organizzazione. In tal modo, gli sviluppatori possono fare a meno di imparare a comporre Dockerfile e concentrarsi piuttosto sulla scrittura di codice. Man mano che le modifiche vengono incorporate nella base di codice, una pipeline CI/CD può compilare in automatico l'asset e quindi archivarlo in un repository di artefatti. Infine, copia l'artefatto nell'immagine appropriata prima di inviarlo a un registro Docker come Amazon ECR. Come minimo, dovresti creare un set di immagini di base da cui gli sviluppatori possano creare i propri Dockerfile. È sconsigliabile estrarre immagini da Docker Hub. Non sempre sai che cosa c'è nell'immagine e circa

un quinto delle 1.000 immagini principali presenta vulnerabilità. Un elenco di tali immagini e delle relative vulnerabilità è disponibile all'indirizzo <https://vulnerablecontainers.org/>.

## Scansione dei pacchetti e delle librerie di applicazioni per individuare eventuali vulnerabilità

L'uso di librerie open source è ormai comune. Come per i sistemi operativi e i pacchetti OS, le librerie possono presentare vulnerabilità. Nell'ambito del ciclo di vita dello sviluppo, è necessario analizzare e aggiornare queste librerie quando vengono rilevate vulnerabilità critiche.

Docker Desktop esegue scansioni locali utilizzando Snyk. Può anche essere utilizzato per trovare vulnerabilità e potenziali problemi di licenza nelle librerie open source. Grazie alla possibilità di integrarlo direttamente nei flussi di lavoro degli sviluppatori, permette di mitigare i rischi posti dalle librerie open source. Per ulteriori informazioni, consulta i seguenti argomenti:

- [Strumenti di sicurezza delle applicazioni open source](#) include un elenco di strumenti per rilevare le vulnerabilità nelle applicazioni.

## Esecuzione dell'analisi statica del codice

È consigliabile eseguire l'analisi statica del codice prima di creare un'immagine di container. Viene eseguita sul codice di origine e consente di identificare errori di codifica e codice che potrebbero essere sfruttati da un malintenzionato, come le iniezioni di errori. Puoi usare Amazon Inspector. Per ulteriori informazioni, consulta [Scansione delle immagini dei container Amazon ECR con Amazon Inspector nella Amazon](#) Inspector User Guide.

## Esecuzione di container come utente non root

È consigliabile eseguire i container come utente non root. Per impostazione predefinita, i container vengono eseguiti come utente `root` a meno che nel Dockerfile sia inclusa la direttiva `USER`. Le funzionalità Linux predefinite assegnate da Docker limitano le operazioni che possono essere eseguite come `root`, ma solo in misura marginale. Ad esempio, un container in esecuzione come `root` non può comunque accedere ai dispositivi.

Nell'ambito della pipeline CI/CD, è consigliabile eseguire il comando `lint` sui Dockerfile per cercare la direttiva `USER` e interrompere la build qualora mancasse. Per ulteriori informazioni, consulta i seguenti argomenti:

- [DockerFile-LINT](#) è uno strumento open source RedHat che può essere utilizzato per verificare se il file è conforme alle migliori pratiche.
- [Hadolint](#) è un altro strumento per creare immagini Docker conformi alle best practice.

## Uso di un file system root di sola lettura

È consigliabile utilizzare un file system root di sola lettura. Il file system root di un container è scrivibile per impostazione predefinita. Quando configuri un container con un file system root RO (di sola lettura), ti viene imposto di definire in modo esplicito dove possono essere conservati i dati. Ciò riduce la superficie di attacco, perché non è possibile scrivere sul file system del container a meno che non vengano concesse autorizzazioni specifiche.

### Note

Disporre di un file system root di sola lettura può causare problemi con alcuni pacchetti OS che si aspettano la possibilità di scrivere sul file system. Se hai intenzione di utilizzare file system root di sola lettura, esegui prima un test approfondito.

## Configura attività con limiti di CPU e memoria (Amazon EC2)

È consigliabile configurare attività con limiti di CPU e memoria per ridurre al minimo i seguenti rischi. I limiti di risorse di un'attività stabiliscono un limite massimo per la quantità di CPU e memoria che può essere riservata da tutti i container all'interno di un'attività. Se non vengono impostati limiti, le attività hanno accesso alla CPU e alla memoria dell'host. Ciò può causare problemi per cui le attività implementate su un host condiviso possono privare altre attività delle risorse di sistema.

### Note

Amazon ECS on AWS Fargate Tasks richiede di specificare i limiti di CPU e memoria perché utilizza questi valori per scopi di fatturazione. Un'attività che occupa tutte le risorse di sistema non rappresenta un problema per Amazon ECS Fargate, perché ogni attività viene eseguita su una propria istanza dedicata. Se non specifichi un limite di memoria, Amazon ECS alloca un minimo di 4 MB a ciascun container. Allo stesso modo, se non è impostato alcun limite di CPU per l'attività, l'agente container Amazon ECS ne assegna almeno 2. CPUs

## Uso di tag immutabili con Amazon ECS

Con Amazon ECR, puoi e dovresti preferibilmente configurare immagini con tag immutabili. Ciò impedisce di inviare una versione alterata o aggiornata di un'immagine al tuo archivio di immagini con un tag identico. In questo modo si evita che un utente malintenzionato invii una versione compromessa di un'immagine sull'immagine con lo stesso tag. Utilizzando tag immutabili, ti costringi effettivamente a inviare una nuova immagine con un tag diverso per ogni modifica.

## Evita di utilizzare i container come privilegiati (Amazon EC2)

È consigliabile evitare di eseguire i container come privilegiati. In background, i container vengono eseguiti come `privileged` vengono eseguiti con privilegi estesi sull'host. Ciò significa che il container eredita tutte le funzionalità di Linux assegnate a `root` sull'host. Il suo uso dovrebbe essere severamente limitato o vietato. Consigliamo di impostare la variabile di ambiente dell'agente di container Amazon ECS `ECS_DISABLE_PRIVILEGED` su `true` per evitare che i container vengano eseguiti come `privileged` su determinati host se l'opzione `privileged` non è necessaria. In alternativa è possibile utilizzare AWS Lambda per scansionare le definizioni delle attività per verificare l'utilizzo del `privileged` parametro.

### Note

L'esecuzione di un container come `privileged` non è supportata in Amazon ECS su AWS Fargate.

## Rimozione delle funzionalità di Linux non necessarie dal container

Di seguito è riportato un elenco delle funzionalità di Linux predefinite assegnate ai container Docker. Per ulteriori informazioni su ciascuna funzionalità, consulta [Panoramica delle funzionalità di Linux](#).

```
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_FOWNER, CAP_FSETID, CAP_KILL,
CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_NET_BIND_SERVICE,
CAP_NET_RAW, CAP_SYS_CHROOT, CAP_MKNOD, CAP_AUDIT_WRITE,
CAP_SETFCAP
```

Se un container non richiede tutte le funzionalità del kernel Docker elencate sopra, valuta la possibilità di eliminarle dal container. Per ulteriori informazioni su ciascuna funzionalità del kernel

Docker, consulta [KernelCapabilities](#). Puoi scoprire quali funzionalità sono in uso completando la procedura seguente:

- Installa il pacchetto OS [libcap-ng](#) ed esegui l'utilità `ps cap` per elencare le funzionalità utilizzate da ciascun processo.
- Puoi inoltre adoperare [capsh](#) per decifrare le funzionalità che un processo utilizza.

## Uso di una chiave gestita dal cliente (CMK) per la crittografia delle immagini inviate ad Amazon ECR

È consigliabile utilizzare una chiave gestita dal cliente (CMK) per crittografare le immagini inviate ad Amazon ECR. Le immagini inviate ad Amazon ECR vengono automaticamente crittografate quando sono inattive con una chiave gestita AWS Key Management Service (AWS KMS). Se preferisci usare la tua chiave, Amazon ECR ora supporta la AWS KMS crittografia con chiavi gestite dai clienti (CMK). Prima di abilitare la crittografia lato server con una CMK, consulta le considerazioni elencate nella documentazione sulla [crittografia a riposo](#).

# Tutorial per Amazon ECS

I seguenti tutorial mostrano come eseguire le attività comuni quando si utilizza Amazon ECS.

Puoi utilizzare uno dei seguenti tutorial per saperne di più su come iniziare a usare Amazon ECS.

| Panoramica del tutorial                                        | Ulteriori informazioni                                                                          |  |
|----------------------------------------------------------------|-------------------------------------------------------------------------------------------------|--|
| Inizia a usare Amazon ECS su Fargate.                          | <a href="#">Scopri come creare un'attività Amazon ECS Linux per il tipo di lancio Fargate</a>   |  |
| Inizia a usare i contenitori Windows su Fargate.               | <a href="#">Scopri come creare un'attività Amazon ECS Windows per il tipo di lancio Fargate</a> |  |
| Inizia a usare i contenitori Windows per il tipo di EC2 avvio. | <a href="#">Scopri come creare un'attività Amazon ECS Windows per il tipo di EC2 avvio</a>      |  |

Puoi utilizzare uno qualsiasi dei seguenti tutorial per distribuire attività su Amazon ECS utilizzando AWS CLI

| Panoramica del tutorial                                | Ulteriori informazioni                                                                               |  |
|--------------------------------------------------------|------------------------------------------------------------------------------------------------------|--|
| Crea un'attività Linux per il tipo di lancio Fargate.  | <a href="#">Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI</a>  |  |
| Crea un'attività Windows per il tipo di avvio Fargate. | <a href="#">Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI</a> |  |
| Crea un'attività Linux per il tipo di EC2 avvio.       | <a href="#">Creazione di un'attività Amazon ECS per il tipo di EC2 avvio con AWS CLI</a>             |  |

Puoi usare uno dei seguenti tutorial per saperne di più sul monitoraggio e la registrazione.

| Panoramica del tutorial                                                                                                                                                                                      | Ulteriori informazioni                                                                                                        |  |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|--|
| Imposta una semplice funzione Lambda che ascolti gli eventi delle attività e li scriva in un flusso di log di CloudWatch Logs.                                                                               | <a href="#">Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events</a>                                       |  |
| Configura una regola di EventBridge evento Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita perché uno dei suoi contenitori essenziali è terminato.       | <a href="#">Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS</a> |  |
| Concatena i messaggi di log che originariamente appartenevano a un contesto ma erano suddivisi su più record o righe di registro.                                                                            | <a href="#">Concatenazione di messaggi di log Amazon ECS multilinea o stack-trace</a>                                         |  |
| Implementa i contenitori Fluent Bit sulle loro istanze Windows in esecuzione in Amazon ECS per trasmettere i log generati dalle attività di Windows ad Amazon per la registrazione centralizzata. CloudWatch | <a href="#">Distribuzione di Fluent Bit su contenitori Amazon ECS Windows</a>                                                 |  |

Puoi utilizzare uno dei seguenti tutorial per saperne di più su come utilizzare l'autenticazione Active Directory con un account di servizio gestito di gruppo su Amazon ECS.



| Panoramica del tutorial                                                                                                                                         | Ulteriori informazioni                                                                                        |  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------|--|
| Usa l'account di servizio gestito di gruppo con contenitori Linux attivi. EC2                                                                                   | <a href="#">Utilizzo gMSA per EC2 Linux contenitori su Amazon ECS</a>                                         |  |
| Usa un account di servizio gestito di gruppo con contenitori Windows attivi EC2.                                                                                | <a href="#">Scopri come usare i contenitori g MSAs for EC2 Windows per Amazon ECS</a>                         |  |
| Usa l'account di servizio gestito di gruppo con contenitori Linux su Fargate.                                                                                   | <a href="#">Utilizzo gMSA for Linux container su Fargate</a>                                                  |  |
| Crea un'attività che esegua un contenitore Windows con credenziali per accedere ad Active Directory con un account di servizio gestito di gruppo senza dominio. | <a href="#">Utilizzo di contenitori Amazon ECS Windows con sistema domainless gMSA utilizzando il AWS CLI</a> |  |

## Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI

La procedura seguente illustra come configurare un cluster, registrare una definizione dei processi, eseguire un processo Linux e come eseguire altri scenari comuni in Amazon ECS con la AWS CLI. Usa la versione più recente di AWS CLI Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).

### Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

## Argomenti

- [Prerequisiti](#)
- [Fase 1: creare un cluster](#)
- [Fase 2: Registra una definizione dei processi Linux](#)
- [Fase 3: Elenca le definizioni di attività](#)
- [Fase 4: Crea un servizio](#)
- [Fase 5: Elenca i servizi](#)
- [Fase 6: Descrivi il servizio in esecuzione](#)
- [Fase 7: Test](#)
- [Fase 8: elimina](#)

## Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, [Installazione o aggiornamento alla versione più recente di](#). AWS CLI
- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di [Amazon ECS FullAccess](#) policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine di container ospitata su Amazon ECR Public, quindi il processo deve avere accesso a Internet. Per assegnare alla tua attività un percorso verso Internet, scegli una delle seguenti opzioni.
  - Utilizza una sottorete privata con un gateway NAT con un indirizzo IP elastico.
  - Utilizza una sottorete pubblica e assegna un indirizzo IP pubblico all'attività.

Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).

Per informazioni su gruppi e regole di sicurezza, consulta [Gruppi di sicurezza predefiniti per te VPCs](#) e [Regole di esempio](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

- Se segui questo tutorial utilizzando una sottorete privata, puoi usare Amazon ECS Exec per interagire direttamente con il container e testare l'implementazione. Dovrai creare un ruolo IAM

dell'attività per usare ECS Exec. Per ulteriori informazioni sul ruolo IAM dell'attività e su altri prerequisiti, consulta [Monitorare i contenitori Amazon ECS con Amazon ECS Exec](#).

- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza dover creare la propria istanza. EC2 Per ulteriori informazioni, consulta [Cos'è AWS CloudShell?](#) nella Guida AWS CloudShell per l'utente.

## Fase 1: creare un cluster

Di default, l'account riceve il cluster default.

### Note

Utilizzare il cluster default offre il vantaggio di non dover specificare l'opzione `--cluster cluster_name` nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi specificare `--cluster cluster_name` per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name fargate-cluster
```

Output:

```
{
 "cluster": {
 "status": "ACTIVE",
 "defaultCapacityProviderStrategy": [],
 "statistics": [],
 "capacityProviders": [],
 "tags": [],
 "clusterName": "fargate-cluster",
 "settings": [
 {
 "name": "containerInsights",
 "value": "disabled"
 }
],
 "registeredContainerInstancesCount": 0,
```

```

 "pendingTasksCount": 0,
 "runningTasksCount": 0,
 "activeServicesCount": 0,
 "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster"
 }
}

```

## Fase 2: Registra una definizione dei processi Linux

Prima di eseguire un'attività nel cluster ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente è una semplice definizione di attività che crea un'app Web PHP utilizzando l'immagine di container httpd ospitata su Docker Hub. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta [Definizioni dei processi di Amazon ECS](#). Per questo tutorial, è necessario utilizzare taskRoleArn solo se stai distribuendo l'attività in una sottorete privata e desideri testare l'implementazione. Sostituisci taskRoleArn con il ruolo dell'attività IAM creato per utilizzare ECS Exec, come indicato in [Prerequisiti](#).

```

{
 "family": "sample-fargate",
 "networkMode": "awsvpc",
 "taskRoleArn": "arn:aws:iam::aws_account_id:role/execCommandRole",
 "containerDefinitions": [
 {
 "name": "fargate-app",
 "image": "public.ecr.aws/docker/library/httpd:latest",
 "portMappings": [
 {
 "containerPort": 80,
 "hostPort": 80,
 "protocol": "tcp"
 }
],
 "essential": true,
 "entryPoint": [
 "sh",
 "-c"
],
 "command": [
 "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>\"

```

```
<h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-
foreground\"""
]
 }
],
"requiresCompatibilities": [
 "FARGATE"
],
"cpu": "256",
"memory": "512"
}
```

Salva il JSON della definizione dell'attività come file e trasmettilo con l'opzione `--cli-input-json file://path_to_file.json`.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/fargate-task.json
```

Il comando `register-task-definition` restituisce una descrizione della definizione di attività una volta completata la registrazione.

## Fase 3: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando `list-task-definitions`. Il risultato restituito dal comando mostra i valori `family` e `revision` che puoi utilizzare insieme nelle chiamate `run-task` o `start-task`.

```
aws ecs list-task-definitions
```

Output:

```
{
 "taskDefinitionArns": [
 "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate:1"
]
}
```

## Fase 4: Crea un servizio

Dopo aver registrato un'attività per il tuo account, puoi creare un servizio per le attività registrate nel tuo cluster. Per questo esempio, crei un servizio con un'istanza della definizione delle attività `sample-fargate:1` in esecuzione nel cluster. L'attività richiede un percorso verso Internet, quindi ci sono due modi per ottenere questo risultato. Un modo consiste nell'utilizzare una sottorete privata configurata con un gateway NAT con un indirizzo IP elastico in una sottorete pubblica. Un altro modo consiste nell'utilizzare una sottorete pubblica e assegnare un indirizzo IP pubblico all'attività. Forniamo di seguito entrambi gli esempi.

Esempio di utilizzo di una sottorete privata. L'opzione `enable-execute-command` è necessaria per utilizzare Amazon ECS Exec.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service --
task-definition sample-fargate:1 --desired-count 1 --launch-type "FARGATE" --network-
configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-
abcd1234]}" --enable-execute-command
```

Esempio di utilizzo di una sottorete pubblica.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service --
task-definition sample-fargate:1 --desired-count 1 --launch-type "FARGATE" --network-
configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-
abcd1234],assignPublicIp=ENABLED}"
```

Il comando `create-service` restituisce una descrizione della definizione di attività una volta completata la registrazione.

## Fase 5: Elenca i servizi

Ottieni un elenco dei servizi per il tuo cluster. Verrà visualizzato il servizio creato nella sezione precedente. Potrai utilizzare più avanti il nome del servizio o l'ARN completo restituito da questo comando per la descrizione del servizio.

```
aws ecs list-services --cluster fargate-cluster
```

Output:

```
{
 "serviceArns": [

```

```
 "arn:aws:ecs:region:aws_account_id:service/fargate-cluster/fargate-service"
]
}
```

## Fase 6: Descrivi il servizio in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi il servizio utilizzando il nome del servizio recuperato in precedenza.

```
aws ecs describe-services --cluster fargate-cluster --services fargate-service
```

In caso di esito positivo, verrà restituita una descrizione degli errori del servizio e dei servizi. Ad esempio, nella sezione `services`, sono disponibili informazioni sulle implementazioni, ad esempio lo stato delle attività in esecuzione o in sospeso. È inoltre possibile trovare informazioni sulla definizione delle attività, la configurazione della rete e gli eventi con indicazione del timestamp. Nella sezione Errori, sono disponibili informazioni sugli eventuali errori associati alla chiamata. Per la risoluzione dei problemi, vedere [Messaggi degli eventi di servizio](#). Per ulteriori informazioni sulla descrizione del servizio, vedere [Descrivere i servizi](#).

```
{
 "services": [
 {
 "networkConfiguration": {
 "awsvpcConfiguration": {
 "subnets": [
 "subnet-abcd1234"
],
 "securityGroups": [
 "sg-abcd1234"
],
 "assignPublicIp": "ENABLED"
 }
 },
 "launchType": "FARGATE",
 "enableECSManagedTags": false,
 "loadBalancers": [],
 "deploymentController": {
 "type": "ECS"
 },
 "desiredCount": 1,
 "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
```

```

 "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
 "deploymentConfiguration": {
 "maximumPercent": 200,
 "minimumHealthyPercent": 100
 },
 "createdAt": 1692283199.771,
 "schedulingStrategy": "REPLICA",
 "placementConstraints": [],
 "deployments": [
 {
 "status": "PRIMARY",
 "networkConfiguration": {
 "awsvpcConfiguration": {
 "subnets": [
 "subnet-abcd1234"
],
 "securityGroups": [
 "sg-abcd1234"
],
 "assignPublicIp": "ENABLED"
 }
 },
 "pendingCount": 0,
 "launchType": "FARGATE",
 "createdAt": 1692283199.771,
 "desiredCount": 1,
 "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-
definition/sample-fargate:1",
 "updatedAt": 1692283199.771,
 "platformVersion": "1.4.0",
 "id": "ecs-svc/9223370526043414679",
 "runningCount": 0
 }
],
 "serviceName": "fargate-service",
 "events": [
 {
 "message": "(service fargate-service) has started 2 tasks: (task
53c0de40-ea3b-489f-a352-623bf1235f08) (task d0aec985-901b-488f-9fb4-61b991b332a3).",
 "id": "92b8443e-67fb-4886-880c-07e73383ea83",
 "createdAt": 1510811841.408
 },
 {

```



```

 "message": "(service fargate-service) has started 2 tasks: (task
b4911bee-7203-4113-99d4-e89ba457c626) (task cc5853e3-6e2d-4678-8312-74f8a7d76474).",
 "id": "d85c6ec6-a693-43b3-904a-a997e1fc844d",
 "createdAt": 1510811601.938
 },
 {
 "message": "(service fargate-service) has started 2 tasks: (task
cba86182-52bf-42d7-9df8-b744699e6cfc) (task f4c1ad74-a5c6-4620-90cf-2aff118df5fc).",
 "id": "095703e1-0ca3-4379-a7c8-c0f1b8b95ace",
 "createdAt": 1510811364.691
 }
],
"runningCount": 0,
"status": "ACTIVE",
"serviceRegistries": [],
"pendingCount": 0,
"createdBy": "arn:aws:iam::aws_account_id:user/user_name",
"platformVersion": "LATEST",
"placementStrategy": [],
"propagateTags": "NONE",
"roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
"taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
sample-fargate:1"
}
],
"failures": []
}

```

## Fase 7: Test

### Test di un'attività implementata utilizzando una sottorete pubblica

Descrivi il processo nel servizio in modo da poter ottenere l'interfaccia di rete elastica (ENI) per il processo.

Per prima cosa, ottieni l'attività ARN.

```
aws ecs list-tasks --cluster fargate-cluster --service fargate-service
```

L'output contiene l'attività ARN.

```
{
```

```

 "taskArns": [
 "arn:aws:ecs:us-east-1:123456789012:task/fargate-service/EXAMPLE"
]
}

```

Descrivi il processo e individua l'ID ENI. Usa l'attività ARN per il parametro `tasks`.

```

aws ecs describe-tasks --cluster fargate-cluster --tasks arn:aws:ecs:us-east-1:123456789012:task/service/EXAMPLE

```

Le informazioni relative all'allegato sono elencate nell'output.

```

{
 "tasks": [
 {
 "attachments": [
 {
 "id": "d9e7735a-16aa-4128-bc7a-b2d5115029e9",
 "type": "ElasticNetworkInterface",
 "status": "ATTACHED",
 "details": [
 {
 "name": "subnetId",
 "value": "subnetabcd1234"
 },
 {
 "name": "networkInterfaceId",
 "value": "eni-0fa40520aeEXAMPLE"
 }
]
 }
]
 }
]
 ...
}

```

Descrivi l'ENI per ottenere l'indirizzo IP pubblico.

```

aws ec2 describe-network-interfaces --network-interface-id eni-0fa40520aeEXAMPLE

```

L'indirizzo IP pubblico è indicato nell'output.

```

{
 "NetworkInterfaces": [

```

```

{
 "Association": {
 "IpOwnerId": "amazon",
 "PublicDnsName": "ec2-34-229-42-222.compute-1.amazonaws.com",
 "PublicIp": "198.51.100.2"
 },
 ...
}

```

Inserisci l'indirizzo IP pubblico nel browser Web; dovresti visualizzare una pagina Web che mostra l'applicazione Amazon ECS di esempio.

## Test di un'attività implementata utilizzando una sottorete privata

Descrivi l'attività e individua managedAgents per verificare che ExecuteCommandAgent sia in esecuzione. Annota privateIPv4Address per utilizzarlo in un secondo momento.

```
aws ecs describe-tasks --cluster fargate-cluster --tasks arn:aws:ecs:us-east-1:123456789012:task/fargate-service/EXAMPLE
```

Le informazioni relative all'agente gestito sono elencate nell'output.

```

{
 "tasks": [
 {
 "attachments": [
 {
 "id": "d9e7735a-16aa-4128-bc7a-b2d5115029e9",
 "type": "ElasticNetworkInterface",
 "status": "ATTACHED",
 "details": [
 {
 "name": "subnetId",
 "value": "subnetabcd1234"
 },
 {
 "name": "networkInterfaceId",
 "value": "eni-0fa40520aeEXAMPLE"
 },
 {
 "name": "privateIPv4Address",
 "value": "10.0.143.156"
 }
]
 }
]
 }
]
}

```

```
]
 }
],
 ...
 "containers": [
 {
 ...
 "managedAgents": [
 {
 "lastStartedAt": "2023-08-01T16:10:13.002000+00:00",
 "name": "ExecuteCommandAgent",
 "lastStatus": "RUNNING"
 }
],
 ...
 }
]
 }
}
```

Dopo aver verificato che `ExecuteCommandAgent` sia in esecuzione, puoi eseguire il comando seguente per utilizzare una shell (interprete di comandi) interattiva nel container dell'attività.

```
aws ecs execute-command --cluster fargate-cluster \
 --task arn:aws:ecs:us-east-1:123456789012:task/fargate-service/EXAMPLE \
 --container fargate-app \
 --interactive \
 --command "/bin/sh"
```

Dopo l'esecuzione della shell (interprete di comandi) interattiva, esegui i comandi seguenti per installare cURL.

```
apt update
```

```
apt install curl
```

Dopo aver installato cURL, esegui il comando seguente utilizzando l'indirizzo IP privato ottenuto in precedenza.

```
curl 10.0.143.156
```

Dovresti visualizzare l'equivalente HTML della pagina Web che mostra l'applicazione di esempio di Amazon ECS.

```
<html>
 <head>
 <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style>
 </head>
 <body>
 <div style=color:white;text-align:center>
 <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in
Amazon ECS.</p>
 </div>
 </body>
</html>
```

## Fase 8: elimina

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Elimina il servizio.

```
aws ecs delete-service --cluster fargate-cluster --service fargate-service --force
```

Elimina il cluster.

```
aws ecs delete-cluster --cluster fargate-cluster
```

## Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI

La procedura seguente illustra come configurare un cluster, registrare una definizione dei processi, eseguire un processo Windows e come eseguire altri scenari comuni in Amazon ECS con la AWS CLI. Assicurati di usare la versione più recente di AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).

### Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

## Argomenti

- [Prerequisiti](#)
- [Fase 1: creare un cluster](#)
- [Fase 2: Registrazione di una definizione di attività di Windows](#)
- [Fase 3: Elenca le definizioni di attività](#)
- [Fase 4: Crea un servizio](#)
- [Fase 5: Elenca i servizi](#)
- [Fase 6: Descrivi il servizio in esecuzione](#)
- [Fase 7: eliminare](#)

## Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere [Installazione o aggiornamento alla versione più recente di](#). AWS CLI
- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di [Amazon ECS FullAccess](#) policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine di container ospitata su Docker Hub, quindi l'attività deve avere accesso a Internet. Per assegnare alla tua attività un percorso verso Internet, scegli una delle seguenti opzioni.
  - Utilizza una sottorete privata con un gateway NAT con un indirizzo IP elastico.
  - Utilizza una sottorete pubblica e assegna un indirizzo IP pubblico all'attività.

Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).

Per informazioni su gruppi e regole di sicurezza, consulta [Gruppi di sicurezza predefiniti per te VPCs](#) e [Regole di esempio](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza dover creare la propria EC2 istanza. Per ulteriori informazioni, consulta [Cos'è AWS CloudShell?](#) nella Guida AWS CloudShell per l'utente.

## Fase 1: creare un cluster

Di default, l'account riceve il cluster default.

### Note

Utilizzare il cluster default offre il vantaggio di non dover specificare l'opzione `--cluster` *cluster\_name* nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi specificare `--cluster` *cluster\_name* per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name fargate-cluster
```

Output:

```
{
 "cluster": {
 "status": "ACTIVE",
 "statistics": [],
 "clusterName": "fargate-cluster",
 "registeredContainerInstancesCount": 0,
 "pendingTasksCount": 0,
 "runningTasksCount": 0,
 "activeServicesCount": 0,
 "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster"
 }
}
```

## Fase 2: Registrazione di una definizione di attività di Windows

Prima di eseguire un processo Windows nel cluster Amazon ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente contiene una semplice definizione di attività che crea un'app Web. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta [Definizioni dei processi di Amazon ECS](#).

```
{
 "containerDefinitions": [
 {
 "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html -Type file
-Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your
application is now running on a container in Amazon ECS.</p>'; C:\\ServiceMonitor.exe
w3svc"],
 "entryPoint": [
 "powershell",
 "-Command"
],
 "essential": true,
 "cpu": 2048,
 "memory": 4096,
 "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
 "name": "sample_windows_app",
 "portMappings": [
 {
 "hostPort": 80,
 "containerPort": 80,
 "protocol": "tcp"
 }
]
 }
],
 "memory": "4096",
 "cpu": "2048",
 "networkMode": "awsvpc",
 "family": "windows-simple-iis-2019-core",
 "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
 "runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
 "requiresCompatibilities": ["FARGATE"]
}
```



```
}
```

L'esempio precedente JSON può essere passato a AWS CLI in due modi: è possibile salvare la definizione dell'attività JSON come file e passarla con l'--cli-input-json file://*path\_to\_file.json* opzione.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/fargate-task.json
```

Il comando register-task-definition restituisce una descrizione della definizione di attività una volta completata la registrazione.

### Fase 3: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando list-task-definitions. Il risultato restituito dal comando mostra i valori family e revision che puoi utilizzare insieme nelle chiamate run-task o start-task.

```
aws ecs list-task-definitions
```

Output:

```
{
 "taskDefinitionArns": [
 "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate-windows:1"
]
}
```

### Fase 4: Crea un servizio

Dopo aver registrato un'attività per il tuo account, puoi creare un servizio per le attività registrate nel tuo cluster. Per questo esempio, crei un servizio con un'istanza della definizione delle attività sample-fargate:1 in esecuzione nel cluster. L'attività richiede un percorso verso Internet, quindi ci sono due modi per ottenere questo risultato. Un modo consiste nell'utilizzare una sottorete privata configurata con un gateway NAT con un indirizzo IP elastico in una sottorete pubblica. Un altro modo consiste nell'utilizzare una sottorete pubblica e assegnare un indirizzo IP pubblico all'attività. Forniamo di seguito entrambi gli esempi.

Esempio di utilizzo di una sottorete privata.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service
--task-definition sample-fargate-windows:1 --desired-count 1 --launch-type
"FARGATE" --network-configuration "awsvpcConfiguration={subnets=[subnet-
abcd1234],securityGroups=[sg-abcd1234]}"
```

Esempio di utilizzo di una sottorete pubblica.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service
--task-definition sample-fargate-windows:1 --desired-count 1 --launch-type
"FARGATE" --network-configuration "awsvpcConfiguration={subnets=[subnet-
abcd1234],securityGroups=[sg-abcd1234],assignPublicIp=ENABLED}"
```

Il comando `create-service` restituisce una descrizione della definizione di attività una volta completata la registrazione.

## Fase 5: Elenca i servizi

Ottieni un elenco dei servizi per il tuo cluster. Verrà visualizzato il servizio creato nella sezione precedente. Potrai utilizzare più avanti il nome del servizio o l'ARN completo restituito da questo comando per la descrizione del servizio.

```
aws ecs list-services --cluster fargate-cluster
```

Output:

```
{
 "serviceArns": [
 "arn:aws:ecs:region:aws_account_id:service/fargate-service"
]
}
```

## Fase 6: Descrivi il servizio in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi il servizio utilizzando il nome del servizio recuperato in precedenza.

```
aws ecs describe-services --cluster fargate-cluster --services fargate-service
```

In caso di esito positivo, verrà restituita una descrizione degli errori del servizio e dei servizi. Ad esempio, nella sezione Servizi, sono disponibili informazioni sulle distribuzioni, ad esempio lo stato

delle attività in esecuzione o in sospeso. È inoltre possibile trovare informazioni sulla definizione delle attività, la configurazione della rete e gli eventi con indicazione del timestamp. Nella sezione Errori, sono disponibili informazioni sugli eventuali errori associati alla chiamata. Per la risoluzione dei problemi, vedere [Messaggi degli eventi di servizio](#). Per ulteriori informazioni sulla descrizione del servizio, vedere [Descrivere i servizi](#).

```
{
 "services": [
 {
 "status": "ACTIVE",
 "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate-windows:1",
 "pendingCount": 2,
 "launchType": "FARGATE",
 "loadBalancers": [],
 "roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/ecs.amazonaws.com/AWSServiceRoleForECS",
 "placementConstraints": [],
 "createdAt": 1510811361.128,
 "desiredCount": 2,
 "networkConfiguration": {
 "awsvpcConfiguration": {
 "subnets": [
 "subnet-abcd1234"
],
 "securityGroups": [
 "sg-abcd1234"
],
 "assignPublicIp": "DISABLED"
 }
 },
 "platformVersion": "LATEST",
 "serviceName": "fargate-service",
 "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
 "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
 "deploymentConfiguration": {
 "maximumPercent": 200,
 "minimumHealthyPercent": 100
 },
 "deployments": [
 {
 "status": "PRIMARY",
 "networkConfiguration": {
```

```

 "awsvpcConfiguration": {
 "subnets": [
 "subnet-abcd1234"
],
 "securityGroups": [
 "sg-abcd1234"
],
 "assignPublicIp": "DISABLED"
 },
 "pendingCount": 2,
 "launchType": "FARGATE",
 "createdAt": 1510811361.128,
 "desiredCount": 2,
 "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-
definition/sample-fargate-windows:1",
 "updatedAt": 1510811361.128,
 "platformVersion": "0.0.1",
 "id": "ecs-svc/9223370526043414679",
 "runningCount": 0
 },
],
 "events": [
 {
 "message": "(service fargate-service) has started 2 tasks: (task
53c0de40-ea3b-489f-a352-623bf1235f08) (task d0aec985-901b-488f-9fb4-61b991b332a3).",
 "id": "92b8443e-67fb-4886-880c-07e73383ea83",
 "createdAt": 1510811841.408
 },
 {
 "message": "(service fargate-service) has started 2 tasks: (task
b4911bee-7203-4113-99d4-e89ba457c626) (task cc5853e3-6e2d-4678-8312-74f8a7d76474).",
 "id": "d85c6ec6-a693-43b3-904a-a997e1fc844d",
 "createdAt": 1510811601.938
 },
 {
 "message": "(service fargate-service) has started 2 tasks: (task
cba86182-52bf-42d7-9df8-b744699e6cfc) (task f4c1ad74-a5c6-4620-90cf-2aff118df5fc).",
 "id": "095703e1-0ca3-4379-a7c8-c0f1b8b95ace",
 "createdAt": 1510811364.691
 }
],
 "runningCount": 0,
 "placementStrategy": []

```

```
 }
],
 "failures": []
}
```

## Fase 7: eliminare

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Elimina il servizio.

```
aws ecs delete-service --cluster fargate-cluster --service fargate-service --force
```

Elimina il cluster.

```
aws ecs delete-cluster --cluster fargate-cluster
```

## Creazione di un'attività Amazon ECS per il tipo di EC2 avvio con AWS CLI

La procedura seguente illustra come configurare un cluster, registrare una definizione dei processi, eseguire un processo e come eseguire altri scenari comuni in Amazon ECS con la AWS CLI. Usa la versione più recente di AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).

### Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

### Argomenti

- [Prerequisiti](#)
- [Fase 1: creare un cluster](#)

- [Fase 2: Avvia un'istanza con l'AMI Amazon ECS](#)
- [Fase 3: Elenca le istanze di container](#)
- [Fase 4: Descrivi la tua istanza di container](#)
- [Fase 5: Registra una definizione di attività](#)
- [Fase 6: Elenca le definizioni di attività](#)
- [Fase 7: Esegui un'attività](#)
- [Fase 8: Elenca le attività](#)
- [Fase 9: Descrivi l'attività in esecuzione](#)

## Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere [Installazione o aggiornamento alla versione più recente di](#). AWS CLI
- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di [Amazon ECS FullAccess](#) policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta [the section called “Crea un cloud privato virtuale”](#).
- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza la necessità di creare la propria EC2 istanza. Per ulteriori informazioni, consulta [Cos'è AWS CloudShell?](#) nella Guida AWS CloudShell per l'utente.

## Fase 1: creare un cluster

All'avvio della prima istanza di container, di default l'account riceve il cluster default.

### Note

Utilizzare il cluster default offre il vantaggio di non dover specificare l'opzione `--cluster cluster_name` nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi

specificare `--cluster cluster_name` per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name MyCluster
```

Output:

```
{
 "cluster": {
 "clusterName": "MyCluster",
 "status": "ACTIVE",
 "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/MyCluster"
 }
}
```

## Fase 2: Avvia un'istanza con l'AMI Amazon ECS

Devi disporre di un'istanza di container Amazon ECS nel cluster prima di potervi eseguire operazioni. Se non disponi di istanze di container nel cluster, consulta [Avvio di un'istanza di container Linux di Amazon ECS](#) per ulteriori informazioni.

## Fase 3: Elenca le istanze di container

Entro pochi minuti dall'avvio dell'istanza di container, l'agente Amazon ECS registra quest'ultima nel cluster predefinito. Per visualizzare l'elenco delle istanze di container in un cluster, esegui questo comando:

```
aws ecs list-container-instances --cluster default
```

Output:

```
{
 "containerInstanceArns": [
 "arn:aws:ecs:us-east-1:aws_account_id:container-instance/container_instance_ID"
]
}
```

## Fase 4: Descrivi la tua istanza di container

Dopo aver ottenuto l'ARN o l'ID di un'istanza di container, puoi utilizzare il comando `describe-container-instances` per ottenere informazioni importanti sull'istanza, ad esempio le risorse di CPU e memoria registrate e quelle ancora disponibili.

```
aws ecs describe-container-instances --cluster default --container-
instances container_instance_ID
```

Output:

```
{
 "failures": [],
 "containerInstances": [
 {
 "status": "ACTIVE",
 "registeredResources": [
 {
 "integerValue": 1024,
 "longValue": 0,
 "type": "INTEGER",
 "name": "CPU",
 "doubleValue": 0.0
 },
 {
 "integerValue": 995,
 "longValue": 0,
 "type": "INTEGER",
 "name": "MEMORY",
 "doubleValue": 0.0
 },
 {
 "name": "PORTS",
 "longValue": 0,
 "doubleValue": 0.0,
 "stringSetValue": [
 "22",
 "2376",
 "2375",
 "51678"
],
 "type": "STRINGSET",
 }
]
 }
]
}
```



```

 "integerValue": 0
 },
 {
 "name": "PORTS_UDP",
 "longValue": 0,
 "doubleValue": 0.0,
 "stringSetValue": [],
 "type": "STRINGSET",
 "integerValue": 0
 }
],
"ec2InstanceId": "instance_id",
"agentConnected": true,
"containerInstanceArn": "arn:aws:ecs:us-west-2:aws_account_id:container-
instance/container_instance_ID",
"pendingTasksCount": 0,
"remainingResources": [
 {
 "integerValue": 1024,
 "longValue": 0,
 "type": "INTEGER",
 "name": "CPU",
 "doubleValue": 0.0
 },
 {
 "integerValue": 995,
 "longValue": 0,
 "type": "INTEGER",
 "name": "MEMORY",
 "doubleValue": 0.0
 },
 {
 "name": "PORTS",
 "longValue": 0,
 "doubleValue": 0.0,
 "stringSetValue": [
 "22",
 "2376",
 "2375",
 "51678"
],
 "type": "STRINGSET",
 "integerValue": 0
 }
],

```

```

 {
 "name": "PORTS_UDP",
 "longValue": 0,
 "doubleValue": 0.0,
 "stringSetValue": [],
 "type": "STRINGSET",
 "integerValue": 0
 }
],
 "runningTasksCount": 0,
 "attributes": [
 {
 "name": "com.amazonaws.ecs.capability.privileged-container"
 },
 {
 "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
 },
 {
 "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
 },
 {
 "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
 },
 {
 "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
 },
 {
 "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
 }
],
 "versionInfo": {
 "agentVersion": "1.5.0",
 "agentHash": "b197edd",
 "dockerVersion": "DockerVersion: 1.7.1"
 }
}
]
}

```

Puoi anche trovare l'ID dell' EC2 istanza Amazon che puoi utilizzare per monitorare l'istanza nella EC2 console Amazon o con il `aws ec2 describe-instances --instance-id instance_id` comando.

## Fase 5: Registra una definizione di attività

Prima di eseguire un'attività nel cluster ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente illustra una semplice definizione di attività che utilizza un'immagine busybox da Docker Hub ed entra semplicemente in sospensione per 360 secondi. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta [Definizioni dei processi di Amazon ECS](#).

```
{
 "containerDefinitions": [
 {
 "name": "sleep",
 "image": "busybox",
 "cpu": 10,
 "command": [
 "sleep",
 "360"
],
 "memory": 10,
 "essential": true
 }
],
 "family": "sleep360"
}
```

L'esempio precedente JSON può essere passato a AWS CLI in due modi: è possibile salvare la definizione dell'attività JSON come file e passarla con l'--cli-input-json file://*path\_to\_file.json* opzione. Oppure puoi inserire il carattere di escape prima delle virgolette nel JSON e trasmettere le definizioni del container JSON nella riga di comando, come nell'esempio riportato di seguito. Se scegli di trasmettere le definizioni del container nella riga di comando, è necessario aggiungere al comando il parametro --family utilizzato per mantenere la reciproca associazione tra più versioni della definizione di attività.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/sleep360.json
```

Per utilizzare una stringa JSON per le definizioni dei container:

```
aws ecs register-task-definition --family sleep360 --container-definitions "[{\\"name\\":\\"sleep\\",\\"image\\":\\"busybox\\",\\"cpu\\":10,\\"command\\":[\\"sleep\\",\\"360\\"],\\"memory\\":10,\\"essential\\":true}]"
```

Il comando `register-task-definition` restituisce una descrizione della definizione di attività una volta completata la registrazione.

```
{
 "taskDefinition": {
 "volumes": [],
 "taskDefinitionArn": "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep360:1",
 "containerDefinitions": [
 {
 "environment": [],
 "name": "sleep",
 "mountPoints": [],
 "image": "busybox",
 "cpu": 10,
 "portMappings": [],
 "command": [
 "sleep",
 "360"
],
 "memory": 10,
 "essential": true,
 "volumesFrom": []
 }
],
 "family": "sleep360",
 "revision": 1
 }
}
```

## Fase 6: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando `list-task-definitions`. Il risultato restituito dal comando mostra i valori `family` e `revision` che puoi utilizzare insieme nelle chiamate `run-task` o `start-task`.

```
aws ecs list-task-definitions
```

Output:

```
{
 "taskDefinitionArns": [
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep300:1",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep300:2",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep360:1",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:3",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:4",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:5",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:6"
]
}
```

## Fase 7: Esegui un'attività

Dopo aver registrato un'attività per il tuo account e aver avviato un'istanza di container registrata nel cluster, puoi eseguire l'attività registrata in tale cluster. In questo esempio, una singola istanza della definizione di attività `sleep360:1` viene posizionata nel tuo cluster predefinito.

```
aws ecs run-task --cluster default --task-definition sleep360:1 --count 1
```

Output:

```
{
 "tasks": [
 {
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
 "overrides": {
 "containerOverrides": [
 {
 "name": "sleep"
 }
]
 },
 "lastStatus": "PENDING",
 "containerInstanceArn": "arn:aws:ecs:us-east-1:aws_account_id:container-instance/container_instance_ID",
 "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/default",
 "desiredStatus": "RUNNING",
 "taskDefinitionArn": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/sleep360:1",
 }
]
}
```

```

 "containers": [
 {
 "containerArn": "arn:aws:ecs:us-
east-1:aws_account_id:container/container_ID",
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
 "lastStatus": "PENDING",
 "name": "sleep"
 }
]
 }
}

```

## Fase 8: Elenca le attività

Ottieni un elenco delle attività per il tuo cluster. Verrà visualizzata l'attività eseguita nella sezione precedente. Potrai utilizzare più avanti l'ID attività o l'ARN completo restituito da questo comando per la descrizione dell'attività.

```
aws ecs list-tasks --cluster default
```

Output:

```

{
 "taskArns": [
 "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID"
]
}

```

## Fase 9: Descrivi l'attività in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi l'attività utilizzando l'ID recuperato in precedenza.

```
aws ecs describe-tasks --cluster default --task task_ID
```

Output:

```

{
 "failures": [],

```

```

"tasks": [
 {
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
 "overrides": {
 "containerOverrides": [
 {
 "name": "sleep"
 }
]
 },
 "lastStatus": "RUNNING",
 "containerInstanceArn": "arn:aws:ecs:us-east-1:aws_account_id:container-
instance/container_instance_ID",
 "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/default",
 "desiredStatus": "RUNNING",
 "taskDefinitionArn": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/
sleep360:1",
 "containers": [
 {
 "containerArn": "arn:aws:ecs:us-
east-1:aws_account_id:container/container_ID",
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
 "lastStatus": "RUNNING",
 "name": "sleep",
 "networkBindings": []
 }
]
 }
]
}

```

## Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events

Scopri come configurare una semplice funzione Lambda che ascolta gli eventi delle attività e li scrive in un flusso di log di CloudWatch Logs.

### Prerequisito: configurazione di un cluster di verifica

Se non disponi di un cluster in esecuzione da cui acquisire eventi, segui la procedura descritta in [the section called “Creazione di un cluster per il tipo di lancio Fargate”](#) per crearne uno. Al termine del

tutorial, potrai eseguire un processo su questo cluster per verificare la corretta configurazione della funzione Lambda.

## Fase 1: Creazione della funzione Lambda

In questa procedura, viene creata una funzione Lambda semplice che funge da destinazione per i messaggi del flusso di eventi di Amazon ECS.

1. Apri la console all' AWS Lambda indirizzo. <https://console.aws.amazon.com/lambda/>
2. Scegli Crea funzione.
3. In Author from scratch (Crea da zero) effettua le seguenti operazioni:
  - a. In Name (Nome), immetti un valore.
  - b. In Runtime, scegli la versione di Python, ad esempio, Python 3.9.
  - c. In Role (Ruolo), scegliere Create a new role with basic Lambda permissions (Crea un nuovo ruolo con le autorizzazioni Lambda di base).
4. Scegli Crea funzione.
5. Nella sezione Function code (Codice della funzione), modifica il codice di esempio affinché corrisponda all'esempio seguente:

```
import json

def lambda_handler(event, context):
 if event["source"] != "aws.ecs":
 raise ValueError("Function only supports input from events with a source
 type of: aws.ecs")

 print('Here is the event:')
 print(json.dumps(event))
```

Si tratta di una funzione Python 3.9 semplice che stampa l'evento inviato da Amazon ECS. Se tutto è configurato correttamente, alla fine di questo tutorial, vedrai che i dettagli dell'evento vengono visualizzati nel flusso di log CloudWatch Logs associato a questa funzione Lambda.

6. Scegli Save (Salva).



## Fase 2: Registrazione di una regola di evento

Successivamente, crei una regola CloudWatch degli eventi che acquisisce gli eventi delle attività provenienti dai tuoi cluster Amazon ECS. Questa regola acquisisce tutti gli eventi provenienti da tutti i cluster all'interno dell'account in cui è definita. Gli stessi messaggi di attività contengono informazioni sull'origine dell'evento, tra cui il cluster in cui esso risiede, utilizzabile per filtrare e ordinare gli eventi in modo programmatico.

### Note

Quando usi AWS Management Console per creare una regola di evento, la console aggiunge automaticamente le autorizzazioni IAM necessarie per concedere a CloudWatch Events l'autorizzazione a chiamare la tua funzione Lambda. Se stai creando una regola di evento utilizzando il AWS CLI, devi concedere questa autorizzazione in modo esplicito. Per ulteriori informazioni, consulta [Events in Amazon EventBridge e Amazon EventBridge Event Patterns](#) nella Amazon EventBridge User Guide.

Come instradare gli eventi alla funzione Lambda

1. Apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nel pannello di navigazione, selezionare prima Events (Eventi), quindi Rules (Regole) e infine Create rule (Crea regola).
3. In Event Source (Origine eventi), scegli ECS come origine evento. Di default, questa regola si applica a tutti gli eventi di Amazon ECS per tutti i gruppi Amazon ECS. In alternativa, puoi selezionare eventi specifici o un gruppo specifico di Amazon ECS.
4. In Targets (Destinazioni), scegli Add target (Aggiungi destinazione), in Target type (Tipo di destinazione) scegli Lambda function (Funzione Lambda) quindi seleziona la tua funzione Lambda.
5. Scegli Configura dettagli.
6. In Rule definition (Definizione regola), digita un nome e una descrizione per la regola, quindi seleziona Create rule (Crea regola).

## Fase 3: creazione di una definizione di attività

Crea una definizione di attività.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel riquadro di navigazione, scegli Definizioni di attività.
3. Scegli Create new Task Definition (Crea nuova definizione di attività), Create new revision with JSON (Crea nuova revisione con JSON).
4. Copia e incolla la seguente definizione di attività di esempio nella casella, quindi scegli Save (Salva).

```
{
 "containerDefinitions": [
 {
 "command": [
 "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top: 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon ECS.</p> </div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-foreground\"\"",
],
 "entryPoint": [
 "sh",
 "-c"
],
 "essential": true,
 "image": "public.ecr.aws/docker/library/httpd:2.4",
 "logConfiguration": {
 "logDriver": "awslogs",
 "options": {
 "awslogs-group" : "/ecs/fargate-task-definition",
 "awslogs-region": "us-east-1",
 "awslogs-stream-prefix": "ecs"
 }
 },
 "name": "sample-fargate-app",
 "portMappings": [
 {
 "containerPort": 80,
 "hostPort": 80,
 "protocol": "tcp"
 }
]
 }
],
}
```

```
"cpu": "256",
"executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
"family": "fargate-task-definition",
"memory": "512",
"networkMode": "awsvpc",
"runtimePlatform": {
 "operatingSystemFamily": "LINUX"
},
"requiresCompatibilities": [
 "FARGATE"
]
}
```

5. Scegli Create (Crea).

## Fase 4: Test della regola

Infine, crei una regola CloudWatch degli eventi che acquisisce gli eventi delle attività provenienti dai tuoi cluster Amazon ECS. Questa regola acquisisce tutti gli eventi provenienti da tutti i cluster all'interno dell'account in cui è definita. Gli stessi messaggi di attività contengono informazioni sull'origine dell'evento, tra cui il cluster in cui esso risiede, utilizzabile per filtrare e ordinare gli eventi in modo programmatico.

Per testare la regola

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Scegli Task definitions (Definizioni di attività).
3. Scegli console-sample-app-static, quindi scegli Distribuisci, Esegui nuova attività.
4. Per Cluster, scegli l'impostazione predefinita, quindi scegli Deploy (Implementa).
5. Apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
6. Nel pannello di navigazione, scegli Logs e seleziona il gruppo di log per la tua funzione Lambda (ad esempio, *my-function* /aws/lambda/).
7. Seleziona un flusso di log per visualizzare i dati di evento.

# Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS

Configura una regola di EventBridge evento Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita perché uno dei suoi contenitori essenziali è terminato. L'evento invia solo eventi di processo con una specifica proprietà `stoppedReason` all'argomento Amazon SNS designato.

## Prerequisito: configurazione di un cluster di verifica

Se non disponi di un cluster in esecuzione da cui acquisire eventi, segui la procedura descritta in [Nozioni di base sull'utilizzo della console con i container Linux su AWS Fargate](#) per crearne uno. Alla fine di questo tutorial, esegui un'attività su questo cluster per verificare di aver configurato correttamente l'argomento e la EventBridge regola di Amazon SNS.

## Prerequisito: configurazione delle autorizzazioni per Amazon SNS

EventBridge Per consentire la pubblicazione su un argomento Amazon SNS, usa i comandi `aws sns get-topic-attributes` e `aws sns set-topic-attributes`

Per ulteriori informazioni su come aggiungere l'autorizzazione, consulta la sezione [Amazon SNS permissions](#) (Autorizzazioni di Amazon SNS) nella Guida per gli sviluppatori di Servizio di notifica semplice Amazon.

Aggiungi le autorizzazioni seguenti:

```
{
 "Sid": "PublishEventsToMyTopic",
 "Effect": "Allow",
 "Principal": {
 "Service": "events.amazonaws.com"
 },
 "Action": "sns: Publish",
 "Resource": "arn:aws:sns:region:account-id:TaskStoppedAlert",
}
```

## Fase 1: Creazione e sottoscrizione a un argomento Amazon SNS

In questo tutorial, configuri un argomento Amazon SNS che funga da destinazione evento per la nuova regola di evento.

Per informazioni sulla creazione e l'abbonamento a un argomento Amazon SNS, consulta [Nozioni di base su Amazon SNS](#) nella Guida per gli sviluppatori di Servizio di notifica semplice Amazon e utilizza la tabella seguente per determinare quali opzioni selezionare.

| Opzione    | Valore                                            |  |
|------------|---------------------------------------------------|--|
| Tipo       | Standard                                          |  |
| Nome       | TaskStoppedAlert                                  |  |
| Protocollo | E-mail                                            |  |
| Endpoint   | Un indirizzo e-mail a cui hai attualmente accesso |  |

## Fase 2: Registrazione di una regola di evento

Successivamente, registra una regola dell'evento che acquisisca solo eventi di arresto dell'attività per attività con container interrotti.

Per informazioni su come creare e sottoscrivere un argomento di Amazon SNS, consulta [Create a rule EventBridge in Amazon](#) nella Amazon EventBridge User Guide e utilizza la tabella seguente per determinare quali opzioni selezionare.

| Opzione           | Valore                                        |  |
|-------------------|-----------------------------------------------|--|
| Tipo di regola    | Regola con un modello di evento               |  |
| Origine eventi    | AWS eventi o eventi per i EventBridge partner |  |
| Modello di evento | Modello personalizzato (editor JSON)          |  |
| Modello di evento | <pre>{   "source": [     "aws.ecs"   ],</pre> |  |

| Opzione                            | Valore                                                                                                                                                                                                                                                                          |  |
|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
|                                    | <pre>       "detail-type":[         "ECS Task State         Change"       ],       "detail":{         "lastStatus":[           "STOPPED"         ],         "stoppedReason":[           "Essential           container in task           exited"         ]       }     } </pre> |  |
| Target type (Tipo di destinazione) | AWS servizio                                                                                                                                                                                                                                                                    |  |
| Target                             | Argomento SNS                                                                                                                                                                                                                                                                   |  |
| Argomento                          | TaskStoppedAlert (L'argomento che hai creato nel passaggio 1)                                                                                                                                                                                                                   |  |

## Fase 3: Test del tuo articolo

Verificare che la regola funzioni eseguendo un'attività che termine poco dopo l'avvio. Se la regola dell'evento è configurata correttamente, riceverai una e-mail contenente il testo dell'evento entro pochi minuti. Se si dispone di una definizione di attività esistente in grado di soddisfare i requisiti della regola, eseguire un'attività utilizzando tale definizione. In caso contrario, le fasi seguenti descrivono la procedura di registrazione di una definizione di attività Fargate e la relativa esecuzione.

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).

4. Nella casella dell'editor JSON, modifica il file JSON e copia quanto segue nell'editor.

```
{
 "containerDefinitions": [
 {
 "command": [
 "sh",
 "-c",
 "sleep 5"
],
 "essential": true,
 "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
 "name": "test-sleep"
 }
],
 "cpu": "256",
 "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
 "family": "fargate-task-definition",
 "memory": "512",
 "networkMode": "awsvpc",
 "requiresCompatibilities": [
 "FARGATE"
]
}
```

5. Scegli Create (Crea).

Per eseguire un'attività dalla console

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegli il cluster che hai creato nei prerequisiti.
3. Dalla scheda Processi, scegli Esegui nuovo processo.
4. Per Tipo di applicazione, scegli Processo.
5. Per Definizione dell'attività, scegli fargate-task-definition.
6. Per Desired tasks (Attività desiderate), specifica il numero di attività da avviare.
7. Scegli Create (Crea).

# Concatenazione di messaggi di log Amazon ECS multilinea o stack-trace

A partire dalla versione 2.22.0 AWS di Fluent Bit, è incluso un filtro multilinea. Il filtro multilinea aiuta a concatenare i messaggi di log originariamente appartenenti a un contesto ma suddivisi su più record o righe di registro. Per ulteriori informazioni sul filtro multilinea, consulta la [documentazione di Fluent Bit](#).

Esempi comuni di messaggi di log divisi sono:

- Tracce dello stack.
- Applicazioni che stampano i log su più righe.
- Registra i messaggi che sono stati divisi perché erano più lunghi della dimensione massima del buffer di runtime specificata. [È possibile concatenare i messaggi di registro suddivisi in base al runtime del contenitore seguendo l'esempio su GitHub: Esempio: FireLens Concatenate log di container parziali/divisi.](#)

## Autorizzazioni IAM richieste

Disponi delle autorizzazioni IAM necessarie affinché l'agente container estragga le immagini del contenitore da Amazon ECR e che il container instrada i log verso Logs. CloudWatch

Per tali autorizzazioni devi disporre anche dei seguenti ruoli:

- Un ruolo IAM del processo.
- Un ruolo IAM per l'esecuzione di attività.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo <https://console.aws.amazon.com/iam/>.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.



4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [{
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogStream",
 "logs:CreateLogGroup",
 "logs:PutLogEvents"
],
 "Resource": "*"
 }]
}
```

6. Scegli Next (Successivo).

#### Note

È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta [Modifica della struttura delle policy](#) nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

## Determinazione dei casi in cui utilizzare l'impostazione del log multilinea

Di seguito sono riportati alcuni esempi di frammenti di log che vedi nella console CloudWatch Logs con l'impostazione di registro predefinita. Puoi guardare la riga che inizia con `log` per determinare se è necessario il filtro multilinea. Quando il contesto è lo stesso, puoi utilizzare l'impostazione di registro multilinea. In questo esempio, il contesto è «com.myproject.model». MyProject».

```
2022-09-20T15:47:56:595-05-00 {"container_id":
 "82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-
```

```
app", "source": "stdout", "log": ": " at com.myproject.modele.
(MyProject.badMethod.java:22)",
{
 "container_id": "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
 "container_name": ": "example-app",
 "source": "stdout",
 "log": ": " at com.myproject.model.MyProject.badMethod(MyProject.java:22)",
 "ecs_cluster": "default",
 "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE",
 "ecs_task_definition": "firelense-example-multiline:3"
}
```

```
2022-09-20T15:47:56:595-05-00 {"container_id":
"82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-app", "stdout",
"log": ": " at com.myproject.modele.(MyProject.oneMoreMethod.java:18)",
{
 "container_id": "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
 "container_name": ": "example-app",
 "source": "stdout",
 "log": ": " at
com.myproject.model.MyProject.oneMoreMethod(MyProject.java:18)",
 "ecs_cluster": "default",
 "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE",
 "ecs_task_definition": "firelense-example-multiline:3"
}
```

Dopo aver utilizzato l'impostazione del log multilinea, l'output sarà simile all'esempio seguente.

```
2022-09-20T15:47:56:595-05-00 {"container_id":
"82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-app",
"stdout",...
{
 "container_id": "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
 "container_name": ": "example-app",
 "source": "stdout",
 "log": "September 20, 2022 06:41:48 Exception in thread \"main\"
java.lang.RuntimeException: Something has gone wrong, aborting!\n
at com.myproject.module.MyProject.badMethod(MyProject.java:22)\n at
```

```
at com.myproject.model.MyProject.oneMoreMethod(MyProject.java:18)
com.myproject.module.MyProject.main(MyProject.java:6)",
 "ecs_cluster": "default",
 "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE",
 "ecs_task_definition": "firelense-example-multiline:2"
}
```

## Opzioni di analisi e concatenazione

Per analizzare i log e concatenare righe divise a causa di nuove righe, puoi utilizzare una di queste due opzioni.

- Utilizza il tuo file parser contenente le regole per analizzare e concatenare le righe che appartengono allo stesso messaggio.
- Utilizzare un parser integrato Fluent Bit. Per l'elenco dei linguaggi supportati dai parser integrati Fluent Bit, consulta la [documentazione di Fluent Bit](#).

Il seguente tutorial illustra i passaggi per ogni caso d'uso. I passaggi mostrano come concatenare più righe e inviare i log ad Amazon. CloudWatch Puoi specificare una destinazione diversa per i log.

### Esempio: utilizzare un parser creato

In questo esempio eseguirai i passaggi seguenti:

1. Crea e carica l'immagine per un container Fluent Bit.
2. Crea e carica l'immagine per un'applicazione demo multilinea che esegue, fallisce e genera una traccia dello stack multilinea.
3. Crea la definizione di un processo e avviare il processo.
4. Visualizza i log per verificare che i messaggi che si estendono su più righe appaiano concatenati.

#### Creare e caricare l'immagine per un container Fluent Bit

Questa immagine includerà il file parser in cui si specifica l'espressione regolare e un file di configurazione che fa riferimento al file parser.

1. Crea una cartella con il nome `FluentBitDockerImage`.

2. Nella cartella, crea il file parser contenente le regole per analizzare il log e concatenare le righe che appartengono allo stesso messaggio.
  - a. Incolla i contenuti seguenti nel file parser:

```
[MULTILINE_PARSER]
 name multiline-regex-test
 type regex
 flush_timeout 1000
 #
 # Regex rules for multiline parsing
 # -----
 #
 # configuration hints:
 #
 # - first state always has the name: start_state
 # - every field in the rule must be inside double quotes
 #
 # rules | state name | regex pattern | next state
 # -----|-----|-----|-----
 rule "start_state" "/(Dec \d+ \d+:\d+:\d+)(.*)/" "cont"
 rule "cont" "/^\s+at.*/" "cont"
```

Mentre personalizzi il modello di regex, ti consigliamo di utilizzare un editor di espressioni regolari per provare l'espressione.

- b. Salva il file con nome `parsers_multiline.conf`.
3. All'interno della cartella `FluentBitDockerImage`, crea un file di configurazione personalizzato che fa riferimento al file parser creato nel passaggio precedente.

Per ulteriori informazioni sul file di configurazione personalizzato, consulta [Specifica di un file di configurazione personalizzato](#) nella Guida per gli sviluppatori di Amazon Elastic Container Service

- a. Incolla i contenuti seguenti nel file:

```
[SERVICE]
 flush 1
 log_level info
 parsers_file /parsers_multiline.conf

[FILTER]
```

```
name multiline
match *
multiline.key_content log
multiline.parser multiline-regex-test
```

 **Note**

È necessario utilizzare il percorso assoluto del parser.

- a. Salva il file con nome `extra.conf`.
4. All'interno della cartella `FluentBitDockerImage`, crea il `Dockerfile` con l'immagine Fluent Bit e il parser e i file di configurazione creati.
- a. Incolla i contenuti seguenti nel file:

```
FROM public.ecr.aws/aws-observability/aws-for-fluent-bit:latest

ADD parsers_multiline.conf /parsers_multiline.conf
ADD extra.conf /extra.conf
```

- b. Salva il file con nome `Dockerfile`.
5. Utilizzando il `Dockerfile`, crea un'immagine Fluent Bit personalizzata con il parser e i file di configurazione personalizzati inclusi.

 **Note**

Puoi posizionare il file di analisi e il file di configurazione in qualsiasi punto dell'immagine Docker, tranne `/fluent-bit/etc/fluent-bit.conf` quando questo percorso del file viene utilizzato da FireLens

- a. Crea l'immagine: `docker build -t fluent-bit-multiline-image .`  
  
Dove: `fluent-bit-multiline-image` è il nome dell'immagine in questo esempio.
- b. Verifica che l'immagine sia stata creata correttamente: `docker images --filter reference=fluent-bit-multiline-image`

In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

6. Carica l'immagine Fluent Bit personalizzata in Amazon Elastic Container Registry.

- a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name fluent-bit-multiline-repo --region us-east-1`

Dove: `fluent-bit-multiline-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

L'output fornisce i dettagli del nuovo repository.

- b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag fluent-bit-multiline-image repositoryUri`

Esempio: `docker tag fluent-bit-multiline-image  
xxxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-repo`

- c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images --filter reference=repositoryUri`

Nell'output, il nome del repository cambia da `fluent-bit-multiline-repo` a `repositoryUri`

- d. Effettua l'autenticazione su Amazon ECR eseguendo il comando `aws ecr get-login-password` e specificando l'ID del log in cui desideri eseguire l'autenticazione: `aws ecr get-login-password | docker login --username AWS --password-stdin registry ID.dkr.ecr.region.amazonaws.com`

Esempio: `aws ecr get-login-password | docker login --username AWS --password-stdin  
xxxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com`

Viene visualizzato un messaggio di accesso riuscito.

- e. Invia l'immagine ad Amazon ECR: `docker push registry ID.dkr.ecr.region.amazonaws.com/repository name`

Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/  
fluent-bit-multiline-repo`

## Crea e carica l'immagine per un'applicazione demo multilinea

Questa immagine includerà un file di script Python che esegue l'applicazione e un file di log di esempio.

Quando esegui il processo, l'applicazione simula le esecuzioni, quindi fallisce e crea una traccia dello stack.

1. Crea una cartella denominata `multiline-app`: `mkdir multiline-app`
2. Crea un file di script Python.
  - a. Nella cartella `multiline-app`, crea un file con il nome `main.py`.
  - b. Incolla i contenuti seguenti nel file:

```
import os
import time
file1 = open('/test.log', 'r')
Lines = file1.readlines()

count = 0

for i in range(10):
 print("app running normally...")
 time.sleep(1)

Strips the newline character
for line in Lines:
 count += 1
 print(line.rstrip())
print(count)
print("app terminated.")
```

- c. Salvare il file `main.py`.
3. Crea un file di log di esempio.
  - a. Nella cartella `multiline-app`, crea un file con il nome `test.log`.
  - b. Incolla i contenuti seguenti nel file:

```
single line...
Dec 14 06:41:08 Exception in thread "main" java.lang.RuntimeException:
Something has gone wrong, aborting!
 at com.myproject.module.MyProject.badMethod(MyProject.java:22)
 at com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)
 at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)
 at com.myproject.module.MyProject.someMethod(MyProject.java:10)
```

```
 at com.myproject.module.MyProject.main(MyProject.java:6)
another line...
```

c. Salvare il file `test.log`.

4. All'interno della cartella `multiline-app`, crea il Dockerfile.

a. Incolla i contenuti seguenti nel file:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest
ADD test.log /test.log

RUN yum upgrade -y && yum install -y python3

WORKDIR /usr/local/bin

COPY main.py .

CMD ["python3", "main.py"]
```

b. Salvare il file `Dockerfile`.

5. Usando il Dockerfile, crea un'immagine.

a. Crea l'immagine: `docker build -t multiline-app-image .`

Dove: `multiline-app-image` è il nome dell'immagine in questo esempio.

b. Verifica che l'immagine sia stata creata correttamente: `docker images --filter reference=multiline-app-image`

In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

6. Carica l'immagine nel registro del container Amazon Elastic.

a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name multiline-app-repo --region us-east-1`

Dove: `multiline-app-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

L'output fornisce i dettagli del nuovo repository. Annota il valore `repositoryUri` poiché sarà necessario nei passaggi successivi.



- b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente:  
`docker tag multiline-app-image repositoryUri`

Esempio: `docker tag multiline-app-image xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

- c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images --filter reference=repositoryUri`

Nell'output, il nome del repository cambia da `multiline-app-repo` al valore `repositoryUri`.

- d. Invia l'immagine ad Amazon ECR: `docker push aws_account_id.dkr.ecr.region.amazonaws.com/repository name`

Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

Crea la definizione di un processo e avviare il processo

1. Crea un file di definizione del processo con il nome del file `multiline-task-definition.json`.
2. Incolla i contenuti seguenti nel file `multiline-task-definition.json`:

```
{
 "family": "firelens-example-multiline",
 "taskRoleArn": "task role ARN",
 "executionRoleArn": "execution role ARN",
 "containerDefinitions": [
 {
 "essential": true,
 "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-image:latest",
 "name": "log_router",
 "firelensConfiguration": {
 "type": "fluentbit",
 "options": {
 "config-file-type": "file",
 "config-file-value": "/extra.conf"
 }
 },
 "memoryReservation": 50
 }
]
}
```

```

 },
 {
 "essential": true,
 "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/multiline-app-
image:latest",
 "name": "app",
 "logConfiguration": {
 "logDriver": "awsfirelens",
 "options": {
 "Name": "cloudwatch_logs",
 "region": "us-east-1",
 "log_group_name": "multiline-test/application",
 "auto_create_group": "true",
 "log_stream_prefix": "multiline-"
 }
 },
 "memoryReservation": 100
 }
],
 "requiresCompatibilities": ["FARGATE"],
 "networkMode": "awsvpc",
 "cpu": "256",
 "memory": "512"
}

```

Sostituisci quanto segue nella definizione del processo multiline-task-definition.json:

a. *task role ARN*

Per trovare l'ARN del ruolo del processo, vai alla Console IAM. Scegli Roles (Roli) e trova il ruolo del processo ecs-task-role-for-firelens che hai creato. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

b. *execution role ARN*

Per trovare l'ARN del ruolo di esecuzione, vai alla Console IAM. Scegli Roles (Ruoli) e trova il ruolo ecsTaskExecutionRole. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

c. *aws\_account\_id*

Per trovare l'aws\_account\_id, accedi alla AWS Management Console. Scegli il tuo nome utente in alto a destra e copia il tuo ID account.

d. *us-east-1*

Sostituisci la regione se necessario.

3. Registra il file di definizione del processo: `aws ecs register-task-definition --cli-input-json file://multiline-task-definition.json --region region`
4. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
5. Nel pannello di navigazione, scegli Task Definitions (Definizioni dei processi) quindi scegli la famiglia `firelens-example-multiline`, perché sopra abbiamo registrato la definizione dell'attività processo in questa famiglia nella prima riga della definizione del processo.
6. Scegli la versione più recente.
7. Scegli Distribuisci, Esegui attività.
8. Nella pagina Esegui attività, in Cluster scegli il cluster, quindi in Reti scegli le sottoreti disponibili per tale attività in Sottoreti.
9. Scegli Create (Crea).

Verifica che i messaggi di log multilinea in Amazon CloudWatch appaiano concatenati

1. Apri la console all'indirizzo. CloudWatch <https://console.aws.amazon.com/cloudwatch/>
2. Nel pannello di navigazione, espandi Logs (Log) e scegli Log groups (Gruppi di log).
3. Scegli il gruppo di log `multiline-test/applicatio`.
4. Scegli il log. Visualizza i messaggi. Le righe che corrispondono alle regole nel file parser vengono concatenate e appaiono come un singolo messaggio.

Il seguente frammento di log mostra le righe concatenate in un singolo evento di traccia dello stack Java:

```
{
 "container_id": "xxxxxx",
 "container_name": "app",
 "source": "stdout",
 "log": "Dec 14 06:41:08 Exception in thread \"main\"
java.lang.RuntimeException: Something has gone wrong, aborting!\\n
at com.myproject.module.MyProject.badMethod(MyProject.java:22)\\n at
com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)\\n
at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)\\n
at com.myproject.module.MyProject.someMethod(MyProject.java:10)\\n at
com.myproject.module.MyProject.main(MyProject.java:6)",
```

```
"ecs_cluster": "default",
"ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
"ecs_task_definition": "firelens-example-multiline:2"
}
```

Il seguente frammento di log mostra come viene visualizzato lo stesso messaggio con una singola riga se esegui un container Amazon ECS non configurato per concatenare messaggi di log multilinea.

```
{
 "log": "Dec 14 06:41:08 Exception in thread \"main\"
java.lang.RuntimeException: Something has gone wrong, aborting!",
 "container_id": "xxxxxx-xxxxxx",
 "container_name": "app",
 "source": "stdout",
 "ecs_cluster": "default",
 "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
 "ecs_task_definition": "firelens-example-multiline:3"
}
```

## Esempio: utilizzo di un parser integrato Fluent Bit

In questo esempio eseguirai i passaggi seguenti:

1. Crea e carica l'immagine per un container Fluent Bit.
2. Crea e carica l'immagine per un'applicazione demo multilinea che esegue, fallisce e genera una traccia dello stack multilinea.
3. Crea la definizione di un processo e avviare il processo.
4. Visualizza i log per verificare che i messaggi che si estendono su più righe appaiano concatenati.

### Crea e carica l'immagine per un container Fluent Bit

Questa immagine includerà un file di configurazione che fa riferimento al parser Fluent Bit.

1. Crea una cartella con il nome `FluentBitDockerImage`.
2. All'interno della cartella `FluentBitDockerImage`, crea un file di configurazione personalizzato che fa riferimento al file parser integrato Fluent Bit.

Per ulteriori informazioni sul file di configurazione personalizzato, consulta [Specifica di un file di configurazione personalizzato](#) nella Guida per gli sviluppatori di Amazon Elastic Container Service

- a. Incolla i contenuti seguenti nel file:

```
[FILTER]
 name multiline
 match *
 multiline.key_content log
 multiline.parser go
```

- b. Salva il file con nome `extra.conf`.
3. All'interno della cartella `FluentBitDockerImage`, crea il Dockerfile con l'immagine Fluent Bit e il parser e i file di configurazione creati.

- a. Incolla i contenuti seguenti nel file:

```
FROM public.ecr.aws/aws-observability/aws-for-fluent-bit:latest
ADD extra.conf /extra.conf
```

- b. Salva il file con nome `Dockerfile`.
4. Utilizzando il Dockerfile, crea un'immagine Fluent Bit personalizzata con il file di configurazione personalizzato inclusi.

#### Note

È possibile posizionare il file di configurazione in qualsiasi punto dell'immagine Docker, a meno che `/fluent-bit/etc/fluent-bit.conf` questo percorso del file non venga utilizzato da FireLens.

- a. Crea l'immagine: `docker build -t fluent-bit-multiline-image .`

Dove: `fluent-bit-multiline-image` è il nome dell'immagine in questo esempio.

- b. Verifica che l'immagine sia stata creata correttamente: `docker images --filter reference=fluent-bit-multiline-image`

In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

## 5. Carica l'immagine Fluent Bit personalizzata in Amazon Elastic Container Registry.

- a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name fluent-bit-multiline-repo --region us-east-1`

Dove: `fluent-bit-multiline-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

L'output fornisce i dettagli del nuovo repository.

- b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag fluent-bit-multiline-image repositoryUri`

Esempio: `docker tag fluent-bit-multiline-image  
xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-repo`

- c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images --filter reference=repositoryUri`

Nell'output, il nome del repository cambia da `fluent-bit-multiline-repo` a `repositoryUri`

- d. Effettua l'autenticazione su Amazon ECR eseguendo il comando `aws ecr get-login-password` e specificando l'ID del log in cui desideri eseguire l'autenticazione: `aws ecr get-login-password | docker login --username AWS --password-stdin registry ID.dkr.ecr.region.amazonaws.com`

Esempio: `aws ecr get-login-password | docker login --username AWS --password-stdin  
xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com`

Viene visualizzato un messaggio di accesso riuscito.

- e. Invia l'immagine ad Amazon ECR: `docker push registry ID.dkr.ecr.region.amazonaws.com/repository name`

Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/  
fluent-bit-multiline-repo`

## Crea e carica l'immagine per un'applicazione demo multilinea

Questa immagine includerà un file di script Python che esegue l'applicazione e un file di log di esempio.

1. Crea una cartella denominata `multiline-app`: `mkdir multiline-app`
2. Crea un file di script Python.
  - a. Nella cartella `multiline-app`, crea un file con il nome `main.py`.
  - b. Incolla i contenuti seguenti nel file:

```
import os
import time
file1 = open('/test.log', 'r')
Lines = file1.readlines()

count = 0

for i in range(10):
 print("app running normally...")
 time.sleep(1)

Strips the newline character
for line in Lines:
 count += 1
 print(line.rstrip())
print(count)
print("app terminated.")
```

- c. Salvare il file `main.py`.
3. Crea un file di log di esempio.
  - a. Nella cartella `multiline-app`, crea un file con il nome `test.log`.
  - b. Incolla i contenuti seguenti nel file:

```
panic: my panic

goroutine 4 [running]:
panic(0x45cb40, 0x47ad70)
```

```

 /usr/local/go/src/runtime/panic.go:542 +0x46c fp=0xc42003f7b8 sp=0xc42003f710
 pc=0x422f7c
main.main.func1(0xc420024120)
 foo.go:6 +0x39 fp=0xc42003f7d8 sp=0xc42003f7b8 pc=0x451339
runtime.goexit()
 /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003f7e0
 sp=0xc42003f7d8 pc=0x44b4d1
created by main.main
 foo.go:5 +0x58

goroutine 1 [chan receive]:
runtime.gopark(0x4739b8, 0xc420024178, 0x46fcd7, 0xc, 0xc420028e17, 0x3)
 /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc420053e30 sp=0xc420053e00
 pc=0x42503c
runtime.goparkunlock(0xc420024178, 0x46fcd7, 0xc, 0x1000f010040c217, 0x3)
 /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc420053e70 sp=0xc420053e30
 pc=0x42512e
runtime.chanrecv(0xc420024120, 0x0, 0xc420053f01, 0x4512d8)
 /usr/local/go/src/runtime/chan.go:506 +0x304 fp=0xc420053f20 sp=0xc420053e70
 pc=0x4046b4
runtime.chanrecv1(0xc420024120, 0x0)
 /usr/local/go/src/runtime/chan.go:388 +0x2b fp=0xc420053f50 sp=0xc420053f20
 pc=0x40439b
main.main()
 foo.go:9 +0x6f fp=0xc420053f80 sp=0xc420053f50 pc=0x4512ef
runtime.main()
 /usr/local/go/src/runtime/proc.go:185 +0x20d fp=0xc420053fe0 sp=0xc420053f80
 pc=0x424bad
runtime.goexit()
 /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc420053fe8
 sp=0xc420053fe0 pc=0x44b4d1

goroutine 2 [force gc (idle)]:
runtime.gopark(0x4739b8, 0x4ad720, 0x47001e, 0xf, 0x14, 0x1)
 /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003e768 sp=0xc42003e738
 pc=0x42503c
runtime.goparkunlock(0x4ad720, 0x47001e, 0xf, 0xc420000114, 0x1)
 /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003e7a8 sp=0xc42003e768
 pc=0x42512e
runtime.forcegchelper()
 /usr/local/go/src/runtime/proc.go:238 +0xcc fp=0xc42003e7e0 sp=0xc42003e7a8
 pc=0x424e5c
runtime.goexit()

```



```

/usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003e7e8
sp=0xc42003e7e0 pc=0x44b4d1
created by runtime.init.4
/usr/local/go/src/runtime/proc.go:227 +0x35

goroutine 3 [GC sweep wait]:
runtime.gopark(0x4739b8, 0x4ad7e0, 0x46fdd2, 0xd, 0x419914, 0x1)
/usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003ef60 sp=0xc42003ef30
pc=0x42503c
runtime.goparkunlock(0x4ad7e0, 0x46fdd2, 0xd, 0x14, 0x1)
/usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003efa0 sp=0xc42003ef60
pc=0x42512e
runtime.bgsweep(0xc42001e150)
/usr/local/go/src/runtime/mgcsweep.go:52 +0xa3 fp=0xc42003efd8
sp=0xc42003efa0 pc=0x419973
runtime.goexit()
/usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003efe0
sp=0xc42003efd8 pc=0x44b4d1
created by runtime.gcenable
/usr/local/go/src/runtime/mgc.go:216 +0x58
one more line, no multiline

```

c. Salvare il file `test.log`.

4. All'interno della cartella `multiline-app`, crea il `Dockerfile`.

a. Incolla i contenuti seguenti nel file:

```

FROM public.ecr.aws/amazonlinux/amazonlinux:latest
ADD test.log /test.log

RUN yum upgrade -y && yum install -y python3

WORKDIR /usr/local/bin

COPY main.py .

CMD ["python3", "main.py"]

```

b. Salvare il file `Dockerfile`.

5. Usando il `Dockerfile`, crea un'immagine.

a. Crea l'immagine: `docker build -t multiline-app-image .`

Dove: `multiline-app-image` è il nome dell'immagine in questo esempio.

- b. Verifica che l'immagine sia stata creata correttamente: `docker images --filter reference=multiline-app-image`

In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

## 6. Carica l'immagine nel registro del container Amazon Elastic.

- a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name multiline-app-repo --region us-east-1`

Dove: `multiline-app-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

L'output fornisce i dettagli del nuovo repository. Annota il valore `repositoryUri` poiché sarà necessario nei passaggi successivi.

- b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag multiline-app-image repositoryUri`

Esempio: `docker tag multiline-app-image xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

- c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images --filter reference=repositoryUri`

Nell'output, il nome del repository cambia da `multiline-app-repo` al valore `repositoryUri`.

- d. Invia l'immagine ad Amazon ECR: `docker push aws_account_id.dkr.ecr.region.amazonaws.com/repository name`

Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

## Crea la definizione di un processo e avviare il processo

1. Crea un file di definizione del processo con il nome del file `multiline-task-definition.json`.
2. Incolla i contenuti seguenti nel file `multiline-task-definition.json`:

```

{
 "family": "firelens-example-multiline",
 "taskRoleArn": "task role ARN",
 "executionRoleArn": "execution role ARN",
 "containerDefinitions": [
 {
 "essential": true,
 "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-image:latest",
 "name": "log_router",
 "firelensConfiguration": {
 "type": "fluentbit",
 "options": {
 "config-file-type": "file",
 "config-file-value": "/extra.conf"
 }
 },
 "memoryReservation": 50
 },
 {
 "essential": true,
 "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/multiline-app-image:latest",
 "name": "app",
 "logConfiguration": {
 "logDriver": "awsfirelens",
 "options": {
 "Name": "cloudwatch_logs",
 "region": "us-east-1",
 "log_group_name": "multiline-test/application",
 "auto_create_group": "true",
 "log_stream_prefix": "multiline-"
 }
 },
 "memoryReservation": 100
 }
],
 "requiresCompatibilities": ["FARGATE"],
 "networkMode": "awsvpc",
 "cpu": "256",
 "memory": "512"
}

```

Sostituisci quanto segue nella definizione del processo `multiline-task-definition.json`:

a. *task role ARN*

Per trovare l'ARN del ruolo del processo, vai alla Console IAM. Scegli Roles (Roli) e trova il ruolo del processo `ecs-task-role-for-firelens` che hai creato. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

b. *execution role ARN*

Per trovare l'ARN del ruolo di esecuzione, vai alla Console IAM. Scegli Roles (Ruoli) e trova il ruolo `ecsTaskExecutionRole`. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

c. *aws\_account\_id*

Per trovare l'`aws_account_id`, accedi alla AWS Management Console. Scegli il tuo nome utente in alto a destra e copia il tuo ID account.

d. *us-east-1*

Sostituisci la regione se necessario.

3. Registra il file di definizione del processo: `aws ecs register-task-definition --cli-input-json file://multiline-task-definition.json --region us-east-1`
4. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
5. Nel pannello di navigazione, scegli Task Definitions (Definizioni dei processi) quindi scegli la famiglia `firelens-example-multiline`, perché sopra abbiamo registrato la definizione dell'attività processo in questa famiglia nella prima riga della definizione del processo.
6. Scegli la versione più recente.
7. Scegli Distribuisci, Esegui attività.
8. Nella pagina Esegui attività, in Cluster scegli il cluster, quindi in Reti scegli le sottoreti disponibili per tale attività in Sottoreti.
9. Scegli Create (Crea).

Verifica che i messaggi di log multilinea in Amazon CloudWatch appaiano concatenati

1. Apri la console all'indirizzo. CloudWatch <https://console.aws.amazon.com/cloudwatch/>
2. Nel pannello di navigazione, espandi Logs (Log) e scegli Log groups (Gruppi di log).

3. Scegli il gruppo di log multiline-test/applicatio.
4. Scegli il log e visualizza i messaggi. Le righe che corrispondono alle regole nel file parser vengono concatenate e appaiono come un singolo messaggio.

Il seguente snippet di log mostra una traccia dello stack Go concatenata in un singolo evento:

```
{
 "log": "panic: my panic\n\nngoroutine 4 [running]:\npanic(0x45cb40,\n0x47ad70)\n /usr/local/go/src/runtime/panic.go:542 +0x46c fp=0xc42003f7b8\nsp=0xc42003f710 pc=0x422f7c\nmain.main.func1(0xc420024120)\n foo.go:6\n+0x39 fp=0xc42003f7d8 sp=0xc42003f7b8 pc=0x451339\nruntime.goexit()\n /usr/\nlocal/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003f7e0 sp=0xc42003f7d8\npc=0x44b4d1\ncreated by main.main\n foo.go:5 +0x58\n\nngoroutine 1 [chan receive]:\nruntime.gopark(0x4739b8, 0xc420024178, 0x46fcd7, 0xc, 0xc420028e17, 0x3)\n /usr/\nlocal/go/src/runtime/proc.go:280 +0x12c fp=0xc420053e30 sp=0xc420053e00 pc=0x42503c\nruntime.goparkunlock(0xc420024178, 0x46fcd7, 0xc, 0x1000f010040c217, 0x3)\n\n /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc420053e70 sp=0xc420053e30\npc=0x42512e\nruntime.chanrecv(0xc420024120, 0x0, 0xc420053f01, 0x4512d8)\n\n /usr/local/go/src/runtime/chan.go:506 +0x304 fp=0xc420053f20 sp=0xc420053e70\npc=0x4046b4\nruntime.chanrecv1(0xc420024120, 0x0)\n /usr/local/go/src/runtime/\nchan.go:388 +0x2b fp=0xc420053f50 sp=0xc420053f20 pc=0x40439b\nmain.main()\n\n foo.go:9 +0x6f fp=0xc420053f80 sp=0xc420053f50 pc=0x4512ef\nruntime.main()\n\n /usr/local/go/src/runtime/proc.go:185 +0x20d fp=0xc420053fe0 sp=0xc420053f80\npc=0x424bad\nruntime.goexit()\n /usr/local/go/src/runtime/asm_amd64.s:2337\n+0x1 fp=0xc420053fe8 sp=0xc420053fe0 pc=0x44b4d1\n\nngoroutine 2 [force gc\n(idle)]:\nruntime.gopark(0x4739b8, 0x4ad720, 0x47001e, 0xf, 0x14, 0x1)\n /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003e768 sp=0xc42003e738\npc=0x42503c\nruntime.goparkunlock(0x4ad720, 0x47001e, 0xf, 0xc420000114, 0x1)\n\n /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003e7a8 sp=0xc42003e768\npc=0x42512e\nruntime.forcegchelper()\n /usr/local/go/src/runtime/proc.go:238\n+0xcc fp=0xc42003e7e0 sp=0xc42003e7a8 pc=0x424e5c\nruntime.goexit()\n /usr/\nlocal/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003e7e8 sp=0xc42003e7e0\npc=0x44b4d1\ncreated by runtime.init.4\n /usr/local/go/src/runtime/proc.go:227\n+0x35\n\nngoroutine 3 [GC sweep wait]:\nruntime.gopark(0x4739b8, 0x4ad7e0,\n0x46fdd2, 0xd, 0x419914, 0x1)\n /usr/local/go/src/runtime/proc.go:280 +0x12c\nfp=0xc42003ef60 sp=0xc42003ef30 pc=0x42503c\nruntime.goparkunlock(0x4ad7e0,\n0x46fdd2, 0xd, 0x14, 0x1)\n /usr/local/go/src/runtime/proc.go:286 +0x5e\nfp=0xc42003efa0 sp=0xc42003ef60 pc=0x42512e\nruntime.bgsweep(0xc42001e150)\n\n /usr/local/go/src/runtime/mgcsweep.go:52 +0xa3 fp=0xc42003efd8 sp=0xc42003efa0\npc=0x419973\nruntime.goexit()\n /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1\nfp=0xc42003efe0 sp=0xc42003efd8 pc=0x44b4d1\ncreated by runtime.gcenable\n /usr/\nlocal/go/src/runtime/mgc.go:216 +0x58",
 "container_id": "xxxxxx-xxxxxx",
```

```
"container_name": "app",
"source": "stdout",
"ecs_cluster": "default",
"ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
"ecs_task_definition": "firelens-example-multiline:2"
}
```

Il seguente snippet di log mostra come viene visualizzato lo stesso evento se si esegue un container ECS non configurato per concatenare messaggi di log multilinea. Il campo di log contiene una singola riga.

```
{
 "log": "panic: my panic",
 "container_id": "xxxxxx-xxxxxx",
 "container_name": "app",
 "source": "stdout",
 "ecs_cluster": "default",
 "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
 "ecs_task_definition": "firelens-example-multiline:3"
}
```

#### Note

Se i tuoi log passano ai file di log anziché all'output standard, ti consigliamo di specificare i parametri di configurazione `multiline.parser` e `multiline.key_content` nel [plug-in Tail input](#) anziché nel filtro.

## Distribuzione di Fluent Bit su contenitori Amazon ECS Windows

Fluent Bit è un processore e router di log veloce e flessibile supportato da vari sistemi operativi. Può essere utilizzato per indirizzare i log verso varie AWS destinazioni come Amazon CloudWatch Logs, Firehose, Amazon S3 e Amazon Service. OpenSearch Fluent Bit supporta soluzioni partner comuni come [Datadog](#), [Splunk](#) e server HTTP personalizzati. Per ulteriori informazioni su Fluent Bit, consulta il [Fluent Bit](#) sito web.

L'immagine AWS for Fluent Bit è disponibile su Amazon ECR sia nella galleria pubblica Amazon ECR che in un repository Amazon ECR nella maggior parte delle regioni per la disponibilità elevata. Per ulteriori informazioni, consulta [aws-for-fluent-bit](#) sul sito GitHub web.

Questo tutorial illustra come distribuire i contenitori Fluent Bit sulle relative istanze Windows in esecuzione in Amazon ECS per trasmettere i log generati dalle attività di Windows ad Amazon per la registrazione centralizzata. CloudWatch

Questo tutorial utilizza il seguente approccio:

- Fluent Bit funziona come servizio con la strategia di pianificazione Daemon. Questa strategia garantisce che una singola istanza di Fluent Bit venga sempre eseguita sulle istanze di container del cluster.
- Ascolta sulla porta 24224 utilizzando il plug-in di input di inoltro.
- Aprire la porta 24224 all'host in modo che il runtime docker possa inviare i log a Fluent Bit utilizzando tale porta.
- Ha una configurazione che consente a Fluent Bit di inviare i record dei log alle destinazioni specificate.
- Avvia tutti gli altri container di attività Amazon ECS utilizzando il driver di registrazione fluentd. Per ulteriori informazioni, consulta [Driver di registrazione Fluentd](#) sul sito Web della documentazione Docker.
- Docker si connette al socket TCP 24224 su localhost all'interno dello spazio dei nomi dell'host.
- L'agente Amazon ECS aggiunge le etichette ai container che includono il nome del cluster, l'ARN dell'attività, il nome della famiglia della definizione di attività, il numero di revisione della definizione di attività, l'ARN dell'attività e il nome del container delle attività. Le stesse informazioni vengono aggiunte al record di log utilizzando l'opzione labels del driver di registrazione fluentd docker. Per ulteriori informazioni, consulta [labels, labels-regex, env ed env-regex](#) sul sito Web contenente la documentazione di Docker.
- Poiché l'opzione async del driver di registrazione fluentd è impostata su true, quando il container Fluent Bit viene riavviato, docker memorizza i log nel buffer fino al riavvio del container Fluent Bit. Puoi aumentare il limite del buffer impostando l'opzione. fluentd-buffer-limit Per ulteriori informazioni, consulta il sito [fluentd-buffer-limit](#) Web della documentazione di Docker.

Di seguito è riportato il flusso di lavoro:

- Il container di Fluent Bit si avvia e ascolta sulla porta 24224 che è esposta all'host.
- Fluent Bit utilizza le credenziali del ruolo IAM dell'attività specificate nella definizione di attività.
- Altre attività avviate sulla stessa istanza utilizzano il driver di registrazione fluentd docker per connettersi al container di Fluent Bit sulla porta 24224.

- Quando i container dell'applicazione generano log, il runtime docker aggiunge un tag a tali record, aggiunge altri metadati specificati nelle etichette e quindi li inoltra sulla porta 24224 dello spazio dei nomi dell'host.
- Fluent Bit riceve il record di log sulla porta 24224 perché è esposto allo spazio dei nomi dell'host.
- Fluent Bit esegue l'elaborazione interna e indirizza i log come specificato.

Questo tutorial utilizza la configurazione CloudWatch Fluent Bit predefinita che esegue le seguenti operazioni:

- Crea un nuovo gruppo di log per ogni cluster e famiglia di definizione di attività.
- Crea un nuovo flusso di log per ogni container di attività nel gruppo di log sopra generato ogni volta che viene avviata una nuova attività. Ogni flusso verrà contrassegnato con l'ID dell'attività a cui appartiene il container.
- Aggiunge altri metadati tra cui il nome del cluster, l'ARN dell'attività, il nome del container di attività, la famiglia di definizioni di attività e il numero di revisione della definizione di attività in ogni voce del log.

Ad esempio, se hai `task_1` with `container_1` `container_2` e `task_2` with `container_3`, i seguenti sono i flussi di CloudWatch log:

- `/aws/ecs/windows.ecs_task_1`  
`task-out.TASK_ID.container_1`  
`task-out.TASK_ID.container_2`
- `/aws/ecs/windows.ecs_task_2`  
`task-out.TASK_ID.container_3`

#### Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

## Fasi



- [Prerequisiti](#)
- [Fase 1: Creazione dei ruoli di accesso IAM](#)
- [Fase 2: Creazione di un'istanza di container Windows di Amazon ECS](#)
- [Fase 3: Configurazione di Fluent Bit](#)
- [Passaggio 4: Registrare una definizione di attività Windows Fluent Bit che indirizza i log a CloudWatch](#)
- [Fase 5: Esecuzione della definizione di attività ecs-windows-fluent-bit come servizio Amazon ECS utilizzando la strategia di pianificazione del daemon](#)
- [Fase 6: Registrazione di una definizione di attività di Windows che genera i log](#)
- [Fase 7: Esecuzione della definizione di attività windows-app-task](#)
- [Fase 8: Verificare gli accessi CloudWatch](#)
- [Fase 9: Pulizia](#)

## Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni, vedere [Installazione o aggiornamento alla versione più recente di AWS CLI](#).
- L'immagine del container `aws-for-fluent-bit` è disponibile per i seguenti sistemi operativi Windows:
  - Windows Server, versione 2019 (Core)
  - Windows Server, versione 2019 (Full)
  - Windows Server 2022 Core
  - Windows Server 2022 Full
- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Hai un cluster. In questo tutorial, il nome del cluster è `FluentBit-cluster`.
- Hai un VPC con una sottorete pubblica in cui verrà avviata l' EC2 istanza. È possibile utilizzare il VPC di default. Puoi anche utilizzare una sottorete privata che consente agli CloudWatch endpoint Amazon di raggiungere la sottorete. Per ulteriori informazioni sugli CloudWatch endpoint Amazon, consulta la sezione [CloudWatch Endpoints e quote Amazon](#) nel. Riferimenti generali di AWS Per ulteriori informazioni sull'utilizzo della procedura guidata di Amazon VPC per creare un VPC, consulta [the section called "Crea un cloud privato virtuale"](#).

## Fase 1: Creazione dei ruoli di accesso IAM

Crea i ruoli IAM di Amazon ECS.

1. Crea il ruolo dell'istanza del contenitore Amazon ECS denominato "ecsInstanceRole". Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container di Amazon ECS](#).
2. Crea un ruolo IAM per l'attività Fluent Bit denominata fluentTaskRole. Per ulteriori informazioni, consulta [the section called "Ruolo IAM del processo"](#).

Le autorizzazioni IAM concesse nel ruolo IAM sono assunte dai container delle attività. Per consentire a Fluent Bit di inviare i log a CloudWatch, devi assegnare le seguenti autorizzazioni al ruolo IAM dell'attività.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "logs:CreateLogStream",
 "logs:CreateLogGroup",
 "logs:DescribeLogStreams",
 "logs:PutLogEvents"
],
 "Resource": "*"
 }
]
}
```

3. Collegare la policy al ruolo.
  - a. Salva il contenuto di cui sopra in un file denominato fluent-bit-policy.json.
  - b. Per collegare la policy in linea al ruolo IAM fluentTaskRole, esegui il comando riportato.

```
aws iam put-role-policy --role-name fluentTaskRole --policy-name
fluentTaskPolicy --policy-document file://fluent-bit-policy.json
```

## Fase 2: Creazione di un'istanza di container Windows di Amazon ECS

Creazione di un'istanza di container Windows di Amazon ECS

## Creazione di un'istanza Amazon ECS

1. Usa il comando `aws ssm get-parameters` per recuperare l'ID AMI per la regione che ospita il tuo VPC. Per ulteriori informazioni, consulta [Recupero dei metadati dell'AMI ottimizzata per Amazon ECS](#).
2. Usa la EC2 console Amazon per avviare l'istanza.
  - a. Apri la EC2 console Amazon all'indirizzo <https://console.aws.amazon.com/ec2/>.
  - b. Seleziona la Regione da utilizzare nella barra di navigazione.
  - c. Dalla EC2 dashboard, scegli Launch instance.
  - d. Per Name (Nome), inserisci un nome univoco.
  - e. Per Application and OS Images (Amazon Machine Image) (Immagini di applicazioni e sistema operativo [Amazon Machine Image]), scegli l'AMI che è stata recuperata nel primo passaggio.
  - f. In Instance type (Tipo di istanza) selezionare `t3.xlarge`.
  - g. Per Key pair (login) (Coppia di chiavi [accesso]), scegli una coppia di chiavi.
  - h. In Network settings (Impostazioni di rete), per Security group (Gruppo di sicurezza), scegli un gruppo di sicurezza esistente o creane uno nuovo.
  - i. In Network settings (Impostazioni di rete), per Auto-assign Public IP (Assegna automaticamente un IP pubblico), seleziona Enable (Abilita).
  - j. In Dettagli avanzati, per il profilo dell'istanza IAM, scegli `ecsInstanceRole`.
  - k. Configura la tua istanza di container Amazon ECS con i seguenti dati utente. In Dettagli avanzati, incolla lo script seguente nel campo Dati utente, sostituendolo `cluster-name` con il nome del cluster.

```
<powershell>
Import-Module ECSTools
Initialize-ECSAgent -Cluster cluster-name -EnableTaskENI -EnableTaskIAMRole -
LoggingDrivers '["awslogs","fluentd"]'
</powershell>
```

- l. Quando sei pronto, seleziona il campo di conferma e scegli Launch Instances (Avvia istanze).
  - m. Una pagina di conferma indicherà che l'istanza si sta avviando. Scegliere View Instances (Visualizza istanze) per chiudere la pagina di conferma e tornare alla console.

## Fase 3: Configurazione di Fluent Bit

È possibile utilizzare la seguente configurazione predefinita fornita da AWS per iniziare rapidamente:

- [Amazon CloudWatch](#) che si basa sul plug-in Fluent Bit per [Amazon CloudWatch](#) nel Manuale ufficiale Fluent Bit.

In alternativa, puoi utilizzare altre configurazioni predefinite fornite da AWS. Per ulteriori informazioni, consulta [Sostituzione del punto di ingresso per l'immagine Windows](#) su `aws-for-fluent-bit` sul sito Web di Github.

La configurazione predefinita di Amazon CloudWatch Fluent Bit è mostrata di seguito.

Sostituisci le seguenti variabili:

- *region* con la regione in cui desideri inviare i CloudWatch log di Amazon.

```
[SERVICE]
 Flush 5
 Log_Level info
 Daemon off

[INPUT]
 Name forward
 Listen 0.0.0.0
 Port 24224
 Buffer_Chunk_Size 1M
 Buffer_Max_Size 6M
 Tag_Prefix ecs.

Amazon ECS agent adds the following log keys as labels to the docker container.
We would use fluentd logging driver to add these to log record while sending it to
Fluent Bit.

[FILTER]
 Name modify
 Match ecs.*
 Rename com.amazonaws.ecs.cluster ecs_cluster
 Rename com.amazonaws.ecs.container-name ecs_container_name
 Rename com.amazonaws.ecs.task-arn ecs_task_arn
 Rename com.amazonaws.ecs.task-definition-family
 ecs_task_definition_family
```

```

 Rename com.amazonaws.ecs.task-definition-version
ecs_task_definition_version

[FILTER]
 Name rewrite_tag
 Match ecs.*
 Rule $ecs_task_arn ^([a-z-:0-9]+)/([a-zA-Z0-9-_]+)/([a-z0-9-9])$
 out.$3.$ecs_container_name false
 Emitter_Name re_emitted

[OUTPUT]
 Name cloudwatch_logs
 Match out.*
 region region
 log_group_name fallback-group
 log_group_template /aws/ecs/$ecs_cluster.$ecs_task_definition_family
 log_stream_prefix task-
 auto_create_group 0n

```

Ogni log che entra in Fluent Bit ha un tag specificato dall'utente o viene generato automaticamente quando non ne viene fornito uno. I tag possono essere utilizzati per indirizzare log diversi verso destinazioni diverse. Per ulteriori informazioni, consulta [Tag](#) nel manuale ufficiale di Fluent Bit.

La configurazione di Fluent Bit sopra descritta ha le seguenti proprietà:

- Il plug-in di input di inoltro ascolta il traffico in entrata sulla porta TCP 24224.
- Ogni voce di log ricevuta su quella porta ha un tag che il plug-in di input di inoltro modifica per aggiungere una stringa `ecs.` al record come prefisso.
- La pipeline interna di Fluent Bit indirizza la voce del log per modificare il filtro utilizzando l'espressione regolare Match. Questo filtro sostituisce le chiavi del record di log JSON nel formato che Fluent Bit può utilizzare.
- La voce di log modificata viene quindi utilizzata dal filtro `rewrite_tag`. Questo filtro modifica il tag del record di registro nel formato `out. TASK_ID. CONTAINER_NAME`.
- Il nuovo tag verrà indirizzato al plug-in `cloudwatch_logs` di output che crea i gruppi e i flussi di log come descritto in precedenza utilizzando le opzioni e del `log_group_template` plug-in di output. `log_stream_prefix` CloudWatch Per ulteriori informazioni, consulta [Configuration parameters](#) (Parametri di configurazione) nel manuale ufficiale di Fluent Bit.

## Passaggio 4: Registrare una definizione di attività Windows Fluent Bit che indirizza i log a CloudWatch

Registrare una definizione di attività Windows Fluent Bit a cui indirizzare i log. CloudWatch

### Note

Questa definizione di attività espone la porta del container 24224 di Fluent Bit alla porta host 24224. Verifica che questa porta non sia aperta nel gruppo di sicurezza dell' EC2 istanza per impedire l'accesso dall'esterno.

Come registrare una definizione di attività

1. Crea un file denominato `fluent-bit.json` con i seguenti contenuti.

Sostituisci le seguenti variabili:

- *task-iam-role* con l'Amazon Resource Name (ARN) del ruolo IAM della tua attività
- *region* con la regione in cui viene eseguita la tua attività

```
{
 "family": "ecs-windows-fluent-bit",
 "taskRoleArn": "task-iam-role",
 "containerDefinitions": [
 {
 "name": "fluent-bit",
 "image": "public.ecr.aws/aws-observability/aws-for-fluent-bit:windowsservercore-latest",
 "cpu": 512,
 "portMappings": [
 {
 "hostPort": 24224,
 "containerPort": 24224,
 "protocol": "tcp"
 }
],
 "entryPoint": [
 "Powershell",
 "-Command"
```

```
],
 "command": [
 "C:\\\\entrypoint.ps1 -ConfigFile C:\\\\ecs_windows_forward_daemon\\
\\cloudwatch.conf"
],
 "environment": [
 {
 "name": "AWS_REGION",
 "value": "region"
 }
],
 "memory": 512,
 "essential": true,
 "logConfiguration": {
 "logDriver": "awslogs",
 "options": {
 "awslogs-group": "/ecs/fluent-bit-logs",
 "awslogs-region": "region",
 "awslogs-stream-prefix": "flb",
 "awslogs-create-group": "true"
 }
 }
}
],
"memory": "512",
"cpu": "512"
}
```

2. Utilizza il comando seguente per registrare la definizione dell'attività.

```
aws ecs register-task-definition --cli-input-json file://fluent-bit.json --
region region
```

Puoi sempre ottenere un elenco delle definizioni di attività per l'account tramite il comando `list-task-definitions`. L'output che mostra i valori della famiglia e delle revisioni che è possibile utilizzare insieme a `run-task` o `start-task`.

## Fase 5: Esecuzione della definizione di attività **ecs-windows-fluent-bit** come servizio Amazon ECS utilizzando la strategia di pianificazione del daemon

Dopo aver registrato una definizione di attività per l'account, puoi eseguire un'attività nel cluster. Per questo tutorial, esegui una istanza della definizione di attività `ecs-windows-fluent-bit:1` nel cluster `FluentBit-cluster`. Esegui l'attività in un servizio che utilizza la strategia di pianificazione dei daemon, che garantisce che una singola istanza di Fluent Bit venga sempre eseguita su ciascuna delle istanze di container.

Per eseguire un'attività

1. Esegui il comando seguente per avviare la definizione di attività `ecs-windows-fluent-bit:1` (registrata nel passaggio precedente) come servizio.

### Note

Questa definizione di attività utilizza il driver di registrazione `awslogs`; l'istanza di container deve disporre delle autorizzazioni necessarie.

Sostituisci le seguenti variabili:

- **region** con la regione in cui viene eseguito il servizio

```
aws ecs create-service \
 --cluster FluentBit-cluster \
 --service-name FluentBitForwardDaemonService \
 --task-definition ecs-windows-fluent-bit:1 \
 --launch-type EC2 \
 --scheduling-strategy DAEMON \
 --region region
```

2. Per elencare le attività, esegui il comando riportato.

Sostituisci le seguenti variabili:

- **region** con la regione in cui vengono eseguite le attività di assistenza



```
aws ecs list-tasks --cluster FluentBit-cluster --region region
```

## Fase 6: Registrazione di una definizione di attività di Windows che genera i log

Registrazione di una definizione di attività di Windows che genera i log Questa definizione di attività implementa l'immagine del container di Windows che scriverà un numero incrementale su `stdout` ogni secondo.

La definizione di attività utilizza il driver di registrazione `fluentd` che si collega alla porta 24224 su cui ascolta il plug-in `Fluent Bit`. L'agente Amazon ECS etichetta ogni container Amazon ECS con tag che includono il nome del cluster, l'ARN dell'attività, il nome della famiglia della definizione di attività, il numero di revisione della definizione di attività e il nome del container delle attività. Queste etichette chiave-valore vengono passate a `Fluent Bit`.

### Note

Questa attività utilizza la modalità di rete `default`. Tuttavia, è anche possibile utilizzare la modalità di rete `awsvpc` con l'attività.

Come registrare una definizione di attività

1. Crea un file denominato `windows-app-task.json` con i seguenti contenuti.

```
{
 "family": "windows-app-task",
 "containerDefinitions": [
 {
 "name": "sample-container",
 "image": "mcr.microsoft.com/windows/servercore:ltsc2019",
 "cpu": 512,
 "memory": 512,
 "essential": true,
 "entryPoint": [
 "Powershell",
 "-Command"
```

```

],
 "command": [
 "$count=1;while(1) { Write-Host $count; sleep 1; $count=$count+1;}"
],
 "logConfiguration": {
 "logDriver": "fluentd",
 "options": {
 "fluentd-address": "localhost:24224",
 "tag": "{{ index .ContainerLabels \"com.amazonaws.ecs.task-definition-
family\" }}",
 "fluentd-async": "true",
 "labels": "com.amazonaws.ecs.cluster,com.amazonaws.ecs.container-
name,com.amazonaws.ecs.task-arn,com.amazonaws.ecs.task-definition-
family,com.amazonaws.ecs.task-definition-version"
 }
 }
},
"memory": "512",
"cpu": "512"
}

```

2. Utilizza il comando seguente per registrare la definizione dell'attività.

Sostituisci le seguenti variabili:

- **region** con la regione in cui viene eseguita l'attività

```
aws ecs register-task-definition --cli-input-json file://windows-app-task.json --
region region
```

Puoi sempre ottenere un elenco delle definizioni di attività per l'account tramite il comando `list-task-definitions`. L'output che mostra i valori della famiglia e delle revisioni che è possibile utilizzare insieme a `run-task` o `start-task`.

## Fase 7: Esecuzione della definizione di attività **windows-app-task**

Dopo aver registrato la definizione di attività `windows-app-task`, eseguila nel cluster `FluentBit-cluster`.

## Per eseguire un'attività

1. Esegui la definizione di attività `windows-app-task:1` registrata nella fase precedente.

Sostituisci le seguenti variabili:

- `region` con la regione in cui viene eseguita l'attività

```
aws ecs run-task --cluster FluentBit-cluster --task-definition windows-app-task:1
--count 2 --region region
```

2. Per elencare le attività, esegui il comando riportato.

```
aws ecs list-tasks --cluster FluentBit-cluster
```

## Fase 8: Verificare gli accessi CloudWatch

Per verificare la configurazione di Fluent Bit, controlla i seguenti gruppi di log nella CloudWatch console:

- `/ecs/fluent-bit-logs`: questo è il gruppo di log che corrisponde al container del daemon Fluent Bit in esecuzione sull'istanza di container.
- `/aws/ecs/FluentBit-cluster.windows-app-task`: questo è il gruppo di log che corrisponde a tutte le attività avviate per la famiglia di definizione di attività `windows-app-task` all'interno del cluster `FluentBit-cluster`.

`task-out.FIRST_TASK_ID.sample-container`: questo flusso di log contiene tutti i log generati dalla prima istanza dell'attività nel container delle attività `sample-container`.

`task-out.SECOND_TASK_ID.sample-container`: questo flusso di log contiene tutti i log generati dalla seconda istanza dell'attività nel container delle attività `sample-container`.

Il flusso di log `task-out.TASK_ID.sample-container` ha campi simili ai seguenti:

```
{
 "source": "stdout",
 "ecs_task_arn": "arn:aws:ecs:region:0123456789012:task/FluentBit-
cluster/13EXAMPLE",
```

```
"container_name": "/ecs-windows-app-task-1-sample-container-cEXAMPLE",
"ecs_cluster": "FluentBit-cluster",
"ecs_container_name": "sample-container",
"ecs_task_definition_version": "1",
"container_id": "61f5e6EXAMPLE",
"log": "10",
"ecs_task_definition_family": "windows-app-task"
}
```

## Verifica della configurazione di Fluent Bit

1. Apri la CloudWatch console all'indirizzo <https://console.aws.amazon.com/cloudwatch/>.
2. Nel pannello di navigazione, selezionare Log groups (Gruppi di log). Assicurati di trovarti nella regione in cui è stato implementato Fluent Bit sui container.

Nell'elenco dei gruppi di log di Regione AWS, dovresti vedere quanto segue:

- /ecs/fluent-bit-logs
- /aws/ecs/FluentBit-cluster.windows-app-task

Se questi gruppi di log sono visualizzati, la configurazione Fluent Bit è verificata.

## Fase 9: Pulizia

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi.

Per eliminare le risorse del tutorial

1. Interrompi l'attività windows-simple-task e l'attività ecs-fluent-bit. Per ulteriori informazioni, consulta [the section called “Interruzione di un'attività”](#).
2. Esegui il comando riportato per eliminare il gruppo di log /ecs/fluent-bit-logs. Per ulteriori informazioni, sull'eliminazione dei gruppi di log, consulta [delete-log-group](#) la sezione AWS Command Line Interface Reference.

```
aws logs delete-log-group --log-group-name /ecs/fluent-bit-logs
aws logs delete-log-group --log-group-name /aws/ecs/FluentBit-cluster.windows-app-task
```

3. Esegui il comando riportato per terminare l'istanza.

```
aws ec2 terminate-instances --instance-ids instance-id
```

4. Esegui i comandi riportati per eliminare i ruoli IAM.

```
aws iam delete-role --role-name ecsInstanceRole
aws iam delete-role --role-name fluentTaskRole
```

5. Esegui il seguente comando per eliminare il cluster Amazon ECS.

```
aws ecs delete-cluster --cluster FluentBit-cluster
```

## Utilizzo gMSA per EC2 Linux contenitori su Amazon ECS

Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux EC2 tramite un tipo speciale di account di servizio chiamato Account di servizio gestito di gruppo (gMSA).

Linux applicazioni di rete basate, come .NET Le applicazioni principali possono utilizzare Active Directory per facilitare l'autenticazione e la gestione delle autorizzazioni tra utenti e servizi. Puoi utilizzare questa funzionalità progettando applicazioni che si integrano con Active Directory e vengono eseguite su server aggiunti al dominio. Ma perché Linux i contenitori non possono essere aggiunti a un dominio, è necessario configurare un Linux contenitore con cui eseguire gMSA.

A Linux contenitore con cui funziona gMSA si basa sul `credentials-fetcher` demone che viene eseguito sull'istanza Amazon host del contenitore. EC2 Cioè, il demone recupera il gMSA credenziali dal controller di dominio Active Directory e quindi trasferisce queste credenziali all'istanza del contenitore. Per ulteriori informazioni sugli account di servizio, consulta [Create gMSAs per contenitori Windows](#) sul sito Web Microsoft Learn.

## Considerazioni

Considera quanto segue prima dell'uso gMSA for Linux contenitori:

- Se i tuoi contenitori funzionano EC2, puoi usare gMSA for Windows contenitori e Linux contenitori. Per informazioni su come usare gMSA per il contenitore Linux su Fargate, vedere. [Utilizzo gMSA for Linux container su Fargate](#)
- Potresti aver bisogno di un Windows computer aggiunto al dominio per completare i prerequisiti. Ad esempio, potresti aver bisogno di un Windows computer che fa parte del dominio per creare

il gMSA in Active Directory con PowerShell. La RSAT PowerShell Gli strumenti Active Director sono disponibili solo per Windows. Per ulteriori informazioni, vedere [Installazione degli strumenti di amministrazione di Active Directory](#).

- Hai scelto tra senza dominio gMSA e unendo ogni istanza a un singolo dominio. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

Quindi, scegli l'archiviazione dei dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per la modalità domainless gMSA.

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account nel contenitore. Tu generi il CredSpec file e poi lo archivia in uno dei CredSpec opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore. Per utilizzare il metodo domainless, è disponibile una sezione facoltativa in CredSpec il file può specificare le credenziali in uno dei domainless user credentials opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore.

| Posizione di archiviazione                 | Linux                            | Windows                                    |
|--------------------------------------------|----------------------------------|--------------------------------------------|
| Amazon Simple Storage Service              | CredSpec                         | CredSpec                                   |
| AWS Secrets Manager                        | credenziali utente senza dominio | credenziali utente senza dominio           |
| Amazon EC2 Systems Manager Parameter Store | CredSpec                         | CredSpec, credenziali utente senza dominio |
| File locale                                | N/D                              | CredSpec                                   |

## Prerequisiti

Prima di utilizzare il gMSA per la funzionalità dei contenitori Linux con Amazon ECS, assicurati di completare quanto segue:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida [introduttiva a AWS Managed Microsoft AD](#) nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta [AWS Direct Connect](#).
- Ne hai già uno gMSA account in Active Directory. Per ulteriori informazioni, consulta [Utilizzo gMSA per EC2 Linux contenitori su Amazon ECS](#).
- Hai installato e stai eseguendo il daemon `credentials-fetcher` su un'istanza di container Amazon ECS Linux. Inoltre, hai aggiunto un set iniziale di credenziali al daemon `credentials-fetcher` per l'autenticazione con Active Directory.

#### Note

Il daemon `credentials-fetcher` è disponibile solo per Amazon Linux 2023 e Fedora 37 e versioni successive. Il daemon non è disponibile per Amazon Linux 2. Per ulteriori informazioni, consulta [aws/credentials-fetcher on GitHub](#).

- Configuri le credenziali per l'autenticazione del daemon `credentials-fetcher` con Active Directory. Le credenziali devono essere un membro del gruppo di sicurezza di Active Directory che ha accesso a gMSA account. Sono disponibili diverse opzioni in [Decidi se vuoi aggiungere le istanze al dominio o usare domainless gMSA](#).
- Hai aggiunto le autorizzazioni IAM richieste. Le autorizzazioni richieste dipendono dai metodi scelti per le credenziali iniziali e per l'archiviazione della specifica delle credenziali:
  - Se usi `domainless gMSA` per le credenziali iniziali, AWS Secrets Manager sono richieste le autorizzazioni IAM per il ruolo di esecuzione dell'attività.
  - Se memorizzi la specifica delle credenziali in SSM Parameter Store, le autorizzazioni IAM per Amazon EC2 Systems Manager Parameter Store sono necessarie per il ruolo di esecuzione dell'attività.
  - Se archivi la specifica delle credenziali in Amazon S3, le autorizzazioni IAM per Amazon Simple Storage Service sono necessarie per il ruolo di esecuzione delle attività.

# Configurazione gMSA-capace Linux Contenitori su Amazon ECS

## Preparazione dell'infrastruttura

I passaggi seguenti sono considerazioni e configurazioni che vengono eseguite una sola volta. Dopo aver completato questi passaggi, puoi automatizzare la creazione delle istanze di container per riutilizzare questa configurazione.

Decidi come fornire le credenziali iniziali e configura i dati EC2 utente in un modello di EC2 avvio riutilizzabile per installare il demone `credentials-fetcher`

### 1. Decidi se vuoi aggiungere le istanze al dominio o usare domainless gMSA.

- Unisci EC2 le istanze al dominio Active Directory
- Aggiunta delle istanze in base ai dati dell'utente

Aggiungi i passaggi per aggiungere il dominio Active Directory ai dati EC2 utente in un modello di EC2 avvio. Più gruppi di Amazon EC2 Auto Scaling possono utilizzare lo stesso modello di lancio.

Puoi utilizzare questi passaggi [Unendoti a un Active Directory o FreeIPA dominio](#) in Fedora Docs.

- Crea un utente Active Directory per un sistema senza dominio gMSA

Il `credentials-fetcher` demone ha una funzionalità chiamata `domainless gMSA`. Questa funzionalità richiede un dominio, ma non è necessario aggiungere l' EC2istanza al dominio. Utilizzando `domainless gMSA`, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza. Invece, fornisci il nome di un segreto nel AWS Secrets Manager CredSpec file. Il segreto deve contenere un nome utente, una password e un dominio a cui accedere.

Questa funzionalità è supportata e può essere utilizzata con container Linux e Windows.

Questa funzionalità è simile alla gMSA support for non-domain-joined container hostscaratteristica. Per ulteriori informazioni sulla funzionalità di Windows, vedere [gMSA architettura e miglioramenti](#) sul sito Web Microsoft Learn.



- a. Crea un utente nel dominio di Active Directory. L'utente in Active Directory deve disporre dell'autorizzazione per accedere a gMSA account di servizio utilizzati nelle attività.
- b. Crea un account segreto AWS Secrets Manager, dopo aver creato l'utente in Active Directory. Per ulteriori informazioni, consulta [Creare un AWS Secrets Manager segreto](#).
- c. Inserisci il nome utente, la password e il dominio dell'utente nelle coppie chiave-valore JSON denominate rispettivamente username, password e domainName.

```
{"username":"username","password":"passw0rd", "domainName":"example.com"}
```

- d. Aggiungere la configurazione al CredSpec file per l'account del servizio. Il HostAccountConfig aggiuntivo contiene il nome della risorsa Amazon (ARN) del segreto in Secrets Manager.

In Windows, PluginGUID deve corrispondere al GUID nel seguente frammento di codice esemplificativo. In Linux, PluginGUID viene ignorato. Sostituisci MySecret con l'esempio del nome della risorsa Amazon (ARN) del segreto.

```
"ActiveDirectoryConfig": {
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
 "PluginInput": {
 "CredentialArn": "arn:aws:secretsmanager:aws-
region:111122223333:secret:MySecret"
 }
 }
}
```

- e. Il senza dominio gMSA la funzionalità richiede autorizzazioni aggiuntive nel ruolo di esecuzione dell'attività. Segui il passaggio [\(Facoltativo\) senza dominio gMSA Secret](#).

## 2. Configurazione delle istanze e installazione del daemon **credentials-fetcher**

Puoi installare il `credentials-fetcher` demone con uno script di dati utente nel tuo EC2 Launch Template. Gli esempi seguenti mostrano due tipi di dati utente, cloud-config YAML oppure bash sceneggiatura. Questi esempi sono per Amazon Linux 2023 (AL2023). Sostituisci `MyCluster` con il nome del cluster Amazon ECS a cui desideri che si aggiungano queste istanze.

- **cloud-config** YAML

```
Content-Type: text/cloud-config
package_reboot_if_required: true
packages:
 # prerequisites
 - dotnet
 - realmd
 - oddjob
 - oddjob-mkhomedir
 - sssd
 - adcli
 - krb5-workstation
 - samba-common-tools
 # https://github.com/aws/credentials-fetcher gMSA credentials management for
 containers
 - credentials-fetcher
write_files:
configure the ECS Agent to join your cluster.
replace MyCluster with the name of your cluster.
- path: /etc/ecs/ecs.config
 owner: root:root
 permissions: '0644'
 content: |
 ECS_CLUSTER=MyCluster
 ECS_GMSA_SUPPORTED=true
runcmd:
start the credentials-fetcher daemon and if it succeeded, make it start after
every reboot
- "systemctl start credentials-fetcher"
- "systemctl is-active credentials-fetcher && systemctl enable credentials-
fetcher"
```

- **bash script**

Se ti senti più a tuo agio con bash gli script e hanno più variabili su cui scrivere `/etc/ecs/ecs.config`, usa il seguente heredoc formato. Questo formato scrive tutti gli elementi nel file di configurazione, inserendoli tra le righe `cat` ed EOF.

```
#!/usr/bin/env bash
set -euxo pipefail
```

```
prerequisites
timeout 30 dnf install -y dotnet realmd oddjob oddjob-mkhomedir sssd adcli
krb5-workstation samba-common-tools
install https://github.com/aws/credentials-fetcher gMSA credentials
management for containers
timeout 30 dnf install -y credentials-fetcher

start credentials-fetcher
systemctl start credentials-fetcher
systemctl is-active credentials-fetcher && systemctl enable credentials-fetcher

cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_GMSA_SUPPORTED=true
EOF
```

Esistono variabili di configurazione opzionali per il daemon `credentials-fetcher` che puoi impostare in `/etc/ecs/ecs.config`. Consigliamo di impostare le variabili nei dati utente nel blocco YAML o heredoc in modo simile agli esempi precedenti. In questo modo si evitano problemi di configurazione parziale che possono verificarsi quando si modifica un file più volte. Per ulteriori informazioni sulla configurazione dell'agente ECS, consulta [Amazon ECS Container Agent on GitHub](#).

- Facoltativamente, puoi utilizzare la variabile `CREDENTIALS_FETCHER_HOST` se modifichi la configurazione del daemon `credentials-fetcher` per spostare il socket in un'altra posizione.

## Configurazione di autorizzazioni e segreti

Esegui i passaggi seguenti una volta per ogni applicazione e ogni definizione dell'attività. Consigliamo di utilizzare la best practice di concedere il privilegio minimo e limitare le autorizzazioni utilizzate nella policy. In questo modo, ogni attività può leggere solo i segreti di cui ha bisogno.

### 1. (Facoltativo) senza dominio gMSA Secret

Se utilizzi il metodo senza dominio in cui l'istanza non è aggiunta al dominio, segui questo passaggio.

Inoltre, devi aggiungere le autorizzazioni seguenti come policy inline al ruolo di esecuzione dell'attività del ruolo IAM. In questo modo il daemon `credentials-fetcher` accede al segreto

di Secrets Manager. Sostituisci l'esempio MySecret con il nome della risorsa Amazon (ARN) del segreto nell'elenco Resource.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": [
 "arn:aws:ssm:aws-region:111122223333:secret:MySecret"
]
 }
]
}
```

#### Note

Se utilizzi la tua chiave KMS per crittografare il tuo segreto, devi aggiungere le autorizzazioni necessarie a questo ruolo e aggiungere questo ruolo alla politica delle chiavi. AWS KMS

## 2. Decidi se utilizzare SSM Parameter Store o S3 per archiviare il CredSpec

Amazon ECS supporta i seguenti modi per fare riferimento al percorso del file nel campo `credentialSpecs` di una definizione di attività.

Se unisci le istanze a un singolo dominio, utilizza il prefisso `credentialSpec:` all'inizio dell'ARN nella stringa. Se usi `domainless gMSA`, quindi usa `credentialSpecdomainless:`

Per ulteriori informazioni su CredSpec, consulta [File di specifica delle credenziali](#).

- Bucket Amazon S3

Aggiungi le specifiche delle credenziali a un bucket Amazon S3. Quindi, fai riferimento al nome della risorsa Amazon (ARN) del bucket Amazon S3 nel campo `credentialSpecs` della definizione di attività.

```
{
```

```

 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpecDomainless:arn:aws:s3:::#{BucketName}/#{ObjectName}"
],
 ...
 }
],
 ...
 }

```

Devi aggiungere le seguenti autorizzazioni come policy inline al ruolo IAM di esecuzione delle attività Amazon ECS per consentire alle attività l'accesso al bucket Amazon S3.

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor",
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::amzn-s3-demo-bucket",
 "arn:aws:s3:::amzn-s3-demo-bucket/{object}"
]
 }
]
}

```

- Parametro dell'archivio parametri di SSM

Aggiungi la specifica delle credenziali a un parametro SSM Parameter Store. Quindi, fai riferimento al nome della risorsa Amazon (ARN) del parametro SSM Parameter Store nel campo `credentialSpecs` della definizione delle attività.

```
{
 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpecDomainless:arn:aws:ssm:aws-
region:111122223333:parameter/parameter_name"
],
 ...
 }
],
 ...
}
```

Aggiungi le seguenti autorizzazioni come policy inline al ruolo IAM di esecuzione delle attività Amazon ECS per consentire alle attività l'accesso al parametro SSM Parameter Store.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ssm:GetParameters"
],
 "Resource": [
 "arn:aws:ssm:aws-region:111122223333:parameter/parameter_name"
]
 }
]
}
```

## File di specifica delle credenziali

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account al Linux contenitore.

Tu generi il CredSpec e fate riferimento ad esso nel `credentialSpecs` campo della definizione dell'attività. Il CredSpec il file non contiene segreti.

Di seguito è riportato un esempio CredSpec file.

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
 "Sid": "S-1-5-21-2554468230-2647958158-2204241789",
 "MachineAccountName": "WebApp01",
 "Guid": "8665abd4-e947-4dd0-9a51-f8254943c90b",
 "DnsTreeName": "example.com",
 "DnsName": "example.com",
 "NetBiosName": "example"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "WebApp01",
 "Scope": "example.com"
 }
],
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
 "PluginInput": {
 "CredentialArn": "arn:aws:secretsmanager:aws-
region:111122223333:secret:MySecret"
 }
 }
 }
}
```

## Creare un CredSpec

Crei un CredSpec utilizzando il CredSpec PowerShell modulo su a Windows computer che fa parte del dominio. Segui la procedura descritta in [Creare una specifica di credenziali](#) sul Microsoft Scopri il sito web.

# Utilizzo gMSA for Linux container su Fargate

Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux su Fargate tramite un tipo speciale di account di servizio chiamato Account di servizio gestito di gruppo (gMSA).

Linux applicazioni di rete basate, come .NET Le applicazioni principali possono utilizzare Active Directory per facilitare l'autenticazione e la gestione delle autorizzazioni tra utenti e servizi. Puoi utilizzare questa funzionalità progettando applicazioni che si integrano con Active Directory e vengono eseguite su server aggiunti al dominio. Ma perché Linux i contenitori non possono essere aggiunti a un dominio, è necessario configurare un Linux contenitore con cui eseguire gMSA.

## Considerazioni

Considerate quanto segue prima dell'uso gMSA for Linux contenitori su Fargate:

- È necessario utilizzare la versione 1.4 o successiva della piattaforma.
- Potresti aver bisogno di un Windows computer aggiunto al dominio per completare i prerequisiti. Ad esempio, potresti aver bisogno di un Windows computer che fa parte del dominio per creare il gMSA in Active Directory con PowerShell. La RSAT PowerShell Gli strumenti Active Director sono disponibili solo per Windows. Per ulteriori informazioni, vedere [Installazione degli strumenti di amministrazione di Active Directory](#).
- È necessario utilizzare domainless gMSA.

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account nel contenitore. Tu generi il CredSpec file, quindi archivialo in un bucket Amazon S3.

- Un'attività può supportare solo un Active Directory.

## Prerequisiti

Prima di utilizzare il gMSA per la funzionalità dei contenitori Linux con Amazon ECS, assicurati di completare quanto segue:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida [introduttiva a AWS Managed Microsoft AD](#) nella Guida all'AWS Directory Service amministrazione.



- Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta [AWS Direct Connect](#).
- Ne hai già uno gMSA account in Active Directory e un utente che dispone dell'autorizzazione per accedere a gMSA account di servizio. Per ulteriori informazioni, consulta [Crea un utente Active Directory senza dominio gMSA](#).
- Hai un bucket Amazon S3. Per ulteriori informazioni, consulta [Creare un bucket nella Guida](#) per l'utente di Amazon S3.

## Configurazione gMSA-capace Linux Contenitori su Amazon ECS

### Preparazione dell'infrastruttura

I passaggi seguenti sono considerazioni e configurazioni che vengono eseguite una sola volta.

- Crea un utente Active Directory senza dominio gMSA

Quando usi la modalità domainless gMSA, il contenitore non è aggiunto al dominio. Le altre applicazioni eseguite sul contenitore non possono utilizzare le credenziali per accedere al dominio. Le attività che utilizzano un dominio diverso possono essere eseguite sullo stesso contenitore. Fornisci il nome di un segreto AWS Secrets Manager nel CredSpec file. Il segreto deve contenere un nome utente, una password e un dominio a cui accedere.

Questa funzionalità è simile alla gMSA support for non-domain-joined container hostscaratteristica. Per ulteriori informazioni sulla funzionalità di Windows, vedere [gMSA architettura e miglioramenti](#) sul sito Web Microsoft Learn.

- a. Configura un utente nel tuo dominio Active Directory. L'utente di Active Directory deve disporre dell'autorizzazione per accedere a gMSA account di servizio utilizzato nelle attività.
- b. Disponi di un VPC e delle sottoreti in grado di risolvere il nome di dominio Active Directory. Configura il VPC con opzioni DHCP con il nome di dominio che punta al nome del servizio Active Directory. Per informazioni su come configurare le opzioni DHCP per un VPC, [consulta Work with DHCP option](#) sets nella Amazon Virtual Private Cloud User Guide.
- c. Crea un account segreto in. AWS Secrets Manager
- d. Crea il file con le specifiche delle credenziali.

### Configurazione di autorizzazioni e segreti

Esegui i passaggi seguenti una volta per ogni applicazione e ogni definizione di attività. Consigliamo di utilizzare la best practice di concedere il privilegio minimo e limitare le autorizzazioni utilizzate nella policy. In questo modo, ogni attività può leggere solo i segreti di cui ha bisogno.

1. Crea un utente nel dominio di Active Directory. L'utente in Active Directory deve disporre dell'autorizzazione per accedere a gMSA account di servizio utilizzati nelle attività.
2. Dopo aver creato l'utente di Active Directory, crea un account segreto in AWS Secrets Manager. Per ulteriori informazioni, consulta [Creazione di un segreto AWS Secrets Manager](#).
3. Inserisci il nome utente, la password e il dominio dell'utente nelle coppie chiave-valore JSON denominate rispettivamente username, password e domainName.

```
{"username": "username", "password": "passw0rd", "domainName": "example.com"}
```

4. Inoltre, devi aggiungere le autorizzazioni seguenti come policy inline al ruolo di esecuzione dell'attività del ruolo IAM. In questo modo il daemon `credentials-fetcher` accede al segreto di Secrets Manager. Sostituisci l'esempio `MySecret` con il nome della risorsa Amazon (ARN) del segreto nell'elenco `Resource`.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:aws-region:111122223333:secret:MySecret"
]
 }
]
}
```

#### Note

Se utilizzi la tua chiave KMS per crittografare il tuo segreto, devi aggiungere le autorizzazioni necessarie a questo ruolo e aggiungere questo ruolo alla politica delle AWS KMS chiavi.

5. Aggiungi le specifiche delle credenziali a un bucket Amazon S3. Quindi, fai riferimento al nome della risorsa Amazon (ARN) del bucket Amazon S3 nel campo `credentialSpecs` della definizione di attività.

```
{
 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpecDomainless:arn:aws:s3:::${BucketName}/${ObjectName}"
],
 ...
 }
],
 ...
}
```

Devi aggiungere le seguenti autorizzazioni come policy inline al ruolo IAM di esecuzione delle attività Amazon ECS per consentire alle attività l'accesso al bucket Amazon S3.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor",
 "Effect": "Allow",
 "Action": [
 "s3:GetObject",
 "s3:ListObject"
],
 "Resource": [
 "arn:aws:s3:::{bucket_name}",
 "arn:aws:s3:::{bucket_name}/{object}"
]
 }
]
}
```

## File di specifica delle credenziali

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account al Linux contenitore. Tu generi il CredSpec e fate riferimento ad esso nel `credentialSpecs` campo della definizione dell'attività. Il CredSpec il file non contiene segreti.

Di seguito è riportato un esempio CredSpec file.

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
 "Sid": "S-1-5-21-2554468230-2647958158-2204241789",
 "MachineAccountName": "WebApp01",
 "Guid": "8665abd4-e947-4dd0-9a51-f8254943c90b",
 "DnsTreeName": "example.com",
 "DnsName": "example.com",
 "NetBiosName": "example"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "WebApp01",
 "Scope": "example.com"
 }
],
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
 "PluginInput": {
 "CredentialArn": "arn:aws:secretsmanager:aws-
region:111122223333:secret:MySecret"
 }
 }
 }
}
```

Creare un CredSpec e caricandolo su un Amazon S3

Crei un CredSpec utilizzando il CredSpec PowerShell modulo su a Windows computer che fa parte del dominio. Segui la procedura descritta in [Creare una specifica di credenziali](#) sul Microsoft Scopri il sito web.

Dopo aver creato il file delle specifiche delle credenziali, caricalo in un bucket Amazon S3. Copia il CredSpec file nel computer o nell'ambiente in cui si eseguono AWS CLI i comandi.

Esegui il AWS CLI comando seguente per caricare il CredSpec su Amazon S3. Sostituisci *amzn-s3-demo-bucket* con il nome del bucket Amazon S3. Puoi archiviare il file come oggetto in qualsiasi bucket e posizione, ma devi consentire l'accesso a tale bucket e tale posizione nella policy associata al ruolo di esecuzione dell'attività.

Per PowerShell, usa il seguente comando:

```
$ Write-S3Object -BucketName "amzn-s3-demo-bucket" -Key "ecs-domainless-gmsa-credspec" -File "gmsa-cred-spec.json"
```

Il AWS CLI comando seguente utilizza i caratteri di continuazione con barra rovesciata utilizzati dalle shell compatibilish.

```
$ aws s3 cp gmsa-cred-spec.json \
s3://amzn-s3-demo-bucket/ecs-domainless-gmsa-credspec
```

## Utilizzo di contenitori Amazon ECS Windows con sistema domainless gMSA utilizzando il AWS CLI

Il tutorial seguente mostra come creare un'attività Amazon ECS che esegue un container Windows con credenziali per accedere ad Active Directory con la AWS CLI. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

### Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare e configurare gMSA account su Active Directory Domain Services \(AD DS\)](#)
- [Fase 2: caricamento delle credenziali in Secrets Manager](#)

- [Passaggio 3: Modifica il tuo CredSpec JSON da includere senza dominio gMSA Informazioni](#)
- [Fase 4: Caricare CredSpec su Amazon S3](#)
- [Fase 5: creazione di un cluster Amazon ECS \(facoltativo\)](#)
- [Fase 6: creazione di un ruolo IAM per le istanze di container](#)
- [Fase 7: creazione di un ruolo di esecuzione dell'attività personalizzata](#)
- [Fase 8: creazione di un ruolo dell'attività per Amazon ECS Exec](#)
- [Passaggio 9: Registrare una definizione di attività che utilizzi la modalità domainless gMSA](#)
- [Fase 10: registrazione di un'istanza di container Windows nel cluster](#)
- [Fase 11: verifica dell'istanza di container](#)
- [Fase 12: esecuzione di un'attività di Windows](#)
- [Passaggio 13: verifica che il contenitore abbia gMSA credenziali](#)
- [Fase 14: pulizia](#)
- [Eseguire il debug di Amazon ECS senza dominio gMSA per contenitori Windows](#)

## Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- Il tuo utente IAM dispone delle autorizzazioni richieste specificate nell'esempio di [Amazon ECS FullAccess](#) policy IAM.
- La versione più recente di AWS CLI è installata e configurata. Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere [Installazione](#) di AWS Command Line Interface

### Note

Puoi utilizzare gli endpoint del servizio dual-stack per interagire con Amazon ECS da AWS CLI, SDKs e l'API Amazon ECS su entrambi e. IPv4 IPv6 Per ulteriori informazioni, consulta [Utilizzo degli endpoint dual-stack Amazon ECS](#).

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:

- Un Active Directory. AWS Directory Service AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida [introduttiva a AWS Managed Microsoft AD](#) nella Guida all'AWS Directory Service amministrazione.
- Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta [AWS Direct Connect](#).
- Disponi di un VPC e delle sottoreti in grado di risolvere il nome di dominio Active Directory.
- Hai scelto tra senza dominio gMSA e unendo ogni istanza a un singolo dominio. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

Quindi, scegli l'archiviazione dei dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per la modalità domainless gMSA.

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account nel contenitore. Tu generi il CredSpec file e poi lo archivia in uno dei CredSpec opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore. Per utilizzare il metodo domainless, è disponibile una sezione facoltativa in CredSpec il file può specificare le credenziali in uno dei domainless user credentials opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore.

| Posizione di archiviazione                 | Linux                            | Windows                                    |
|--------------------------------------------|----------------------------------|--------------------------------------------|
| Amazon Simple Storage Service              | CredSpec                         | CredSpec                                   |
| AWS Secrets Manager                        | credenziali utente senza dominio | credenziali utente senza dominio           |
| Amazon EC2 Systems Manager Parameter Store | CredSpec                         | CredSpec, credenziali utente senza dominio |
| File locale                                | N/D                              | CredSpec                                   |

- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza la necessità di creare una propria EC2 istanza. Per ulteriori informazioni, consulta [Cos'è AWS CloudShell?](#) nella Guida AWS CloudShell per l'utente.

## Fase 1: Creare e configurare gMSA account su Active Directory Domain Services (AD DS)

Creare e configurare un gMSA account nel dominio Active Directory.

1. Genera una chiave principale del servizio di distribuzione delle chiavi

### Note

Se lo stai utilizzando AWS Directory Service, puoi saltare questo passaggio.

La chiave principale di KDS e gMSA le autorizzazioni sono configurate con Microsoft AD AWS gestito.

Se non hai già creato un gMSA Account di servizio nel tuo dominio, devi prima generare una chiave radice del Key Distribution Service (KDS). Il KDS è responsabile della creazione, della rotazione e del rilascio di gMSA password per gli host autorizzati. Quando è `ccg.exe` necessario recuperare gMSA credenziali, contatta KDS per recuperare la password corrente.

Per verificare se la chiave radice KDS è già stata creata, esegui il seguente PowerShell cmdlet con privilegi di amministratore di dominio su un controller di dominio utilizzando il modulo. `ActiveDirectory PowerShell` Per ulteriori informazioni sul modulo, vedere [ActiveDirectory Module](#) nel sito Web Microsoft Learn.

```
PS C:\> Get-KdsRootKey
```

Se il comando restituisce un ID chiave, puoi ignorare il resto della procedura. In caso contrario, crea la chiave principale KDS eseguendo il comando seguente:

```
PS C:\> Add-KdsRootKey -EffectiveImmediately
```

Sebbene l'argomento `EffectiveImmediately` del comando implichi l'efficacia immediata della chiave, devi attendere 10 ore prima che la chiave principale KDS venga replicata e sia disponibile per l'uso su tutti i controller di dominio.



## 2. Crea il gMSA account

Per creare il gMSA account e consentire ccg.exe loro di recuperare il gMSA password, esegui PowerShell i seguenti comandi da un server o client Windows con accesso al dominio. Sostituiscilo ExampleAccount con il nome che desideri assegnare al gMSA conto.

a. 

```
PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
```

b. 

```
PS C:\> New-ADGroup -Name "ExampleAccount Authorized Hosts" -SamAccountName "ExampleAccountHosts" -GroupScope DomainLocal
```

c. 

```
PS C:\> New-ADServiceAccount -Name "ExampleAccount" -DnsHostName "contoso" -ServicePrincipalNames "host/ExampleAccount", "host/contoso" -PrincipalsAllowedToRetrieveManagedPassword "ExampleAccountHosts"
```

d. Crea un utente con una password permanente senza scadenza. Queste credenziali vengono archiviate AWS Secrets Manager e utilizzate da ogni attività per entrare a far parte del dominio.

```
PS C:\> New-ADUser -Name "ExampleAccount" -AccountPassword (ConvertTo-SecureString -AsPlainText "Test123" -Force) -Enabled 1 -PasswordNeverExpires 1
```

e. 

```
PS C:\> Add-ADGroupMember -Identity "ExampleAccountHosts" -Members "ExampleAccount"
```

f. Installa il PowerShell modulo per la creazione CredSpec oggetti in Active Directory e restituisci CredSpec JSON.

```
PS C:\> Install-PackageProvider -Name NuGet -Force
```

```
PS C:\> Install-Module CredentialSpec
```

g. 

```
PS C:\> New-CredentialSpec -AccountName ExampleAccount
```

3. Copia l'output JSON del comando precedente in un file denominato gmsa-cred-spec.json. Questa è la CredSpec file. che viene utilizzato nella fase 3, [Passaggio 3: Modifica il tuo CredSpec JSON da includere senza dominio gMSA Informazioni](#).

## Fase 2: caricamento delle credenziali in Secrets Manager

Copia le credenziali di Active Directory in un sistema di archiviazione sicuro delle credenziali, in modo che ogni attività possa recuperarle. Questa è la versione senza dominio gMSA metodo. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

Questo passaggio utilizza il. AWS CLI Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

- Esegui il AWS CLI comando seguente e sostituisci il nome utente, la password e il nome di dominio in modo che corrispondano al tuo ambiente. Mantieni l'ARN del segreto da utilizzare nella fase successiva, [Passaggio 3: Modifica il tuo CredSpec JSON da includere senza dominio gMSA Informazioni](#)

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws secretsmanager create-secret \
--name gmsa-plugin-input \
--description "Amazon ECS - gMSA Portable Identity." \
--secret-string "{\"username\": \"ExampleAccount\", \"password\": \"Test123\",
\"domainName\": \"contoso.com\"}"
```

## Passaggio 3: Modifica il tuo CredSpec JSON da includere senza dominio gMSA Informazioni

Prima di caricare il CredSpec a una delle opzioni di archiviazione, aggiungi informazioni al CredSpec con l'ARN del segreto in Secrets Manager del passaggio precedente. Per [ulteriori informazioni, vedi lo use case Additional Credential Spec Configuration for non-domain-joined Container Host nel sito Web Microsoft Learn](#).

1. Aggiungi le seguenti informazioni al CredSpec file all'interno di `ActiveDirectoryConfig`. Sostituisci l'ARN con il segreto in Secrets Manager del passaggio precedente.

Tieni presente che il valore PluginGUID deve corrispondere al GUID nel seguente frammento di codice esemplificativo ed è obbligatorio.

```
"HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
 "PluginInput": "{\"credentialArn\": \"arn:aws:secretsmanager:aws-region:111122223333:secret:gmsa-plugin-input\"}"
}
```

Puoi anche usare un segreto nell'archivio parametri SSM con l'ARN in questo formato:  
 \"arn:aws:ssm:*aws-region*:111122223333:parameter/*gmsa-plugin-input*\".

2. Dopo aver modificato il CredSpec il file dovrebbe essere simile al seguente esempio:

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
 "Sid": "S-1-5-21-4066351383-705263209-1606769140",
 "MachineAccountName": "ExampleAccount",
 "Guid": "ac822f13-583e-49f7-aa7b-284f9a8c97b6",
 "DnsTreeName": "contoso",
 "DnsName": "contoso",
 "NetBiosName": "contoso"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "ExampleAccount",
 "Scope": "contoso"
 },
 {
 "Name": "ExampleAccount",
 "Scope": "contoso"
 }
],
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
```

```
"PluginInput": "{\\"credentialArn\\": \\"arn:aws:secretsmanager:aws-
region:111122223333:secret:gmsa-plugin-input\\"}"
}
}
}
```

## Fase 4: Caricare CredSpec su Amazon S3

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

1. Copia il CredSpec file nel computer o nell'ambiente in cui si eseguono AWS CLI i comandi.
2. Esegui il AWS CLI comando seguente per caricare il CredSpec su Amazon S3. Sostituisci MyBucket con il nome del bucket Amazon S3. Puoi archiviare il file come oggetto in qualsiasi bucket e posizione, ma devi consentire l'accesso a tale bucket e tale posizione nella policy associata al ruolo di esecuzione dell'attività.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws s3 cp gmsa-cred-spec.json \
s3://MyBucket/ecs-domainless-gmsa-credspec
```

## Fase 5: creazione di un cluster Amazon ECS (facoltativo)

Per impostazione predefinita, l'account dispone di un cluster Amazon ECS denominato default. Questo cluster viene utilizzato per impostazione predefinita in AWS CLI SDKs, e AWS CloudFormation. Puoi utilizzare cluster aggiuntivi per raggruppare e organizzare le attività e l'infrastruttura e assegnare impostazioni predefinite per alcune configurazioni.

È possibile creare un cluster da AWS Management Console AWS CLI, SDKs, o AWS CloudFormation. Le impostazioni e la configurazione nel cluster non influiscono gMSA.

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

```
$ aws ecs create-cluster --cluster-name windows-domainless-gmsa-cluster
```

### Important

Se vuoi creare un cluster personalizzato, devi specificare `--cluster clusterName` per ogni comando che prevedi di usare con tale cluster.

## Fase 6: creazione di un ruolo IAM per le istanze di container

Un'istanza container è un computer host per eseguire contenitori in attività ECS, ad esempio EC2 istanze Amazon. Ogni istanza di container viene registrata in un cluster Amazon ECS. Prima di avviare EC2 le istanze Amazon e registrarle in un cluster, devi creare un ruolo IAM da utilizzare per le istanze di container.

Per creare il ruolo dell'istanza di container, consulta [Ruolo IAM delle istanze di container Amazon ECS](#). Il ruolo `ecsInstanceRole` predefinito dispone di autorizzazioni sufficienti per completare questo tutorial.

## Fase 7: creazione di un ruolo di esecuzione dell'attività personalizzata

Amazon ECS può utilizzare un ruolo IAM diverso per le autorizzazioni necessarie all'avvio di ogni attività, anziché il ruolo dell'istanza di container. Questo ruolo è il ruolo di esecuzione dell'attività. Ti consigliamo di creare un ruolo di esecuzione dell'attività con le sole autorizzazioni necessarie per l'esecuzione dell'attività da parte di ECS, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sul principio del privilegio minimo, consulta [SEC03-BP02 Concessione dell'accesso con privilegio minimo](#) nel Framework AWS Well-Architected.

1. Per creare un ruolo di esecuzione dell'attività, consulta [Creazione del ruolo di esecuzione attività](#). Le autorizzazioni predefinite consentono all'istanza del contenitore di estrarre le immagini dei contenitori da Amazon Elastic Container Registry `stdout` e `stderr` dalle tue applicazioni per registrarle su Amazon CloudWatch Logs.

Poiché il ruolo richiede autorizzazioni personalizzate per questo tutorial, puoi assegnare al ruolo un nome diverso da `ecsTaskExecutionRole`. Questo tutorial utilizza `ecsTaskExecutionRole` nelle fasi successive.

2. Aggiungi le seguenti autorizzazioni creando una policy personalizzata, una policy in linea che esiste solo per questo ruolo o una policy che puoi riutilizzare. Sostituisci l'ARN della Resource

nella prima istruzione con il bucket e la posizione di Amazon S3 e la seconda Resource con l'ARN del segreto in Secrets Manager.

Se esegui la crittografia del segreto in Secrets Manager con una chiave personalizzata, devi consentire anche `kms:Decrypt` per la chiave.

Se al posto di Secrets Manager utilizzi l'archivio parametri SSM, devi consentire `ssm:GetParameter` per il parametro, anziché `secretsmanager:GetSecretValue`.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "s3:GetObject"
],
 "Resource": "arn:aws:s3:::MyBucket/ecs-domainless-gmsa-credspec/gmsa-cred-
spec.json"
 },
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": "arn:aws:secretsmanager:aws-region:111122223333:secret:gmsa-
plugin-input"
 }
]
}
```

## Fase 8: creazione di un ruolo dell'attività per Amazon ECS Exec

Questo tutorial utilizza Amazon ECS Exec per verificare la funzionalità eseguendo un comando all'interno di un'attività in esecuzione. Per utilizzare ECS Exec, è necessario attivare ECS Exec nel servizio o nell'attività. Inoltre, il ruolo dell'attività (non il ruolo di esecuzione dell'attività) deve disporre delle autorizzazioni `ssmmessages`. Per informazioni sulla policy IAM richiesta, consulta [Autorizzazioni ECS Exec](#).

Questo passaggio utilizza il. AWS CLI Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

Per creare un ruolo di attività utilizzando il AWS CLI, procedi nel seguente modo.

1. Crea un file denominato `ecs-tasks-trust-policy.json` con i seguenti contenuti:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": "ecs-tasks.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Crea un ruolo IAM. Puoi sostituire il nome `ecs-exec-demo-task-role` ma mantenerlo per i passaggi successivi.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws iam create-role --role-name ecs-exec-demo-task-role \
--assume-role-policy-document file://ecs-tasks-trust-policy.json
```

Puoi eliminare il file `ecs-tasks-trust-policy.json`.

3. Crea un file denominato `ecs-exec-demo-task-role-policy.json` con i seguenti contenuti:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ssmmessages:CreateControlChannel",
 "ssmmessages:CreateDataChannel",

```

```

 "ssmmessages:OpenControlChannel",
 "ssmmessages:OpenDataChannel"
],
 "Resource": "*"
}
]
}

```

4. Crea una policy IAM e collegala al ruolo della fase precedente.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```

$ aws iam put-role-policy \
 --role-name ecs-exec-demo-task-role \
 --policy-name ecs-exec-demo-task-role-policy \
 --policy-document file://ecs-exec-demo-task-role-policy.json

```

Puoi eliminare il file `ecs-exec-demo-task-role-policy.json`.

## Passaggio 9: Registrare una definizione di attività che utilizzi la modalità domainless gMSA

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

1. Crea un file denominato `windows-gmsa-domainless-task-def.json` con i seguenti contenuti:

```

{
 "family": "windows-gmsa-domainless-task",
 "containerDefinitions": [
 {
 "name": "windows_sample_app",
 "image": "mcr.microsoft.com/windows/servercore/iis",
 "cpu": 1024,
 "memory": 1024,
 "essential": true,
 "credentialSpecs": [

```



```

 "credentialSpecDomainless:arn:aws:s3:::ecs-domainless-gmsa-
credspec/gmsa-cred-spec.json"
],
 "entryPoint": [
 "powershell",
 "-Command"
],
 "command": [
 "New-Item -Path C:\\inetpub\\wwwroot\\index.html -ItemType file -Value
'<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your
application is now running on a container in Amazon ECS.</p>' -Force ; C:\\
\\ServiceMonitor.exe w3svc"
],
 "portMappings": [
 {
 "protocol": "tcp",
 "containerPort": 80,
 "hostPort": 8080
 }
]
}
],
"taskRoleArn": "arn:aws:iam::111122223333:role/ecs-exec-demo-task-role",
"executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole"
}

```

2. Registra la definizione di attività con il comando seguente:

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```

$ aws ecs register-task-definition \
--cli-input-json file://windows-gmsa-domainless-task-def.json

```

## Fase 10: registrazione di un'istanza di container Windows nel cluster

Avvia un'istanza Amazon EC2 Windows ed esegui l'agente contenitore ECS per registrarla come istanza di contenitore nel cluster. ECS esegue le attività sulle istanze di container registrate nel cluster in cui vengono avviate le attività.

1. Per avviare un'istanza Amazon EC2 Windows configurata per Amazon ECS in AWS Management Console, consulta [Avvio di un'istanza di container Windows di Amazon ECS](#). Fermati alla fase relativa ai dati utente.
2. In gMSA, i dati utente devono impostare la variabile di ambiente ECS\_GMSA\_SUPPORTED prima di avviare l'agente contenitore ECS.

Per ECS Exec, l'agente deve iniziare con l'argomento `-EnableTaskIAMRole`.

Per proteggere il ruolo IAM dell'istanza impedendo alle attività di raggiungere il servizio web EC2 IMDS per recuperare le credenziali del ruolo, aggiungi l'argomento. `-AwsVpcBlockIMDS` Ciò si applica solo alle attività che utilizzano la modalità di rete `awsVpc`.

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_GMSA_SUPPORTED", $TRUE, "Machine")
Import-Module ECSTools
Initialize-ECSAgent -Cluster windows-domainless-gmsa-cluster -EnableTaskIAMRole -
AwsVpcBlockIMDS
</powershell>
```

3. Analizza un riepilogo della configurazione dell'istanza nel pannello Summary (Riepilogo) e, quando è tutto pronto, scegli Launch instance (Avvia istanza).

## Fase 11: verifica dell'istanza di container

Puoi verificare la presenza di un'istanza di container nel cluster utilizzando la AWS Management Console. Tuttavia, gMSA necessita di funzionalità aggiuntive indicate come attributi. Questi attributi non sono visibili in AWS Management Console, quindi questo tutorial utilizza il AWS CLI.

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

1. Elenca le istanze di container nel cluster. Le istanze del contenitore hanno un ID diverso dall'ID dell' EC2 istanza.

```
$ aws ecs list-container-instances
```

Output:

```
{
 "containerInstanceArns": [
 "arn:aws:ecs:aws-region:111122223333:container-
instance/default/MyContainerInstanceID"
]
}
```

Ad esempio, 526bd5d0ced448a788768334e79010fd è un ID di istanza di container valido.

2. Utilizza l'ID istanza di container della fase precedente per informazioni dettagliate sull'istanza di container. Sostituisci `MyContainerInstanceID` con l'ID.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws ecs describe-container-instances \
 ----container-instances MyContainerInstanceID
```

Nota che l'output è molto lungo.

3. Verifica che l'elenco `attributes` contenga un oggetto con la chiave denominata `name` e un valore `ecs.capability.gmsa-domainless`. Di seguito è mostrato un esempio dell'oggetto.

Output:

```
{
 "name": "ecs.capability.gmsa-domainless"
}
```

## Fase 12: esecuzione di un'attività di Windows

Esegui un'attività Amazon ECS. Se nel cluster è presente una sola istanza di container, puoi utilizzare `run-task`. Se sono presenti molte istanze di container diverse, potrebbe essere più semplice utilizzare `start-task` e specificare l'ID dell'istanza di container su cui eseguire l'attività piuttosto che aggiungere vincoli di posizionamento alla definizione di attività per controllare il tipo di istanza su cui eseguire l'attività.

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero `bash`.

1.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da `sh` e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
aws ecs run-task --task-definition windows-gmsa-domainless-task \
--enable-execute-command --cluster windows-domainless-gmsa-cluster
```

Prendi nota dell'ID attività restituito dal comando.

2. Esegui il comando seguente per verificare che l'attività sia stata avviata. Questo comando resta in attesa e non restituisce il prompt della shell (interprete di comandi) fino all'avvio dell'attività. Sostituisci `MyTaskID` con l'ID attività ottenuto nella fase precedente.

```
$ aws ecs wait tasks-running --task MyTaskID
```

## Passaggio 13: verifica che il contenitore abbia gMSA credenziali

Verifica che il contenitore dell'operazione abbia un Kerberos gettone. gMSA

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero `bash`.

1.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da `sh` e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws ecs execute-command \
--task MyTaskID \
--container windows_sample_app \
--interactive \
--command powershell.exe
```

L'output sarà un PowerShell prompt.

2. Esegui il seguente comando nel PowerShell terminale all'interno del contenitore.

```
PS C:\> klist get ExampleAccount$
```

Tieni presente che il Principal presente nell'output è uguale a quello creato in precedenza.

## Fase 14: pulizia

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero bash.

1. Interrompi l'attività. Sostituisci MyTaskID con l'ID attività della fase 12, [Fase 12: esecuzione di un'attività di Windows](#).

```
$ aws ecs stop-task --task MyTaskID
```

2. Termina l' EC2 istanza Amazon. Successivamente, l'istanza di container nel cluster verrà eliminata automaticamente dopo un'ora.

Puoi trovare e terminare l'istanza utilizzando la EC2 console Amazon. o puoi eseguire il comando seguente. Per eseguire il comando, trova l'ID dell' EC2 istanza nell'output del `aws ecs describe-container-instances` comando del passaggio 1 [Fase 11: verifica dell'istanza di container](#). i-10a64379 è un esempio di ID di istanza. EC2

```
$ aws ec2 terminate-instances --instance-ids MyInstanceID
```

3. Elimina le foto o i video CredSpec file in Amazon S3. Sostituisci MyBucket con il nome del bucket Amazon S3.

```
$ aws s3api delete-object --bucket MyBucket --key ecs-domainless-gmsa-credspec/gmsa-cred-spec.json
```

4. Elimina il segreto da Secrets Manager. Se invece hai utilizzato l'archivio parametri SSM, elimina il parametro.

Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da sh e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws secretsmanager delete-secret --secret-id gmsa-plugin-input \
--force-delete-without-recovery
```

5. Annulla la registrazione della definizione di attività ed eliminala. Dopo aver annullato la registrazione, la definizione di attività viene contrassegnata come inattiva, in modo che non possa essere utilizzata per avviare nuove attività. Puoi quindi eliminare la definizione di attività.
  - a. Annulla la registrazione della definizione di attività specificando la versione. ECS crea automaticamente delle versioni delle definizioni di attività, numerate a partire da 1. Fai riferimento alle versioni nello stesso formato delle etichette sulle immagini di container, ad esempio :1.

```
$ aws ecs deregister-task-definition --task-definition windows-gmsa-domainless-task:1
```

- b. Elimina la definizione di attività.

```
$ aws ecs delete-task-definitions --task-definition windows-gmsa-domainless-task:1
```

6. (Facoltativo) Elimina il cluster ECS, se ne hai creato uno.

```
$ aws ecs delete-cluster --cluster windows-domainless-gmsa-cluster
```

## Eseguire il debug di Amazon ECS senza dominio gMSA per contenitori Windows

## Stato dell'attività di Amazon ECS

ECS tenta di avviare esattamente un'attività una volta. Se un'attività presenta un problema, viene interrotta e impostata nello stato STOPPED. Esistono due tipi comuni di problemi relativi alle attività. Innanzitutto, le attività che non possono essere avviate. In secondo luogo, le attività in cui l'applicazione si è interrotta all'interno di uno dei container. Nel campo Motivo interrotto dell'attività AWS Management Console, esamina il motivo per cui l'attività è stata interrotta. Nella AWS CLI, descrivi l'attività e osserva il campo `stoppedReason`. Per i passaggi da seguire AWS CLI, consulta [Visualizzazione degli errori delle attività interrotte da Amazon ECS](#). AWS Management Console

## Eventi Windows

Eventi Windows per gMSA i contenitori in vengono registrati nel file di Microsoft-Windows-Containers-CCG registro e sono disponibili nel Visualizzatore eventi nella sezione Applicazioni e servizi in `Logs\Microsoft\Windows\Containers-CCG\Admin`. Per ulteriori suggerimenti sul debug, consulta [Risoluzione dei problemi relativi ai contenitori g MSAs for Windows](#) sul sito Web Microsoft Learn.

## Agente ECS gMSA plugin

Registrazione per gMSA il plug-in per l'agente ECS sull'istanza del contenitore Windows si trova nella seguente directory, `C:/ProgramData/Amazon/gmsa-plugin/` Verifica all'interno del log se le credenziali utente senza dominio sono state scaricate dalla posizione di archiviazione, ad esempio Secrets Manager, e se il formato delle credenziali è stato letto correttamente.

# Scopri come usare i contenitori g MSAs for EC2 Windows per Amazon ECS

Amazon ECS supporta l'autenticazione di Active Directory per i container Windows tramite un tipo speciale di account di servizio denominato account di servizio gestito di gruppo (gMSA, group Managed Service Account).

Le applicazioni di rete basate su Windows, ad esempio le applicazioni .NET, utilizzano spesso Active Directory per facilitare l'autenticazione e la gestione delle autorizzazioni tra utenti e servizi. Gli sviluppatori di solito progettano le loro applicazioni per l'integrazione con Active Directory e l'esecuzione su server aggiunti al dominio a tale scopo. Poiché i container Windows non possono essere aggiunti al dominio, è necessario configurare un container Windows per l'esecuzione con account gMSA.

Un container Windows in esecuzione con gMSA si affida all'istanza EC2 Amazon host per recuperare le credenziali gMSA dal controller di dominio Active Directory e fornirle all'istanza del contenitore. [Per ulteriori informazioni, consulta \*Create g for Windows containers. MSAs\*](#)

 Note

Questa funzione non è supportata per i container Windows su Fargate.

## Argomenti

- [Considerazioni](#)
- [Prerequisiti](#)
- [Configurazione gMSA per contenitori Windows su Amazon ECS](#)

## Considerazioni

Quando si utilizzano contenitori g MSAs per Windows, è necessario considerare quanto segue:

- Quando utilizzi l'AMI Windows Server 2016 Full ottimizzata per Amazon ECS per le istanze di container, il nome host del container deve essere uguale al nome dell'account gMSA definito nel file delle specifiche delle credenziali. Per specificare un nome host per un container, usa il parametro di definizione del container `hostname`. Per ulteriori informazioni, consulta [Impostazioni di rete](#).
- Hai scelto tra senza dominio gMSAe unendo ogni istanza a un singolo dominio. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

Quindi, scegli l'archiviazione dei dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per la modalità domainless gMSA.

Amazon ECS utilizza un file di specifiche delle credenziali di Active Directory (CredSpec). Questo file contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account nel contenitore. Tu generi il CredSpec file e poi lo archivia in uno dei CredSpec opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore. Per utilizzare il metodo domainless, è disponibile una sezione facoltativa in CredSpec il file può specificare le credenziali in



uno dei domainless user credentials opzioni di archiviazione nella tabella seguente, specifiche del sistema operativo delle istanze del contenitore.

| Posizione di archiviazione                 | Linux                            | Windows                                    |
|--------------------------------------------|----------------------------------|--------------------------------------------|
| Amazon Simple Storage Service              | CredSpec                         | CredSpec                                   |
| AWS Secrets Manager                        | credenziali utente senza dominio | credenziali utente senza dominio           |
| Amazon EC2 Systems Manager Parameter Store | CredSpec                         | CredSpec, credenziali utente senza dominio |
| File locale                                | N/D                              | CredSpec                                   |

## Prerequisiti

Prima di utilizzare il gMSA per la funzionalità dei container Windows con Amazon ECS, assicurati di completare quanto segue:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida [introduttiva a AWS Managed Microsoft AD](#) nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta [AWS Direct Connect](#).
- Ne hai già uno gMSA account in Active Directory. Per ulteriori informazioni, consulta [Creare contenitori g MSAs per Windows](#).
- Hai scelto di utilizzare la modalità domainless gMSA oppure l'istanza del contenitore Amazon ECS Windows che ospita l'attività Amazon ECS deve appartenere al dominio Active Directory ed essere membro del gruppo di sicurezza Active Directory che ha accesso all'account gMSA.

Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

- Hai aggiunto le autorizzazioni IAM richieste. Le autorizzazioni richieste dipendono dai metodi scelti per le credenziali iniziali e per l'archiviazione della specifica delle credenziali:
  - Se utilizzi `domainless gMSA` per le credenziali iniziali, AWS Secrets Manager sono richieste le autorizzazioni IAM per il ruolo dell' EC2 istanza Amazon.
  - Se memorizzi la specifica delle credenziali in SSM Parameter Store, le autorizzazioni IAM per Amazon EC2 Systems Manager Parameter Store sono necessarie per il ruolo di esecuzione dell'attività.
  - Se archivi la specifica delle credenziali in Amazon S3, le autorizzazioni IAM per Amazon Simple Storage Service sono necessarie per il ruolo di esecuzione delle attività.

## Configurazione gMSA per contenitori Windows su Amazon ECS

Per configurare gMSA per Windows Containers su Amazon ECS, puoi seguire il tutorial completo che include la configurazione dei prerequisiti. [Utilizzo di contenitori Amazon ECS Windows con sistema `domainless gMSA` utilizzando il AWS CLI](#)

Le seguenti sezioni trattano CredSpec configurazione in dettaglio.

### Argomenti

- [Esempio CredSpec](#)
- [Senza dominio gMSA setup](#)
- [Riferimento a un file di specifiche delle credenziali in una definizione di attività](#)

## Esempio CredSpec

Amazon ECS utilizza un file di specifiche delle credenziali che contiene gMSA metadati utilizzati per propagare il gMSA contesto dell'account nel contenitore di Windows. Puoi generare il file di specifiche delle credenziali e farvi riferimento nel campo `credentialSpec` della definizione di attività. Il file delle specifiche delle credenziali non contiene segreti.

Di seguito è riportato un file di specifiche delle credenziali di esempio:

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
```

```

 "Sid": "S-1-5-21-2554468230-2647958158-2204241789",
 "MachineAccountName": "WebApp01",
 "Guid": "8665abd4-e947-4dd0-9a51-f8254943c90b",
 "DnsTreeName": "contoso.com",
 "DnsName": "contoso.com",
 "NetBiosName": "contoso"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "WebApp01",
 "Scope": "contoso.com"
 }
]
 }
}

```

## Senza dominio gMSA setup

Consigliamo la modalità domainless gMSA invece di unire le istanze del contenitore a un singolo dominio. Utilizzando domainless gMSA, l'istanza del contenitore non è aggiunta al dominio, le altre applicazioni sull'istanza non possono utilizzare le credenziali per accedere al dominio e le attività che si uniscono a domini diversi possono essere eseguite sulla stessa istanza.

1. Prima di caricare il CredSpec a una delle opzioni di archiviazione, aggiungi informazioni al CredSpec con l'ARN del segreto in Secrets Manager o SSM Parameter Store. Per [ulteriori informazioni, vedi lo use case Additional Credential Spec Configuration for non-domain-joined Container Host nel](#) sito Web Microsoft Learn.

Senza dominio gMSA formato delle credenziali

Di seguito è riportato il formato JSON per i sistemi senza dominio gMSA credenziali per Active Directory. Archivia le credenziali in Secrets Manager o SSM Parameter Store.

```

{
 "username": "WebApp01",
 "password": "Test123!",
 "domainName": "contoso.com"
}

```

2. Aggiungi le seguenti informazioni al CredSpec file all'interno di ActiveDirectoryConfig. Sostituisci l'ARN con il segreto in Secrets Manager o SSM Parameter Store.

Tieni presente che il valore PluginGUID deve corrispondere al GUID nel seguente frammento di codice esemplificativo ed è obbligatorio.

```
"HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
 "PluginInput": "{\"credentialArn\": \"arn:aws:secretsmanager:aws-region:111122223333:secret:gmsa-plugin-input\"}"
}
```

Puoi anche usare un segreto nell'archivio parametri SSM con l'ARN in questo formato:  
 \"arn:aws:ssm:*aws-region*:111122223333:parameter/*gmsa-plugin-input*\".

3. Dopo aver modificato il CredSpec il file dovrebbe essere simile al seguente esempio:

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
 "Sid": "S-1-5-21-4066351383-705263209-1606769140",
 "MachineAccountName": "WebApp01",
 "Guid": "ac822f13-583e-49f7-aa7b-284f9a8c97b6",
 "DnsTreeName": "contoso",
 "DnsName": "contoso",
 "NetBiosName": "contoso"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "WebApp01",
 "Scope": "contoso"
 },
 {
 "Name": "WebApp01",
 "Scope": "contoso"
 }
]
 },
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
```

```

 "PluginInput": "{ \"credentialArn\": \"arn:aws:secretsmanager:aws-region:111122223333:secret:gmsa-plugin-input\" }"
 }
}

```

## Riferimento a un file di specifiche delle credenziali in una definizione di attività

Amazon ECS supporta i seguenti modi per fare riferimento al percorso del file nel campo `credentialSpecs` di una definizione di attività. Per ognuna di queste opzioni, puoi fornire `credentialSpec: odomainlesscredentialSpec:`, a seconda che tu stia unendo le istanze del contenitore a un singolo dominio o utilizzando domainless gMSA, rispettivamente.

### Bucket Amazon S3

Aggiungi le specifiche delle credenziali a un bucket Amazon S3 e quindi fai riferimento all'Amazon Resource Name (ARN) del bucket Amazon S3 nel campo `credentialSpecs` della definizione di attività.

```

{
 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpecdomainless:arn:aws:s3:::${BucketName}/${ObjectName}"
],
 ...
 }
],
 ...
}

```

Devi inoltre aggiungere le seguenti autorizzazioni come policy in linea al ruolo IAM di esecuzione di processi Amazon ECS per consentire ai processi l'accesso al bucket Amazon S3.

```

{
 "Version": "2012-10-17",

```

```

 "Statement": [
 {
 "Sid": "VisualEditor",
 "Effect": "Allow",
 "Action": [
 "s3:Get*",
 "s3:List*"
],
 "Resource": [
 "arn:aws:s3:::{bucket_name}",
 "arn:aws:s3:::{bucket_name}/{object}"
]
 }
]
 }
}

```

### Parametro dell'archivio parametri di SSM

Aggiungi le specifiche delle credenziali a un parametro dell'archivio parametri di SSM e quindi fai riferimento all'Amazon Resource Name (ARN) del parametro dell'archivio parametri di SSM nel campo `credentialSpecs` della definizione di attività.

```

{
 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpecDomainless:arn:aws:ssm:region:111122223333:parameter/parameter_name"
],
 ...
 }
],
 ...
}

```

È inoltre necessario aggiungere le seguenti autorizzazioni come policy in linea al ruolo IAM di esecuzione del processo di Amazon ECS per consentire ai processi l'accesso al parametro dell'archivio parametri di SSM.

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ssm:GetParameters"
],
 "Resource": [
 "arn:aws:ssm:region:111122223333:parameter/parameter_name"
]
 }
]
}
```

## File locale

Con i dettagli delle specifiche delle credenziali in un file locale, fai riferimento al percorso del file nel campo `credentialSpecs` della definizione di attività. Il percorso del file a cui si fa riferimento deve essere relativo alla directory `C:\ProgramData\Docker\CredentialSpecs` e utilizzare la barra rovesciata (`\`) come separatore del percorso del file.

```
{
 "family": "",
 "executionRoleArn": "",
 "containerDefinitions": [
 {
 "name": "",
 ...
 "credentialSpecs": [
 "credentialSpec:file://CredentialSpecDir\CredentialSpecFile.json"
],
 ...
 }
],
 ...
}
```

# Utilizzo di EC2 Image Builder per creare modelli personalizzati ottimizzati per Amazon ECS AMIs

AWS consiglia di utilizzare Amazon ECS Optimized AMIs perché sono preconfigurati con i requisiti e i consigli per eseguire i carichi di lavoro dei container. In alcuni casi potrebbe essere necessario personalizzare l'AMI per aggiungere software aggiuntivo. È possibile utilizzare EC2 Image Builder per la creazione, la gestione e la distribuzione delle immagini del server. Mantieni la proprietà delle immagini personalizzate create nel tuo account. È possibile utilizzare le pipeline di EC2 Image Builder per automatizzare gli aggiornamenti e l'applicazione di patch di sistema per le immagini oppure utilizzare un comando autonomo per creare un'immagine con le risorse di configurazione definite.

Crei una ricetta per la tua immagine. La ricetta include un'immagine principale ed eventuali componenti aggiuntivi. Puoi anche creare una pipeline che distribuisce la tua AMI personalizzata.

Crei una ricetta per la tua immagine. Una ricetta di immagini di Image Builder è un documento che definisce l'immagine di base e i componenti che vengono applicati all'immagine di base per produrre la configurazione desiderata per l'immagine AMI di output. Puoi anche creare una pipeline che distribuisce la tua AMI personalizzata. Per ulteriori informazioni, vedere [Come funziona EC2 Image Builder](#) nella Guida per l'utente di Image EC2 Builder.

Ti consigliamo di utilizzare una delle seguenti immagini ottimizzate per Amazon ECS AMIs come «immagine principale» in Image EC2 Builder:

- Linux
  - 023 x86 ottimizzato per Amazon ECS AL2
  - AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS
  - AMI Amazon Linux 2 kernel 5 ottimizzata per Amazon ECS
  - AMI Amazon Linux 2 x86 ottimizzata per Amazon ECS
- Windows
  - Windows 2022 Full x86 ottimizzato per Amazon ECS
  - Windows 2022 Core x86 ottimizzato per Amazon ECS
  - Windows 2019 Full x86 ottimizzato per Amazon ECS
  - Windows 2019 Core x86 ottimizzato per Amazon ECS
  - Windows 2016 completo x86 ottimizzato per Amazon ECS



Ti consigliamo inoltre di selezionare «Usa l'ultima versione del sistema operativo disponibile». La pipeline utilizzerà il controllo delle versioni semantiche per l'immagine principale, che aiuta a rilevare gli aggiornamenti delle dipendenze nei lavori pianificati automaticamente. Per ulteriori informazioni, vedere Controllo delle [versioni semantiche](#) nella Guida per l'utente di Image EC2 Builder.

AWS aggiorna regolarmente le immagini AMI ottimizzate per Amazon ECS con patch di sicurezza e la nuova versione di Container Agent. Quando utilizzi un ID AMI come immagine principale nella ricetta dell'immagine, devi controllare regolarmente gli aggiornamenti dell'immagine principale. Se sono disponibili aggiornamenti, è necessario creare una nuova versione della ricetta con l'AMI aggiornata. Ciò garantisce che le immagini personalizzate incorporino la versione più recente dell'immagine principale. Per informazioni su come creare un flusso di lavoro per aggiornare automaticamente le EC2 istanze nel cluster Amazon ECS con quelle appena create AMIs, vedi [Come creare una pipeline di rafforzamento dell'AMI e automatizzare gli aggiornamenti alla](#) tua flotta di istanze ECS.

Puoi anche specificare l'Amazon Resource Name (ARN) di un'immagine principale pubblicata tramite una pipeline di Image EC2 Builder gestita. Amazon pubblica regolarmente immagini AMI ottimizzate per Amazon ECS tramite pipeline gestite. Queste immagini sono accessibili al pubblico. È necessario disporre delle autorizzazioni corrette per accedere all'immagine. Quando utilizzi un'immagine ARN anziché un'AMI nella tua ricetta Image Builder, la pipeline utilizza automaticamente la versione più recente dell'immagine principale ogni volta che viene eseguita. Questo approccio elimina la necessità di creare manualmente nuove versioni di ricette per ogni aggiornamento.

## Utilizzo dell'immagine ARN con infrastruttura come codice (IaC)

È possibile configurare la ricetta utilizzando la console EC2 Image Builder o l'infrastruttura come codice (ad esempio AWS CloudFormation) o l' AWS SDK. Quando specificate un'immagine principale nella ricetta, potete specificare un ID EC2 AMI, un ARN dell'immagine Image Builder, un ID Marketplace AWS prodotto o un'immagine del contenitore. AWS pubblica pubblicamente sia l'immagine AMI che IDs l'immagine Image Builder ARNs di Amazon AMIs ECS ottimizzata. Di seguito è riportato il formato ARN per l'immagine:

```
arn:${Partition}:imagebuilder:${Region}:${Account}:image/${ImageName}/${ImageVersion}
```

Il formato ImageVersion è il seguente. Sostituisci *major* *minor* e *patch* con i valori più recenti.

```
<major>.<minor>.<patch>
```

È possibile sostituire `major`, `minor` e `patch` utilizzare valori specifici oppure utilizzare l'ARN senza versione di un'immagine, in modo che la pipeline rimanga up-to-date con la versione più recente dell'immagine principale. Un ARN senza versione utilizza il formato wildcard `'x.x.x'` per rappresentare la versione dell'immagine. Questo approccio consente al servizio Image Builder di passare automaticamente alla versione più recente dell'immagine. L'utilizzo di un ARN senza versione garantisce che il riferimento punti sempre all'ultima immagine disponibile, semplificando il processo di mantenimento delle immagini aggiornate durante l'implementazione. Quando crei una ricetta con la console, EC2 Image Builder identifica automaticamente l'ARN per l'immagine principale. Quando si utilizza iAc per creare la ricetta, è necessario identificare l'ARN e selezionare la versione dell'immagine desiderata o utilizzare l'arn senza versione per indicare l'ultima immagine disponibile. Ti consigliamo di creare uno script automatico per filtrare e visualizzare solo le immagini in linea con i tuoi criteri. Il seguente script Python mostra come recuperare un elenco di applicazioni ottimizzate per Amazon ECS. AMIs

Lo script accetta due argomenti opzionali: `owner` e `platform`, con i valori predefiniti di «Amazon» e «Windows» rispettivamente. I valori validi per l'argomento proprietario sono: `SelfShared`, `Amazon`, e `ThirdParty`. I valori validi per l'argomento `platform` sono `Windows` e `Linux`. Ad esempio, se esegui lo script con l'`owner` argomento impostato su `Amazon` e impostato su `Linux`, lo script genera un elenco di immagini pubblicate da Amazon, incluse immagini ottimizzate per Amazon ECS.

`platform`

```
import boto3
import argparse

def list_images(owner, platform):
 # Create a Boto3 session
 session = boto3.Session()

 # Create an EC2 Image Builder client
 client = session.client('imagebuilder')

 # Define the initial request parameters
 request_params = {
 'owner': owner,
 'filters': [
 {
 'name': 'platform',
 'values': [platform]
 }
]
 }
```

```
}

Initialize the results list
all_images = []

Get the initial response with the first page of results
response = client.list_images(**request_params)

Extract images from the response
all_images.extend(response['imageVersionList'])

While 'nextToken' is present, continue paginating
while 'nextToken' in response and response['nextToken']:
 # Update the token for the next request
 request_params['nextToken'] = response['nextToken']

 # Get the next page of results
 response = client.list_images(**request_params)

 # Extract images from the response
 all_images.extend(response['imageVersionList'])

return all_images

def main():
 # Initialize the parser
 parser = argparse.ArgumentParser(description="List AWS images based on owner and platform")

 # Add the parameters/arguments
 parser.add_argument("--owner", default="Amazon", help="The owner of the images. Default is 'Amazon'.")
 parser.add_argument("--platform", default="Windows", help="The platform type of the images. Default is 'Windows'.")

 # Parse the arguments
 args = parser.parse_args()

 # Retrieve all images based on the provided owner and platform
 images = list_images(args.owner, args.platform)

 # Print the details of the images
 for image in images:
```

```

 print(f"Name: {image['name']}, Version: {image['version']}, ARN:
{image['arn']}")

if __name__ == "__main__":
 main()

```

## Utilizzo dell'immagine ARN con AWS CloudFormation

Una ricetta di immagini Image Builder è un progetto che specifica l'immagine principale e i componenti necessari per ottenere la configurazione prevista dell'immagine di output. La risorsa viene utilizzata. `AWS::ImageBuilder::ImageRecipe` Imposta il `ParentImage` valore sull'ARN dell'immagine. Usa l'ARN senza versione dell'immagine desiderata per assicurarti che la pipeline utilizzi sempre la versione più recente dell'immagine. Ad esempio, `arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2023-ecs-optimized-x86/x.x.x`. La seguente definizione di `AWS::ImageBuilder::ImageRecipe` risorsa utilizza un ARN di immagine gestita da Amazon:

```

ECSRecipe:
 Type: AWS::ImageBuilder::ImageRecipe
 Properties:
 Name: MyRecipe
 Version: '1.0.0'
 Components:
 - ComponentArn: [<The component arns of the image recipe>]
 ParentImage: "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2023-ecs-optimized-x86/x.x.x"

```

Per ulteriori informazioni sulla [AWS::ImageBuilder::ImageRecipe](#) risorsa, consulta la Guida per l'AWS CloudFormation utente.

Puoi automatizzare la creazione di nuove immagini nella tua pipeline impostando la `Schedule` proprietà della `AWS::ImageBuilder::ImagePipeline` risorsa. La pianificazione include una condizione di avvio e un'espressione cron. Per ulteriori informazioni, consulta [AWS::ImageBuilder::ImagePipeline](#) nella Guida per l'utente di AWS CloudFormation .

L'`AWS::ImageBuilder::ImagePipeline` esempio seguente fa in modo che la pipeline esegua una build ogni giorno alle 10:00 UTC (Coordinated Universal Time). Imposta i seguenti valori:

`Schedule`

- Imposta `PipelineExecutionStartCondition` su `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`. La build inizia solo se vengono aggiornate risorse dipendenti come l'immagine principale o i componenti, che utilizzano il carattere jolly 'x' nelle loro versioni semantiche. Ciò garantisce che la build incorpori gli ultimi aggiornamenti di tali risorse.
- Imposta `ScheduleExpression` sull'espressione cron. (`0 10 * * ? *`)

```
ECSPipeline:
 Type: AWS::ImageBuilder::ImagePipeline
 Properties:
 Name: my-pipeline
 ImageRecipeArn: <arn of the recipe you created in previous step>
 InfrastructureConfigurationArn: <ARN of the infrastructure configuration
associated with this image pipeline>
 Schedule:
 PipelineExecutionStartCondition:
EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE
 ScheduleExpression: 'cron(0 10 * * ? *)'
```

## Utilizzo dell'immagine ARN con Terraform

L'approccio per specificare l'immagine e la pianificazione principali della pipeline in Terraform è in linea con quello in AWS CloudFormation. Usi la risorsa `aws_imagebuilder_image_recipe`. Imposta il `parent_image` valore sull'ARN dell'immagine. Usa l'ARN senza versione dell'immagine desiderata per assicurarti che la pipeline utilizzi sempre la versione più recente dell'immagine. Per ulteriori informazioni, consulta la documentazione di [aws\\_imagebuilder\\_image\\_recipe](#) Terraform.

Nel blocco di configurazione della schedulazione di `aws_imagebuilder_image_pipeline` resource, imposta il valore dell'`schedule_expression` argomento su un'espressione cron a tua scelta per specificare la frequenza di esecuzione della pipeline e imposta su `pipeline_execution_start_condition` `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`. Per ulteriori informazioni, consulta la documentazione [aws\\_imagebuilder\\_image\\_pipeline](#) di Terraform.

# Utilizzo di AWS Deep Learning Containers su Amazon ECS

AWS I Deep Learning Containers forniscono un set di immagini Docker per addestrare e servire modelli in TensorFlow Apache MXNet (Incubating) su Amazon ECS. I Deep Learning Containers consentono ambienti ottimizzati con TensorFlow le librerie NVIDIA CUDA (per istanze GPU) e Intel MKL (per istanze CPU). Le immagini dei contenitori per Deep Learning Container sono disponibili in Amazon ECR per fare riferimento alle definizioni di processo di Amazon ECS. Puoi utilizzare i Deep Learning Containers insieme a Amazon Elastic Inference per ridurre i costi di inferenza.

Per iniziare a usare Deep Learning Containers without Elastic Inference su Amazon ECS, consulta la [configurazione di Amazon ECS](#) nella Developer Guide.AWS Deep Learning AMIs

# Service Quotas di Amazon ECS

Quando crei le tue Account AWS, impostiamo quote predefinite (chiamate anche limiti) per le tue AWS risorse in base alla regione. Se tenti di superare la quota per una risorsa, la richiesta ha esito negativo.

La console Service Quotas è una posizione centrale in cui è possibile visualizzare e gestire le quote per AWS i servizi e richiedere un aumento della quota per molte delle risorse utilizzate. Utilizza le informazioni sulla quota che forniamo per gestire le tue risorse. Pianifica le richieste di incremento delle quote con un certo anticipo rispetto a quando ne avrai effettivamente bisogno.

Per ulteriori informazioni sulle quote di Amazon ECS e Fargate, consulta gli [endpoint e le quote di Amazon ECS](#) nel. Riferimenti generali di Amazon Web Services Per informazioni sulla limitazione (della larghezza di banda della rete) delle API nell'API Amazon ECS, consulta la sezione [Limitazione \(della larghezza di banda della rete\) delle richieste per l'API Amazon ECS](#).

## Gestione delle quote di Amazon ECS e dei AWS Fargate servizi in AWS Management Console

Amazon ECS si è integrato con Service Quotas, AWS un servizio che consente di visualizzare e gestire le quote da una posizione centrale. Per ulteriori informazioni, consulta [Cos'è Service Quotas?](#) nella Guida per l'utente di Service Quotas.

Service Quotas semplifica la ricerca del valore di tutte le quote di servizio di Amazon ECS.

### AWS Management Console

Come visualizzare le quote di servizio di Amazon ECS e Fargate tramite la AWS Management Console

1. Apri la console Service Quotas all'indirizzo <https://console.aws.amazon.com/servicequotas/>.
2. Nel pannello di navigazione, scegliere servizi AWS .
3. Dall'elenco Servizi AWS , cerca e seleziona Amazon Elastic Container Service (Amazon ECS) o AWS Fargate.

Nell'elenco Service Quotas, è possibile visualizzare il nome della quota di servizio, il valore applicato (se è disponibile), la quota predefinita AWS e se il valore della quota è adattabile.

4. Per visualizzare ulteriori informazioni su una quota di servizio, ad esempio la descrizione, scegli il nome della quota.
5. (Facoltativo) Per richiedere un aumento della quota, seleziona la quota che desideri aumentare, seleziona Richiedi un aumento della quota, inserisci o seleziona le informazioni richieste e seleziona Richiedi.

Per lavorare meglio con le quote di servizio, AWS Management Console consulta la [Service Quotas](#) User Guide. Per richiedere un aumento delle quote, consultare [Richiesta di aumento delle quote](#) nella Guida dell'utente di Service Quotas.

## AWS CLI

Come visualizzare le quote di servizio di Amazon ECS e Fargate tramite la AWS CLI

Esegui questo comando per visualizzare le quote di default di Amazon ECS.

```
aws service-quotas list-aws-default-service-quotas \
 --query 'Quotas[*]'.
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
 --service-code ecs \
 --output table
```

Esegui questo comando per visualizzare le quote di default di Fargate.

```
aws service-quotas list-aws-default-service-quotas \
 --query 'Quotas[*]'.
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
 --service-code fargate \
 --output table
```

Esegui questo comando per visualizzare le quote di Fargate applicate.

```
aws service-quotas list-service-quotas \
 --service-code fargate
```

### Note

Amazon ECS non supporta le quote applicate.



Per ulteriori informazioni sull'utilizzo delle quote di servizio utilizzando il AWS CLI, vedere Service Quotas [Command AWS CLI Reference](#). Per richiedere un aumento delle quote, consultare il comando [request-service-quota-increase](#) nella [Documentazione di riferimento sui comandi AWS CLI](#).

## Gestisci le quote di servizio Amazon ECS e i limiti di limitazione delle API

Amazon ECS è integrato con diversi sistemi Servizi AWS, tra cui Elastic Load Balancing AWS Cloud Map e Amazon. EC2 Grazie a questa stretta integrazione, Amazon ECS include diverse funzionalità come il bilanciamento del carico di servizio, Service Connect, il task networking e la scalabilità automatica dei cluster. Amazon ECS e l'altro Servizi AWS che integra con tutti mantengono le quote di servizio e i limiti di velocità delle API per garantire prestazioni e utilizzo costanti. Queste quote di servizio impediscono inoltre l'approvvigionamento accidentale di più risorse del necessario e proteggono da azioni dannose che potrebbero far aumentare la bolletta.

Acquisendo dimestichezza con le quote di servizio e i limiti di velocità delle AWS API, puoi pianificare la scalabilità dei carichi di lavoro senza preoccuparti di un peggioramento imprevisto delle prestazioni. Per ulteriori informazioni, consulta [Request throttling for the Amazon ECS API](#).

Quando si ridimensionano i carichi di lavoro su Amazon ECS, si consiglia di prendere in considerazione la seguente quota di servizio.

- AWS Fargate prevede quote che limitano il numero di attività in esecuzione simultanee per ciascuna di esse. Regione AWS Esistono quote per le attività On-Demand e Fargate Spot su Amazon ECS. Ogni quota di servizio include anche tutti i pod Amazon EKS che esegui su Fargate.
- Per le attività eseguite su EC2 istanze Amazon, il numero massimo di EC2 istanze Amazon che puoi registrare per ogni cluster è 5.000. Se utilizzi l'auto scaling del cluster Amazon ECS con un fornitore di capacità di gruppo Auto Scaling o se gestisci le istanze EC2 Amazon per il tuo cluster da solo, questa quota potrebbe diventare un collo di bottiglia nell'implementazione. Se hai bisogno di maggiore capacità, puoi creare più cluster o richiedere un aumento della quota di servizio.
- Se utilizzi l'autoscaling del cluster Amazon ECS con un fornitore di capacità di gruppo Auto Scaling, quando ridimensioni i tuoi servizi considera la quota. `Tasks in the PROVISIONING state per cluster` Questa quota è il numero massimo di attività nello PROVISIONING stato per ogni cluster per le quali i fornitori di capacità possono aumentare la capacità. Quando avvii un gran numero di attività contemporaneamente, puoi facilmente raggiungere questa quota.

Un esempio è l'implementazione simultanea di decine di servizi, ciascuno con centinaia di attività. Quando ciò accade, il provider di capacità deve avviare nuove istanze di container per effettuare le attività quando la capacità del cluster è insufficiente. Mentre il provider di capacità sta lanciando EC2 istanze Amazon aggiuntive, è probabile che il service scheduler di Amazon ECS continuerà ad avviare attività in parallelo. Tuttavia, questa attività potrebbe essere limitata a causa dell'insufficiente capacità del cluster. Lo strumento di pianificazione dei servizi Amazon ECS implementa una strategia di limitazione esponenziale e di back-off per riprovare a posizionare le attività man mano che vengono lanciate nuove istanze di container. Di conseguenza, potrebbero verificarsi tempi di implementazione o scalabilità più lenti. Per evitare questa situazione, è possibile pianificare le implementazioni dei servizi in uno dei seguenti modi. Distribuire un gran numero di attività non richiede un aumento della capacità del cluster oppure mantenere la capacità del cluster di riserva per il lancio di nuove attività.

Oltre a considerare la quota di servizio di Amazon ECS durante la scalabilità dei carichi di lavoro, considera anche la quota di servizio per gli altri Servizi AWS servizi integrati con Amazon ECS.

## Sistema di bilanciamento del carico elastico

Puoi configurare i tuoi servizi Amazon ECS per utilizzare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività. Per ulteriori informazioni e best practice consigliate su come scegliere un sistema di bilanciamento del carico, consulta. [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#)

### Quote del servizio Elastic Load Balancing

Quando scalate i carichi di lavoro, prendete in considerazione le seguenti quote di servizio Elastic Load Balancing. La maggior parte delle quote del servizio Elastic Load Balancing è regolabile ed è possibile richiedere un aumento nella console Service Quotas.

#### Application Load Balancer

Quando si utilizza un Application Load Balancer, a seconda del caso d'uso, potrebbe essere necessario richiedere un aumento della quota per:

- La **Targets per Application Load Balancer** quota, ossia il numero di obiettivi alla base dell'Application Load Balancer.
- La **Targets per Target Group per Region** quota, che è il numero di obiettivi alla base dei gruppi target.

Per ulteriori informazioni, consulta [Quotas for your Application Load Balancers](#) nella User Guide for Application Load Balancers.

## Network Load Balancer

Esistono limitazioni più rigorose sul numero di target che è possibile registrare con un Network Load Balancer. Quando si utilizza un Network Load Balancer, spesso si desidera abilitare il supporto tra zone, che comporta limitazioni di scalabilità aggiuntive sul Targets per Availability Zone. Per Network Load Balancer numero massimo di destinazioni per zona di disponibilità per ogni Network Load Balancer. Per ulteriori informazioni, consulta [Quotas for your Network Load Balancer nella Guida per l'utente di Network Load Balancers](#).

## Limitazione dell'API Elastic Load Balancing

Quando configuri un servizio Amazon ECS per utilizzare un sistema di bilanciamento del carico, i controlli di integrità del gruppo target devono essere superati prima che il servizio venga considerato integro. Per eseguire questi controlli di integrità, Amazon ECS richiama le operazioni dell'API Elastic Load Balancing per tuo conto. Se hai un gran numero di servizi configurati con sistemi di bilanciamento del carico nel tuo account, potresti rallentare le distribuzioni dei servizi a causa di potenziali limitazioni specifiche per le operazioni dell'API Elastic Load Balancing e di RegisterTarget Elastic DeregisterTarget Load Balancing. DescribeTargetHealth. Quando si verifica una limitazione, si verificano errori di limitazione nei messaggi relativi agli eventi del servizio Amazon ECS.

Se riscontri una limitazione AWS Cloud Map delle API, puoi contattarci Supporto per ricevere indicazioni su come aumentare i limiti di limitazione delle API. AWS Cloud Map Per ulteriori informazioni sul monitoraggio e la risoluzione di tali errori di limitazione, consulta. [Gestisci i problemi di limitazione di Amazon ECS](#)

## Interfacce di rete elastiche

Se le tue attività utilizzano la modalità di awsvpc rete, Amazon ECS fornisce un'interfaccia di rete elastica (ENI) unica per ogni attività. Quando i tuoi servizi Amazon ECS utilizzano un sistema di bilanciamento del carico Elastic Load Balancing, queste interfacce di rete vengono registrate anche come destinazioni per il gruppo target appropriato definito nel servizio.

## Quote di servizio di interfaccia di rete elastiche

Quando si eseguono attività che utilizzano la modalità di awsvpc rete, a ciascuna attività viene collegata un'interfaccia elastica di rete unica. Se tali attività devono essere raggiunte tramite Internet,

assegna un indirizzo IP pubblico all'elastic network interface per tali attività. Quando ridimensioni i tuoi carichi di lavoro Amazon ECS, prendi in considerazione queste due quote importanti:

- La `Network interfaces per Region` quota, ossia il numero massimo di interfacce di rete disponibili per il tuo account Regione AWS .
- La `Elastic IP addresses per Region` quota che è il numero massimo di indirizzi IP elastici in un Regione AWS.

Entrambe queste quote di servizio sono regolabili e per queste puoi richiedere un aumento dalla tua console Service Quotas. Per ulteriori informazioni, consulta le [quote dei servizi Amazon VPC](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Per i carichi di lavoro Amazon ECS ospitati su EC2 istanze Amazon, quando esegui attività che utilizzano la modalità di `awsvpc` rete, considera la quota di `Maximum network interfaces` servizio, il numero massimo di istanze di rete per ogni istanza Amazon. EC2 Questa quota limita il numero di attività che puoi effettuare su un'istanza. Non è possibile modificare la quota e non è disponibile nella console Service Quotas. Per ulteriori informazioni, [consulta Indirizzi IP per interfaccia di rete per tipo di istanza](#) nella Amazon EC2 User Guide.

Sebbene non sia possibile modificare il numero di interfacce di rete che possono essere collegate a un' EC2 istanza Amazon, puoi utilizzare la funzionalità elastic network interface trunking per aumentare il numero di interfacce di rete disponibili. Ad esempio, per impostazione predefinita, un'`c5.1large` istanza può avere fino a tre interfacce di rete. L'interfaccia di rete principale per l'istanza viene calcolata come unica. Pertanto, è possibile collegare altre due interfacce di rete all'istanza. Poiché ogni attività che utilizza la modalità di `awsvpc` rete richiede un'interfaccia di rete, in genere è possibile eseguire solo due di queste attività su questo tipo di istanza. Ciò può portare a un sottoutilizzo della capacità del cluster. Se abiliti l'elastic network interface trunking, puoi aumentare la densità dell'interfaccia di rete per inserire un numero maggiore di attività su ciascuna istanza. Con il trunking attivato, un'`c5.1large` istanza può avere fino a 12 interfacce di rete. L'istanza ha l'interfaccia di rete principale e Amazon ECS crea e collega un'interfaccia di rete «trunk» all'istanza. Di conseguenza, con questa configurazione puoi eseguire 10 attività sull'istanza anziché le due attività predefinite. Per ulteriori informazioni, consulta [Aumento delle interfacce di rete di istanze di container Amazon ECS Linux](#).

## Limitazione delle API dell'interfaccia di rete elastica

Quando esegui attività che utilizzano la modalità `awsvpc` di rete, Amazon ECS si affida al seguente Amazon. EC2 APIs. Ciascuno di questi APIs ha diverse accelerazioni API. Per ulteriori informazioni, consulta la sezione [Request throttling for the Amazon EC2 API](#) Reference EC2 .

- `CreateNetworkInterface`
- `AttachNetworkInterface`
- `DetachNetworkInterface`
- `DeleteNetworkInterface`
- `DescribeNetworkInterfaces`
- `DescribeVpcs`
- `DescribeSubnets`
- `DescribeSecurityGroups`
- `DescribeInstances`

Se le chiamate EC2 API di Amazon vengono limitate durante i flussi di lavoro di provisioning dell'interfaccia di rete elastica, lo scheduler dei servizi di Amazon ECS riprova automaticamente con back-off esponenziali. Questi ritiri automatici possono talvolta causare un ritardo nell'avvio delle attività, con conseguente rallentamento della velocità di implementazione. Quando si verifica la limitazione delle API, viene visualizzato il messaggio `Operations are being throttled. Will try again later.` nei messaggi relativi agli eventi di servizio. Se soddisfi costantemente i limiti di limitazione delle EC2 API di Amazon, puoi contattarci Supporto per ricevere indicazioni su come aumentare i limiti di limitazione delle API. [Per ulteriori informazioni sul monitoraggio e la risoluzione degli errori di limitazione, consulta Gestione dei problemi di limitazione.](#)

## AWS Cloud Map

Amazon ECS service discovery e Service Connect vengono utilizzati AWS Cloud Map APIs per gestire i namespace per i tuoi servizi Amazon ECS. Se i tuoi servizi prevedono un gran numero di attività, prendi in considerazione i seguenti consigli.

### AWS Cloud Map quote di servizio

Quando i servizi Amazon ECS sono configurati per utilizzare service discovery o Service Connect, la `Tasks per service` quota, che è il numero massimo di attività per il servizio, è influenzata dalla

quota di AWS Cloud Map Instances per service servizio, che è il numero massimo di istanze per quel servizio. In particolare, la quota AWS Cloud Map di servizio riduce il numero di istanze eseguibili fino a un massimo di 1.000 istanze per servizio. Non è possibile modificare la quota. AWS Cloud Map Per ulteriori informazioni, consulta [Service Quotas di AWS Cloud Map](#).

## AWS Cloud Map Limitazione delle API

Amazon ECS chiama il `ListInstances`, `GetInstancesHealthStatus`, `RegisterInstance`, e per tuo `DeregisterInstance` AWS Cloud Map APIs conto. Aiutano a scoprire il servizio ed eseguono controlli di integrità quando avvii un'attività. Quando vengono distribuiti contemporaneamente più servizi che utilizzano il service discovery con un gran numero di attività, ciò può comportare il superamento dei limiti di limitazione delle AWS Cloud Map API. Quando ciò accade, è probabile che venga visualizzato il seguente messaggio: `Operations are being throttled. Will try again later` nei messaggi relativi agli eventi del servizio Amazon ECS e una velocità di implementazione e avvio delle attività più lente. AWS Cloud Map non documenta i limiti di limitazione per questi. APIs Se riscontri una limitazione dovuta a questi, puoi contattarci Supporto per ricevere assistenza su come aumentare i limiti di limitazione delle API. Per ulteriori consigli sul monitoraggio e la risoluzione di tali errori di limitazione, consulta. [Gestisci i problemi di limitazione di Amazon ECS](#)

# Documentazione di riferimento dell'API Amazon ECS

Oltre a AWS Management Console and the AWS Command Line Interface (AWS CLI), Amazon ECS fornisce anche un'API. Puoi usare l'API per automatizzare le attività per la gestione delle risorse Amazon ECS.

- Per un elenco delle operazioni API in base alla risorsa Amazon ECS, consulta [Azioni della risorsa Amazon ECS](#).
- Per un elenco alfabetico delle operazioni API, consulta [Operazioni](#).
- Per un elenco alfabetico dei tipi di dati, consulta la pagina [Tipi di dati](#).
- Per un elenco di parametri di query comuni, consulta la pagina [Parametri Comuni](#).
- Per le descrizioni dei codici di errore, consulta la pagina [Errori comuni](#).

Per ulteriori informazioni su AWS CLI, consulta il [AWS Command Line Interface riferimento per Amazon ECS](#).

## Cronologia dei documenti

La tabella riportata di seguito illustra i principali aggiornamenti e le nuove funzioni per la Guida per gli sviluppatori di Amazon Elastic Container Service. Inoltre, aggiorniamo frequentemente la documentazione per tener conto del feedback inviatoci.

| Modifica                                                                                                        | Descrizione                                                                                                                                                                                                                                                                                                                                                 | Data             |
|-----------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per la velocità di inizializzazione dei volumi Amazon EBS collegati alle attività di Amazon ECS.       | Ora puoi impostare un <code>volumeInitializationRate</code> per determinare la velocità, in MiB/s, con cui i dati vengono recuperati da uno snapshot di un volume Amazon EBS esistente per creare nuovi volumi da allegare alle attività di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">volumeInitializationRate</a> .                     | 13 maggio 2025   |
| Support per l'interruzione dell'implementazione di un servizio.                                                 | Ora puoi interrompere la distribuzione di un servizio. Per ulteriori informazioni, consulta <a href="#">Interruzione delle distribuzioni di servizi Amazon ECS</a> .                                                                                                                                                                                        | 2 maggio 2025    |
| Support per endpoint dual-stack Amazon ECS.                                                                     | Ora puoi utilizzare gli endpoint dual-stack per interagire con Amazon ECS. Gli endpoint dual-stack supportano le richieste ad Amazon ECS tramite Internet Protocol versione 4 (IPv4) e Internet Protocol versione 6 (IPv6). Per ulteriori informazioni, consulta <a href="#">Usare gli endpoint dual-stack di Amazon ECS</a> .                              | 30 aprile 2025   |
| Support per la configurazione di una modalità di log driver predefinita a livello di account in un Regione AWS. | È ora possibile configurare una modalità di consegna predefinita dei messaggi di registro da un contenuto al driver di registro specificata nell'impostazione di <code>logConfiguration</code> utilizzo dell' <code>defaultLogDriverMode</code> account. Per ulteriori informazioni, consulta <a href="#">Modalità predefinita del driver di registro</a> . | 17 aprile 2025   |
| Support per la migrazione di un servizio con un formato ARN breve al formato ARN lungo.                         | È ora possibile migrare un servizio con un formato ARN breve al pacchetto in formato ARN lungo. Per ulteriori informazioni, consulta <a href="#">Migrare un ARN di servizio breve Amazon ECS a un ARN lungo</a> .                                                                                                                                           | 13 febbraio 2025 |



| Modifica                                                                                                       | Descrizione                                                                                                                                                                                                                                                                                               | Data             |
|----------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Support per tre tipi di fault injection su Amazon ECS e Fargate.                                               | Ora puoi utilizzare la porta blackhole di rete, la latenza di rete e l'iniezione di errori di perdita dei pacchetti di rete con Amazon ECS e Fargate. Per ulteriori informazioni, consulta <a href="#">Usalo AWS Fault Injection Service per testare i carichi di lavoro Amazon ECS.</a>                  | 19 dicembre 2024 |
| Support per CloudWatch Container Insights con osservabilità migliorata.                                        | Puoi utilizzare CloudWatch Container Insights con una migliore osservabilità. Per ulteriori informazioni, consulta <a href="#">Monitorare i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata.</a>                                                                       | 2 dicembre 2024  |
| Support per la scalabilità predittiva.                                                                         | Puoi utilizzare la scalabilità predittiva per anticipare le esigenze future e aumentare in modo proattivo le attività del tuo servizio secondo necessità. Per ulteriori informazioni, consulta <a href="#">Utilizzare modelli storici per scalare i CloudWatch servizi con la scalabilità predittiva.</a> | 21 novembre 2024 |
| Support per il ribilanciamento dei servizi.                                                                    | Ora puoi fare in modo che Amazon ECS bilanci automaticamente le attività di servizio tra le zone di disponibilità. Per ulteriori informazioni, consulta <a href="#">Bilanciamento di un servizio Amazon ECS tra le zone di disponibilità.</a>                                                             | 19 novembre 2024 |
| Support per la configurazione della risoluzione dei tag delle immagini del contenitore nei digest di immagini. | È ora possibile configurare la risoluzione dei tag di immagine del contenitore in digest per ogni contenitore nelle attività che fanno parte di un servizio. Per ulteriori informazioni, consultare <a href="#">Risoluzione dell'immagine del contenitore</a> e <a href="#">versionConsistency</a> .      | 19 novembre 2024 |
| Aggiungi una nuova AmazonECSInfrastructureRole PolicyForVpcLattice policy IAM                                  | La <a href="#">AmazonECSInfrastructureRole PolicyForVpcLattice</a> policy fornisce l'accesso ad altre risorse di AWS servizio necessarie per gestire la funzionalità VPC Lattice nei carichi di lavoro Amazon ECS per tuo conto.                                                                          | 18 novembre 2024 |

| Modifica                                                                                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                             | Data             |
|-----------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| VPC Lattice è ora disponibile per i servizi Amazon ECS.                                             | Ora puoi usare VPC Lattice per standardizzare la service-to-service connettività, la sicurezza e l'osservabilità. Per ulteriori informazioni, consulta <a href="#">Usa Amazon VPC Lattice per connettere, osservare e proteggere i tuoi servizi Amazon ECS</a> .                                                                                                                        | 18 novembre 2024 |
| Support per la cronologia dei servizi utilizzando distribuzioni e revisioni dei servizi.            | È possibile utilizzare le distribuzioni e le revisioni dei servizi per visualizzare la cronologia delle distribuzioni. Per ulteriori informazioni, consulta <a href="#">Visualizzazione della cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS e le revisioni dei servizi Amazon ECS</a> .                                                                    | 30 ottobre 2024  |
| Support per i volumi Amazon EBS per il tipo di EC2 avvio che utilizza il sistema operativo Windows. | Amazon ECS supporta i volumi Amazon EBS collegati alle attività in esecuzione su EC2 istanze con sistema operativo Windows. Per ulteriori informazioni, consulta <a href="#">Usare i volumi Amazon EBS con Amazon ECS</a> .                                                                                                                                                             | 24 ottobre 2024  |
| Aggiungi autorizzazioni ad Amazon ECSInfrastructureRolePolicyForVolumes.                            | La AmazonECSInfrastructureRolePolicyForVolumes policy è stata aggiornata per consentire ai clienti di creare un volume Amazon EBS da uno snapshot. Per ulteriori informazioni, consulta <a href="#">Amazon ECSInfrastructure RolePolicyForVolumes</a> .                                                                                                                                 | 10 ottobre 2024  |
| Fargate Spot è disponibile per Linux ARM.                                                           | Il supporto per Fargate Spot su Linux ARM è stato aggiunto ad Amazon ECS.                                                                                                                                                                                                                                                                                                               | 6 settembre 2024 |
| Politica di riavvio dei container per singoli contenitori nelle attività di Amazon ECS.             | Puoi abilitare le politiche di riavvio per i contenuti essenziali e non essenziali definiti nella definizione dell'attività per superare più rapidamente i guasti transitori e mantenere la disponibilità delle attività. Per ulteriori informazioni, consulta <a href="#">Riavvia i singoli contenitori nelle attività di Amazon ECS con le politiche di riavvio dei contenitori</a> . | 15 agosto 2024   |

| Modifica                                                                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Autorizzazioni aggiunte a <a href="#">the section called “Amazon ECS_FullAccess”</a>                                  | La AmazonECS_FullAccess policy è stata aggiornata per aggiungere iam:PassRole le autorizzazioni per i ruoli IAM per un ruolo denominato. ecsInfrastructureRole Questo è il ruolo IAM predefinito creato da AWS Management Console che deve essere utilizzato come ruolo dell'infrastruttura ECS che consente ad Amazon ECS di gestire i volumi Amazon EBS collegati alle attività ECS.                                              | 13 agosto 2024 |
| Aggiunto il supporto per Amazon Linux 2023, CentOS Stream 9, Fedora 40 e Ubuntu 24 per ECS Anywhere                   | Il supporto per i sistemi operativi Amazon Linux 2023, CentOS Stream 9, Fedora 40 e Ubuntu 24 è stato aggiunto a ECS Anywhere. Per ulteriori informazioni, consulta <a href="#">Sistemi operativi e architetture di sistema supportati</a> .                                                                                                                                                                                        | 23 luglio 2024 |
| Risoluzione dell'immagine del contenitore per i servizi che utilizzano il tipo di distribuzione Rolling Update (ECS). | Per garantire che tutte le attività in un servizio che utilizza il controller di distribuzione Rolling Update utilizzino la stessa immagine del contenitore, Amazon ECS risolve i nomi delle immagini dei contenitori e tutti i tag di immagine specificati nella definizione dell'attività in digest di immagini del contenitore. <a href="#">Per ulteriori informazioni, consulta Risoluzione dell'immagine del contenitore</a> . | 10 luglio 2024 |
| Aggiunto il supporto per Debian 11 e Debian 12 per ECS Anywhere                                                       | Il supporto per i sistemi operativi Debian 11 e Debian 12 è stato aggiunto a ECS Anywhere. Per ulteriori informazioni, consulta <a href="#">Sistemi operativi e architetture di sistema supportati</a> .                                                                                                                                                                                                                            | 28 marzo 2024  |
| Supporto gMSA per contenitori Linux su Fargate                                                                        | Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux su Fargate tramite un tipo speciale di account di servizio chiamato Managed Service Account (gMSA) di gruppo. Per ulteriori informazioni, consulta Utilizzo <a href="#">gMSA for Linux container su</a> Fargate.                                                                                                                                        | 5 marzo 2024   |

| Modifica                                                                      | Descrizione                                                                                                                                                                                                                                            | Data            |
|-------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| CloudWatch parametri aggiunti per i volumi Amazon EBS collegati alle attività | Amazon ECS ora pubblica i CloudWatch parametri per l'utilizzo dello storage Amazon EBS per le attività a cui è collegato un volume Amazon EBS. Per ulteriori informazioni, consulta i <a href="#">parametri di Amazon ECS. CloudWatch</a>              | 8 febbraio 2024 |
| Service Connect TLS                                                           | Ora puoi usare <a href="#">TLS con Service Connect</a> .                                                                                                                                                                                               | 22 gennaio 2024 |
| Politica gestita da Service Connect TLS                                       | Aggiunta una nuova ECSInfrastructure RolePolicyForServiceConnectTransportLayerSecurity politica di <a href="#">Amazon</a> .                                                                                                                            | 22 gennaio 2024 |
| Aggiornamento della configurazione del timeout di Service Connect             | La <a href="#">configurazione del timeout</a> di Service Connect può ora essere aggiornata e include due parametri opzionali: <code>idleTimeout</code> e <code>perRequestTimeout</code> .                                                              | 22 gennaio 2024 |
| Drenaggio delle istanze gestite da Amazon ECS                                 | Puoi utilizzare il <a href="#">drenaggio gestito delle istanze</a> di Amazon ECS per facilitare la chiusura ordinata delle istanze Amazon ECS.                                                                                                         | 19 gennaio 2024 |
| È stato aggiunto il supporto per Ubuntu 2.2 per ECS Anywhere                  | Il supporto per il sistema operativo Ubuntu 22 è stato aggiunto a ECS Anywhere. Per ulteriori informazioni, consulta <a href="#">Sistemi operativi e architetture di sistema supportati</a> .                                                          | 16 gennaio 2024 |
| Aggiungi la politica AmazonECSInfrastructureRolePolicyForVolumes IAM          | È stato aggiunto <a href="#">AmazonECSInfrastructureRolePolicyForVolumes</a> . La policy concede le autorizzazioni necessarie ad Amazon ECS per effettuare chiamate AWS API per gestire i volumi Amazon EBS associati ai carichi di lavoro Amazon ECS. | 11 gennaio 2024 |

| Modifica                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data             |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Volume di dati Amazon EBS per attività Amazon ECS                     | Puoi configurare 1 <a href="#">volume di dati Amazon EBS</a> per attività durante la distribuzione per il collegamento a attività Amazon ECS autonome o attività gestite da un servizio ECS. La configurazione di un volume al momento della distribuzione consente di creare definizioni di attività riutilizzabili non limitate a tipi o impostazioni di volume specifici. I volumi Amazon EBS forniscono uno storage a blocchi ad alta disponibilità, conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. | 11 gennaio 2024  |
| La console classica Amazon ECS ha raggiunto la fine del ciclo di vita | La console Amazon ECS ha raggiunto la fine del ciclo di vita.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | 4 dicembre 2023  |
| Policy aggiornata                                                     | La policy IAM ECSService RolePolicy gestita da <a href="#">Amazon</a> è stata aggiornata con nuove events autorizzazioni e autorizzazioni autoscaling e autoscaling-plans aggiuntive.                                                                                                                                                                                                                                                                                                                                                                               | 4 dicembre 2023  |
| Supporto per il monitoraggio del runtime                              | Puoi utilizzare Runtime Monitoring per monitorare e i carichi di lavoro Amazon ECS e identificare comportamenti dannosi o non autorizzati. <a href="#">Per ulteriori informazioni, consulta Runtime Monitoring.</a>                                                                                                                                                                                                                                                                                                                                                 | 26 novembre 2023 |
| Policy aggiornata                                                     | La policy IAM <a href="#">AmazonECSServiceRolePolicy</a> gestita è stata aggiornata per consentire l'accesso all' AWS Cloud Map DiscoverInstancesRevision API.                                                                                                                                                                                                                                                                                                                                                                                                      | 4 ottobre 2023   |
| AWS Fargate configurazione del ritiro delle attività                  | <a href="#">Puoi configurare il periodo di attesa prima che le attività di Fargate vengano ritirate. Per ulteriori informazioni, Consulta Manutenzione delle attività di AWS Fargate.</a>                                                                                                                                                                                                                                                                                                                                                                           | 5 settembre 2023 |

| Modifica                                                                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Parametri aggiuntivi di definizione delle attività in AWS Fargate                   | AWS Fargate aggiunge il supporto per <code>pidMode</code> e <code>systemControls</code> nella versione della piattaforma Linux1.4.0. Per ulteriori informazioni, consulta <a href="#">Definizioni delle attività</a> .                                                                                                                                                                                                                                              | 9 agosto 2023  |
| Riprogettazione della pagina di definizione delle attività della console Amazon ECS | La pagina di definizione delle attività nella console Amazon ECS è stata riprogettata e contiene opzioni aggiuntive. Per ulteriori informazioni, consulta <a href="#">Creazione di una definizione delle attività utilizzando la console</a> .                                                                                                                                                                                                                      | 26 luglio 2023 |
| Fargate supporta il caricamento lento con gli indici Seekable OCI                   | AWS Fargate sta introducendo gli indici Seekable OCI (SOCI). Con SOCI, i container trascorrono solo pochi secondi sul recupero dell'immagine prima di avviarsi, lasciando così il tempo necessario per la configurazione dell'ambiente e la creazione di istanze dell'applicazione mentre l'immagine viene scaricata in background. Per ulteriori informazioni, consulta <a href="#">Lazy loading delle immagini dei container utilizzando Seekable OCI (SOC)</a> . | 17 luglio 2023 |
| Supporto migliorato per gMSA su Linux e Windows                                     | La definizione delle attività ha un nuovo campo <code>credentialSpecs</code> per gMSA per Linux e Windows. È stato aggiunto un nuovo tutorial completo per gMSA senza dominio su Windows, consulta <a href="#">Tutorial: utilizzo di container Windows con gMSA senza dominio utilizzando l' AWS CLI</a> . Per ulteriori informazioni, vedere <a href="#">Uso di g MSAs per contenitori Linux e Utilizzo di g MSAs per contenitori Windows</a> .                    | 14 luglio 2023 |

| Modifica                                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Data           |
|---------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Documentazione migliorata sulle versioni dell'agente di ECS                           | La documentazione per le versioni dell'agente di Amazon ECS è stata aggiornata. Consigliamo di utilizzare la versione v20.10.13 o successive del Docker con la versione più recente dell'agente del container Amazon ECS. Le versioni rilasciate e le modifiche all'agente sono disponibili su GitHub. Per maggiori informazioni, consulta <a href="#">Versioni dell'agente del container Amazon ECS Linux</a> su GitHub.                                                                                         | 20 giugno 2023 |
| Disponibilità regionale aggiornata per il supporto ARM64 Fargate                      | La disponibilità regionale per il ARM64 supporto Fargate è stata aggiornata. Per ulteriori informazioni, consulta <a href="#">Considerazioni</a> .                                                                                                                                                                                                                                                                                                                                                                | 19 giugno 2023 |
| Miglioramento della documentazione relativa al dimensionamento automatico del cluster | La documentazione per la scalabilità Amazon ECS di Amazon Auto EC2 Scaling presenta miglioramenti significativi in termini di precisione e leggibilità. Per maggiori informazioni, consulta <a href="#">Dimensionamento automatico del cluster Amazon ECS</a> .                                                                                                                                                                                                                                                   | 4 maggio 2023  |
| Autorizzazione all'assegnazione di tag per la creazione di risorse.                   | Gli utenti devono disporre dell'autorizzazione per le operazioni che creano la risorsa, ad esempio <code>ecsCreateCluster</code> . Quando crei una risorsa e specifichi i tag per quella risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vi siano le autorizzazioni per creare tag. Per ulteriori informazioni, consulta <a href="#">Autorizzazione all'assegnazione di tag</a> e <a href="#">Concessione dell'autorizzazione per assegnare tag alle risorse al momento della creazione</a> . | 18 aprile 2023 |
| Support per contenitori gMSA per Linux su EC2                                         | Puoi usare gMSA per autenticarti su contenitori Active Directory per Linux. EC2 Per ulteriori informazioni, consulta <a href="#">Using g MSAs for</a> Linux Containers.                                                                                                                                                                                                                                                                                                                                           | 14 aprile 2023 |

| Modifica                                                                       | Descrizione                                                                                                                                                                                                                                                                                                         | Data             |
|--------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per l'archiviazione temporanea per i container Windows su AWS Fargate | Puoi utilizzare lo spazio di archiviazione temporaneo per i container Windows su AWS Fargate. Per ulteriori informazioni, consulta <a href="#">Archiviazione delle attività di Fargate</a> .                                                                                                                        | 14 aprile 2023   |
| AWS Cost Management supporto per dati CUR a livello di attività                | Puoi attivare i costi e l'utilizzo delle risorse a livello di attività in Report di costi e utilizzo. Ciò aggiunge i dati di allocazione dei costi suddivisi per le attività eseguite su AWS EC2 Fargate e. Per ulteriori informazioni, consulta <a href="#">Report di costi e utilizzo a livello di attività</a> . | 12 aprile 2023   |
| AMI Amazon Linux 2023 ottimizzata per Amazon ECS                               | Puoi implementare carichi di lavoro sull'AMI Amazon Linux 2023 ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                             | 10 aprile 2023   |
| AWS Fargate Standard federale per l'elaborazione delle informazioni (FIPS) 140 | Puoi distribuire carichi di lavoro su Amazon ECS AWS Fargate in modo conforme al Federal Information Processing Standard (FIPS) 140. Per ulteriori informazioni, consulta <a href="#">AWS Fargate Standard federale per l'elaborazione delle informazioni (FIPS-140)</a> .                                          | 10 aprile 2023   |
| Eliminazione di definizioni di attività                                        | Puoi eliminare una definizione di attività utilizzando la console, l'SDK e la AWS CLI di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Eliminazione di una revisione della definizione delle attività utilizzando la console</a> e <a href="#">Definizioni delle attività</a> .                      | 24 febbraio 2023 |
| AWS Fargate consigli sui servizi in Compute Optimizer                          | AWS Compute Optimizer genera consigli sulle dimensioni delle attività e dei contenitori in base all'utilizzo delle attività in esecuzione nei servizi Amazon ECS su Fargate. AWS Per ulteriori informazioni, consulta <a href="#">Visualizzazione dei suggerimenti per i servizi Amazon ECS su Fargate</a> .        | 27 gennaio 2023  |



| Modifica                                                                                      | Descrizione                                                                                                                                                                                                                                                                    | Data             |
|-----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Console Amazon ECS                                                                            | La nuova console Amazon ECS è ora la console predefinita. Per ulteriori informazioni, consulta <a href="#">Nuova console Amazon ECS</a> .                                                                                                                                      | 19 gennaio 2023  |
| Policy IAM AmazonECS_FullAccess aggiornata                                                    | La policy IAM AmazonECS_FullAccess è stata aggiornata in modo da includere le autorizzazioni per aggiungere tag ai sistemi di bilanciamento del carico durante la creazione. Per ulteriori informazioni, consulta <a href="#">Amazon ECS_FullAccess</a> .                      | 4 gennaio 2023   |
| Usa gli CloudWatch allarmi per rilevare gli errori di distribuzione del servizio Amazon ECS   | Puoi configurare Amazon ECS in modo che la distribuzione non sia riuscita quando rileva che uno specifico CloudWatch allarme è passato allo stato ALARM. Per ulteriori informazioni, consulta <a href="#">the section called "Implementazioni di aggiornamento continuo"</a> . | 19 dicembre 2022 |
| Supporto per la mappatura delle porte dei container                                           | È possibile impostare un intervallo di numeri di porta nel container associato all'intervallo di porte host mappato in maniera dinamica. Per ulteriori informazioni, consulta <a href="#">the section called "Mappature di porte"</a> .                                        | 15 dicembre 2022 |
| Disponibilità generale di Amazon ECS Service Connect                                          | Questa funzionalità aggiunge la funzione di individuazione dei servizi e la mesh di servizi controllati dalle implementazioni dei servizi Amazon ECS. Per ulteriori informazioni, consulta <a href="#">the section called "Service Connect"</a> .                              | 27 novembre 2022 |
| L'esperienza della nuova console Amazon ECS per le definizioni di attività è stata aggiornata | L'esperienza della nuova console Amazon ECS ora contiene un editor JSON per le definizioni di attività. Per ulteriori informazioni, consulta <a href="#">the section called "Creazione di una definizione di attività attraverso la nuova console"</a> .                       | 27 ottobre 2022  |

| Modifica                                                                                      | Descrizione                                                                                                                                                                                                                                                                                                 | Data              |
|-----------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| L'esperienza della nuova console Amazon ECS per le definizioni di attività è stata aggiornata | L'esperienza della nuova console Amazon ECS ora contiene un editor JSON per le definizioni di attività. Per ulteriori informazioni, consulta <a href="#">the section called “Creazione di una definizione di attività attraverso la nuova console”</a> .                                                    | 27 ottobre 2022   |
| L'esperienza della nuova console Amazon ECS è stata aggiornata                                | L'esperienza della nuova console Amazon ECS è stata aggiornata con parametri di attività e servizi aggiuntivi. Per ulteriori informazioni, consultare <a href="#">the section called “Creazione di un servizio”</a> e <a href="#">the section called “Esecuzione di un'applicazione come attività”</a> .    | 7 ottobre 2022    |
| Nuove informazioni nell'endpoint dei metadati delle attività versione 4                       | L'endpoint dei metadati delle attività versione 4 ora include l'ID VPC e il nome del servizio. Per ulteriori informazioni, consulta <a href="#">the section called “Endpoint metadati delle attività versione 4”</a> .                                                                                      | 7 ottobre 2022    |
| Dimensione della nuova definizione di attività                                                | Amazon ECS su Fargate ora supporta le dimensioni di attività a 8 vCPU e 16 vCPU. Per ulteriori informazioni, consulta <a href="#">the section called “Dimensioni processo”</a> .                                                                                                                            | 16 settembre 2022 |
| Pagine della CLI ECS archiviate                                                               | La documentazione della CLI ECS è stata archiviata. Ti consigliamo di utilizzare AWS Copilot per le tue esigenze relative agli strumenti da riga di comando. Per ulteriori informazioni, consulta <a href="#">Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot</a> . | 15 settembre 2022 |
| Nuove quote Fargate                                                                           | Fargate sta passando da quote basate sul numero di attività a quote basate su vCPU. Per ulteriori informazioni, consulta <a href="#">Endpoint e quote di Amazon ECS</a> .                                                                                                                                   | 8 settembre 2022  |

| Modifica                                                       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Data             |
|----------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Support per le piscine calde di Amazon EC2 Auto Scaling.       | Ora puoi utilizzare i pool caldi di Amazon EC2 Auto Scaling per scalare le tue applicazioni più velocemente e risparmiare sui costi. Per ulteriori informazioni, consulta <a href="#">Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling</a> .                                                                                                                                                                                                                                                                                                                                                      | 23 marzo 2022    |
| Supporto per le istanze Windows in ECS Anywhere.               | ECS Anywhere ora supporta le istanze Windows. Per ulteriori informazioni, consulta <a href="#">Cluster Amazon ECS per il tipo di lancio esterno</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 | 3 marzo 2022     |
| Aggiunto il supporto ECS Exec per istanze esterne              | ECS Exec ora è supportato per le istanze esterne. Per ulteriori informazioni, consulta <a href="#">Monitora i contenitori Amazon ECS con ECS Exec</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | 24 gennaio 2022  |
| L'esperienza della nuova console Amazon ECS è stata aggiornata | L'esperienza della nuova console Amazon ECS supporta la creazione e l'eliminazione di un cluster, l'aggiornamento di una definizione di attività e l'annullamento della registrazione di una definizione di attività. Per ulteriori informazioni, consulta <a href="#">Creazione di un cluster Amazon ECS per il tipo di lancio Fargate</a> , <a href="#">Eliminazione di un cluster Amazon ECS</a> , <a href="#">Aggiornamento di una definizione di attività Amazon ECS tramite la console</a> e <a href="#">Annullamento della registrazione di una revisione della definizione di attività di Amazon ECS tramite la console</a> . | 8 dicembre 2021  |
| L'esperienza della nuova console Amazon ECS è stata aggiornata | L'esperienza della nuova console Amazon ECS supporta la creazione di una definizione di attività. Per ulteriori informazioni, consulta <a href="#">Creazione di una definizione di attività Amazon ECS utilizzando la console</a> .                                                                                                                                                                                                                                                                                                                                                                                                   | 23 novembre 2021 |
| Amazon ECS supporta l'architettura ARM a 64 bit per Linux.     | Amazon ECS supporta l'architettura ARM a 64 bit della CPU per il sistema operativo Linux. Per ulteriori informazioni, consulta <a href="#">the section called "Definizioni delle attività per carichi di lavoro ARM a 64 bit"</a> .                                                                                                                                                                                                                                                                                                                                                                                                   | 23 novembre 2021 |

| Modifica                                                          | Descrizione                                                                                                                                                                                                                                                                                                                       | Data             |
|-------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto Amazon ECS per l'opzione fluentd log-driver-buffer-limit | Amazon ECS supporta l'opzione <code>log-driver-buffer-limit</code> fluentd. Per ulteriori informazioni, consulta <a href="#">Inviare i log di Amazon ECS a un servizio o AWSAWS Partner</a> .                                                                                                                                     | 22 novembre 2021 |
| Script di build per AMI Linux ottimizzata per Amazon ECS          | Amazon ECS ha reso open-source gli script della build che vengono utilizzati per creare le varianti Linux dell'AMI ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Script di build per AMI Linux ottimizzata per Amazon ECS</a> .                                                                    | 19 novembre 2021 |
| Integrità delle istanze di container                              | Amazon ECS aggiunge il supporto per il monitoraggio dell'integrità delle istanze di container. Per ulteriori informazioni, consulta <a href="#">Monitora lo stato delle istanze dei container Amazon ECS</a> .                                                                                                                    | 10 novembre 2021 |
| Supporto per Windows Amazon ECS Exec                              | Amazon ECS Exec supporta Windows. Per ulteriori informazioni, consulta <a href="#">Monitora i contenitori Amazon ECS con ECS Exec</a> .                                                                                                                                                                                           | 1° novembre 2021 |
| Supporto per container Windows su Fargate.                        | Amazon ECS supporta container Windows su Fargate. Per ulteriori informazioni, consulta <a href="#">Registro delle modifiche della versione della piattaforma Fargate Windows</a> .                                                                                                                                                | 28 ottobre 2021  |
| Supporto GPU per istanze esterne su Amazon ECS Anywhere           | Amazon ECS supporta la specifica dei requisiti GPU nella definizione di attività per i processi eseguiti su istanze esterne. Per ulteriori informazioni, consultare <a href="#">Definizioni di attività Amazon ECS per carichi di lavoro GPU</a> e <a href="#">Registrazione di un'istanza esterna in un cluster Amazon ECS</a> . | 8 ottobre 2021   |
| Supporto della modalità di rete awsipc su Windows                 | Amazon ECS supporta la modalità di rete <code>awsipc</code> su Windows. Per ulteriori informazioni, consulta <a href="#">Assegna un'interfaccia di rete per un'attività Amazon ECS</a> .                                                                                                                                          | 15 luglio 2021   |

| Modifica                                                   | Descrizione                                                                                                                                                                                                                             | Data           |
|------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Disponibilità generale di Bottlerocket                     | Amazon ECS supporta una variante AMI ottimizzata per Amazon ECS del sistema operativo Bottlerocket fornito come AMI. Per ulteriori informazioni, consulta <a href="#">Ottimizzato per Amazon ECS Bottlerocket AMIs</a> .                | 30 giugno 2021 |
| Aggiornamento dei processi pianificati di Amazon ECS       | Amazon EventBridge ha aggiunto il supporto per parametri aggiuntivi durante la creazione di regole che attivano le attività pianificate di Amazon ECS.                                                                                  | 25 giugno 2021 |
| AWS politiche gestite per Amazon ECS                       | Amazon ECS ha aggiunto la documentazione delle politiche AWS gestite per i ruoli collegati ai servizi. Per ulteriori informazioni, consulta <a href="#">AWS politiche gestite per Amazon Elastic Container Service</a> .                | 8 giugno 2021  |
| Iniziare con AWS CDK                                       | È stata aggiunta una guida introduttiva per l'utilizzo AWS CDK con Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Creazione di risorse Amazon ECS utilizzando AWS CDK</a> .                                               | 27 maggio 2021 |
| Amazon ECS Anywhere                                        | Amazon ECS ha aggiunto il supporto per la registrazione di un server on-premise o di una macchina virtuale (VM) con il cluster. Per ulteriori informazioni, consulta <a href="#">Cluster Amazon ECS per il tipo di lancio esterno</a> . | 25 maggio 2021 |
| AMI di base Windows Server 20H2 ottimizzata per Amazon ECS | Amazon ECS ha aggiunto il supporto per una nuova variante AMI ottimizzata per Windows Amazon ECS per Windows Server 20H2 Core. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .             | 19 aprile 2021 |
| Amazon ECS Exec                                            | Amazon ECS ha rilasciato un nuovo strumento di debug chiamato ECS Exec. Per ulteriori informazioni, consulta <a href="#">Monitora i contenitori Amazon ECS con ECS Exec</a> .                                                           | 15 marzo 2021  |

| Modifica                                                                       | Descrizione                                                                                                                                                                                                                                                                                                                                                 | Data             |
|--------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per la policy degli endpoint VPC                                      | Amazon ECS ora supporta le policy dell'endpoint VPC. Per ulteriori informazioni, consulta <a href="#">Creazione di una policy per l'endpoint VPC per Amazon ECS</a> .                                                                                                                                                                                       | 11 gennaio 2021  |
| Nuova esperienza della console                                                 | Amazon ECS ha rilasciato una nuova esperienza della console che supporta la creazione o l'aggiornamento di un servizio o l'esecuzione di un processo autonomo. Per ulteriori informazioni, consultare <a href="#">Creazione di un servizio Amazon ECS utilizzando la console</a> e <a href="#">Esecuzione di un'applicazione come attività Amazon ECS</a> . | 28 dicembre 2020 |
| Aggiornamento del provider di capacità                                         | Amazon ECS ha aggiunto il supporto per l'aggiornamento di un provider di capacità del gruppo Auto Scaling esistente.                                                                                                                                                                                                                                        | 23 novembre 2020 |
| ECS ora supporta le attività di Amazon FSx for Windows File Server per Windows | Amazon ECS ha aggiunto il supporto per specificare i volumi Amazon FSx for Windows File Server nelle definizioni delle attività di Windows. Per ulteriori informazioni, consulta <a href="#">Utilizzo FSx per volumi Windows File Server con Amazon ECS</a> .                                                                                               | 11 novembre 2020 |
| Aggiunto il supporto della modalità dual-stack VPC                             | Amazon ECS ha aggiunto il supporto per l'utilizzo di un VPC in modalità dual-stack con attività che utilizzano aws-vpc la modalità di rete, che fornisce supporto per gli indirizzi IPv6. Per ulteriori informazioni, consulta <a href="#">Utilizzo di un VPC in modalità dual-stack</a> .                                                                  | 5 novembre 2020  |
| Aggiornamento endpoint metadati dei processi v4                                | Amazon ECS ha aggiunto ulteriori metadati all'output dell'endpoint dei metadati del processo v4. Per ulteriori informazioni, consulta <a href="#">Endpoint di metadati delle attività Amazon ECS versione 4</a> .                                                                                                                                           | 5 novembre 2020  |

| Modifica                                                                                    | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                 | Data             |
|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per Local Zones e zone Wavelength                                                  | Amazon ECS ha aggiunto il supporto per carichi di lavoro in Local Zones e zone Wavelength. Per ulteriori informazioni, consulta <a href="#">Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones</a> .                                                                                                                                                                                            | 4 settembre 2020 |
| AMI della variante di Bottlerocket per Amazon ECS                                           | Bottlerocket è un sistema operativo open source basato su Linux creato appositamente per l'esecuzione di container. AWS Una variante dell'AMI ottimizzata per Amazon ECS del sistema operativo Bottlerocket è fornita come AMI che può essere utilizzata all'avvio di istanze di container di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Ottimizzato per Amazon ECS Bottlerocket AMIs</a> .               | 31 agosto 2020   |
| Endpoint metadati dei processi versione 4 aggiornato per le statistiche di velocità di rete | La versione 4 dell'endpoint dei metadati delle attività è stata aggiornata per fornire statistiche sulla velocità di rete per le attività di Amazon ECS che utilizzano le modalità di bridge rete awsvpc o ospitate su EC2 istanze Amazon che eseguono almeno una versione 1.43.0 dell'agente contenitore. Per ulteriori informazioni, consulta <a href="#">Endpoint di metadati delle attività Amazon ECS versione 4</a> . | 10 agosto 2020   |
| Parametri di utilizzo di Fargate                                                            | AWS Fargate fornisce metriche di CloudWatch utilizzo che forniscono visibilità sull'utilizzo delle risorse Fargate On-Demand e Fargate Spot da parte degli account. Per ulteriori informazioni, consulta <a href="#">Parametri di utilizzo</a> .                                                                                                                                                                            | 3 agosto 2020    |

| Modifica                                                                       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                | Data           |
|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| AWS Copilot versione 0.1.0                                                     | È stata lanciata la nuova AWS CLI di Copilot, che fornisce comandi di alto livello per semplificare la modellazione, la creazione, il rilascio e la gestione di applicazioni containerizzate su Amazon ECS da un ambiente di sviluppo locale. Per ulteriori informazioni, consulta <a href="#">Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot</a> .               | 9 luglio 2020  |
| AWS Fargate pianificazione della deprecazione delle versioni della piattaforma | È stato aggiunto il programma di dichiarazione delle versioni della piattaforma Fargate come obsolete. Per ulteriori informazioni, consulta <a href="#">AWS Deprecazione della versione della piattaforma Fargate Linux</a> .                                                                                                                                                                              | 8 luglio 2020  |
| AWS Espansione della regione di Fargate                                        | Amazon ECS on AWS Fargate si è esteso alla regione Europa (Milano).                                                                                                                                                                                                                                                                                                                                        | 25 giugno 2020 |
| AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS rilasciata              | Amazon ECS ha rilasciato un'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS per carichi di lavoro inferenziali.<br><br>Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                 | 24 giugno 2020 |
| Aggiunto il supporto per l'eliminazione dei provider di capacità               | Amazon ECS ha aggiunto il supporto per l'eliminazione dei provider di capacità del gruppo Auto Scaling.                                                                                                                                                                                                                                                                                                    | 11 giugno 2020 |
| AWS Aggiornamento della piattaforma Fargate versione 1.4.0                     | A partire dal 28 maggio 2020, qualsiasi nuovo processo Fargate avviato utilizzando la versione 1.4.0 della piattaforma avrà la sua archiviazione temporanea da 20 GB crittografata con un algoritmo di crittografia AES-256 tramite una chiave di crittografia gestita da AWS Fargate. Per ulteriori informazioni, consulta <a href="#">Archiviazione effimera delle attività Fargate per Amazon ECS</a> . | 28 maggio 2020 |



| Modifica                                | Descrizione                                                                                                                                                                                                                                                                                                           | Data           |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Supporto file di variabili di ambiente  | È stato aggiunto il supporto per specificare i file delle variabili di ambiente in una definizione di attività, che consente di aggiungere in blocco variabili di ambiente ai contenitori. Per ulteriori informazioni, consulta <a href="#">Passa una singola variabile di ambiente a un contenitore Amazon ECS</a> . | 18 maggio 2020 |
| AWS Espansione della regione di Fargate | AWS Fargate con Amazon ECS si è estesa alla regione Africa (Città del Capo).                                                                                                                                                                                                                                          | 11 maggio 2020 |
| Quota di servizio aggiornata            | <p>È stata aggiornata la seguente quota di servizio:</p> <ul style="list-style-type: none"><li>• I numero di cluster per account è stato aumentato da 2,000 a 10,000.</li></ul> <p>Per ulteriori informazioni, consulta <a href="#">Endpoint e quote di Amazon ECS</a>.</p>                                           | 17 aprile 2020 |

| Modifica                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Data          |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| AWS Piattaforma Fargate versione 1.4.0 | <p>AWS Viene rilasciata la versione 1.4.0 della piattaforma Fargate, che contiene le seguenti funzionalità:</p> <ul style="list-style-type: none"><li>• Aggiunto il supporto per l'utilizzo dei volumi del file system Amazon EFS per l'archiviazione dei processi persistenti. Per ulteriori informazioni, consulta <a href="#">Usa i volumi Amazon EFS con Amazon ECS</a>.</li><li>• L'archiviazione effimera delle attività è stata aumentata a 20 GB. Per ulteriori informazioni, consulta <a href="#">Archiviazione effimera delle attività Fargate per Amazon ECS</a>.</li><li>• Il comportamento del traffico di rete da e verso le attività è stato aggiornato. A partire dalla versione 1.4 della piattaforma, tutti i processi Fargate ricevono un'unica interfaccia di rete elastica (denominata ENI di processo) e tutto il traffico di rete scorre attraverso tale ENI all'interno del VPC e sarà visibile all'utente attraverso i log di flusso del VPC. Per ulteriori informazioni, consulta le <a href="#">opzioni di task networking di Amazon ECS per il tipo di lancio Fargate</a>.</li><li>• Task ENIs aggiunge il supporto per i jumbo frame. Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimensione e del payload più grande che si adatta all'interno di un singolo frame. Più grande è l'MTU, più il payload dell'applicazione può essere adattato all'interno di un singolo fotogramma, riducendo il sovraccarico per fotogramma e aumentando l'efficienza. Il supporto dei frame jumbo riduce il sovraccarico quando il percorso di rete tra l'attività e la destinazione</li></ul> | 8 aprile 2020 |

| Modifica | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              | Data |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|
|          | <p>supporta frame jumbo, ad esempio tutto il traffico che rimane all'interno del VPC.</p> <ul style="list-style-type: none"> <li>CloudWatch Container Insights includerà metriche delle prestazioni di rete per le attività di Fargate. Per ulteriori informazioni, consulta <a href="#">Monitora i contenuti Amazon ECS utilizzando Container Insights con osservabilità migliorata</a>.</li> <li>Aggiunto il supporto per l'endpoint dei metadati dei processi v4 che fornisce informazioni aggiuntive per i processi Fargate, incluse le statistiche di rete per l'attività e la zona di disponibilità in cui l'attività è in esecuzione. Per ulteriori informazioni, consulta <a href="#">Endpoint di metadati delle attività Amazon ECS versione 4</a>.</li> <li>Aggiunto il supporto per il parametro Linux <code>SYS_PTRACE</code> nelle definizioni dei container. Per ulteriori informazioni, consulta <a href="#">Parametri Linux</a>.</li> <li>L'agente del container Fargate sostituisce l'uso dell'agente del container Amazon ECS per tutti i processi di Fargate. Questa modifica non dovrebbe avere effetto sull'esecuzione delle attività.</li> <li>Il runtime del container ora utilizza Containerd invece di Docker. Questa modifica non dovrebbe avere effetto sull'esecuzione delle attività. Si noterà che alcuni messaggi di errore che hanno origine dal runtime del container passeranno dal menzionare Docker a errori più generici.</li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Versioni della piattaforma Fargate per Amazon ECS</a>.</p> |      |

| Modifica                                                                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | Data             |
|-----------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per file system Amazon EFS per i volumi di processi                                                          | I file system Amazon EFS possono essere utilizzati come volumi di dati per i processi Amazon ECS e Fargate. Per ulteriori informazioni, consulta <a href="#">Usa i volumi Amazon EFS con Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                     | 8 aprile 2020    |
| Endpoint metadati dei processi versione 4 di Amazon ECS                                                               | A partire dalla versione dell'agente del container Amazon ECS 1.39.0 e della piattaforma Fargate 1.4.0, una variabile di ambiente denominata ECS_CONTAINER_METADATA_URI_V4 viene inserita in ciascun container di un processo. Quando esegui una query sull'endpoint dei metadati delle attività versione 4, per le attività vengono resi disponibili diversi metadati delle attività e <a href="#">statistiche Docker</a> . Per ulteriori informazioni, consulta <a href="#">Endpoint di metadati delle attività Amazon ECS versione 4</a> . | 8 aprile 2020    |
| Supporto per versioni specifiche di segreti di Secrets Manager da inserire come variabili d'ambiente                  | Aggiunto il supporto per specificare dati sensibili utilizzando versioni specifiche di segreti di Secrets Manager. Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a> .                                                                                                                                                                                                                                                                                                              | 24 febbraio 2020 |
| Sono state aggiunte opzioni di configurazione CodeDeploy di distribuzione aggiuntive per le implementazioni blu/verdi | Il CodeDeploy servizio ha aggiunto nuove configurazioni di distribuzione canarie e lineari per il tipo di distribuzione Amazon ECS. È inoltre possibile definire configurazioni di distribuzione personalizzate. Per ulteriori informazioni, consulta <a href="#">Convalida lo stato di un servizio Amazon ECS prima della distribuzione</a> .                                                                                                                                                                                                | 6 febbraio 2020  |
| È stato aggiunto il parametro di definizione dell'attività efsVolume Configuration                                    | Il parametro di definizione dei processi efsVolume Configuration è in anteprima pubblica, il che semplifica l'utilizzo dei file system Amazon EFS con i processi Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Usa i volumi Amazon EFS con Amazon ECS</a> .                                                                                                                                                                                                                                                                    | 17 gennaio 2020  |

| Modifica                                                                       | Descrizione                                                                                                                                                                                                                                                                                   | Data             |
|--------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Comportamento di registrazione dell'agente del container Amazon ECS aggiornato | I percorsi di registrazione e il comportamento di rotazione dell'agente del container di Amazon ECS sono stati aggiornati. Per ulteriori informazioni, consulta <a href="#">Parametri di configurazione del registro dell'agente container Amazon ECS</a> .                                   | 13 gennaio 2020  |
| Fargate Spot                                                                   | Amazon ECS ha aggiunto il supporto per l'esecuzione di processi che utilizzano Fargate Spot. Per ulteriori informazioni, consulta <a href="#">Cluster Amazon ECS per Fargate</a> .                                                                                                            | 3 dicembre 2019  |
| Auto Scaling del cluster                                                       | Il dimensionamento automatico del cluster Amazon ECS consente di avere maggiore controllo sul dimensionamento dei processi in un cluster. Per ulteriori informazioni, consulta <a href="#">Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster</a> . | 3 dicembre 2019  |
| Provider di capacità del cluster                                               | I provider di capacità del cluster Amazon ECS determinano l'infrastruttura da utilizzare per i processi. Per ulteriori informazioni, consulta <a href="#">Cluster Amazon ECS</a> .                                                                                                            | 3 dicembre 2019  |
| Creazione di un cluster su un AWS Outposts                                     | Amazon ECS ora supporta la creazione di cluster su un AWS Outposts. Per ulteriori informazioni, consulta <a href="#">the section called “Amazon Elastic Container Service su AWS Outposts”</a> .                                                                                              | 3 dicembre 2019  |
| Eventi di operazioni di servizio                                               | Amazon ECS ora invia eventi ad Amazon EventBridge quando si verificano determinate azioni di servizio. Per ulteriori informazioni, consulta <a href="#">Eventi di azione del servizio Amazon ECS</a> .                                                                                        | 25 novembre 2019 |

| Modifica                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                             | Data              |
|--------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| L'AMI ottimizzata per GPU di Amazon ECS supporta istanze G4. | Amazon ECS ha aggiunto il supporto per la famiglia di tipi di istanze g4 quando si utilizza l'AMI ottimizzata per GPU di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Definizioni di attività Amazon ECS per carichi di lavoro GPU</a> .                                                                                                                                | 8 ottobre 2019    |
| FireLens per Amazon ECS                                      | FireLens per Amazon ECS è disponibile a livello generale. FireLens per Amazon ECS consente di utilizzare i parametri di definizione delle attività per indirizzare i log verso una destinazione di AWS servizio o partner per l'archiviazione e l'analisi dei log. Per ulteriori informazioni, consulta <a href="#">Inviare i log di Amazon ECS a un servizio o AWSAWS Partner</a> .    | 30 settembre 2019 |
| AWS Espansione della regione di Fargate                      | AWS Fargate con Amazon ECS si è estesa alle regioni di Europa (Parigi), Europa (Stoccolma) e Medio Oriente (Bahrein).                                                                                                                                                                                                                                                                   | 30 settembre 2019 |
| Deep Learning Containers con Elastic Inference su Amazon ECS | Amazon ECS supporta il collegamento degli acceleratori Amazon Elastic Inference ai container per rendere più efficiente l'esecuzione dei carichi di lavoro di inferenza di deep learning.                                                                                                                                                                                               | 3 settembre 2019  |
| FireLens per Amazon ECS                                      | FireLens per Amazon ECS è disponibile in anteprima pubblica. FireLens per Amazon ECS consente di utilizzare i parametri di definizione delle attività per indirizzare i log verso una destinazione di AWS servizio o partner per l'archiviazione e l'analisi dei log. Per ulteriori informazioni, consulta <a href="#">Inviare i log di Amazon ECS a un servizio o AWSAWS Partner</a> . | 30 agosto 2019    |

| Modifica                                                           | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|--------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| CloudWatch Informazioni sui container                              | CloudWatch Container Insights è ora disponibile al pubblico. Consente di raccogliere, aggregare e riepilogare parametri e log dalle applicazioni e dai microservizi containerizzati. Per ulteriori informazioni, consulta <a href="#">Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata</a> .                                                                                                                                                                                                           | 30 agosto 2019 |
| Configurazione dello swap a livello di container                   | Amazon ECS ha aggiunto il supporto per il controllo dell'utilizzo dello spazio di memoria swap sulle istanze di container Linux a livello di container. Utilizzando una configurazione swap per container, ogni container all'interno di una definizione di attività può avere lo swap abilitato o disabilitato e, per chi lo ha abilitato, la quantità massima di spazio di swapping utilizzato o può essere limitata. Per ulteriori informazioni, consulta <a href="#">Gestione dello spazio di memoria di swap dei container su Amazon ECS</a> . | 16 agosto 2019 |
| AWS Espansione della regione di Fargate                            | AWS Fargate con Amazon ECS si è estesa alla regione Asia Pacifico (Hong Kong).                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | 6 agosto 2019  |
| Trunking dell'interfaccia di rete elastica                         | Sono stati aggiunti altri tipi di EC2 istanze Amazon supportati per la funzionalità di trunking ENI. Per ulteriori informazioni, consulta <a href="#">Istanze supportate per interfacce di rete di container Amazon ECS potenziate</a> .                                                                                                                                                                                                                                                                                                            | 1 agosto 2019  |
| Registrazione di più gruppi di destinazione con un servizio        | Aggiunto il supporto per la specifica di più gruppi di destinazione in una definizione del servizio. Per ulteriori informazioni, consulta <a href="#">Registrazione di più gruppi target con un servizio Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                           | 30 luglio 2019 |
| Specifica di dati sensibili utilizzando segreti di Secrets Manager | Aggiunto un tutorial per la specifica di dati sensibili tramite segreti di Secrets Manager. Per ulteriori informazioni, consulta <a href="#">Specificazione di dati sensibili utilizzando i segreti di Secrets Manager in Amazon ECS</a> .                                                                                                                                                                                                                                                                                                          | 20 luglio 2019 |

| Modifica                                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data           |
|------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| CloudWatch Container Insights                                                | Amazon ECS ha aggiunto il supporto per CloudWatch Container Insights. Per ulteriori informazioni, consulta <a href="#">Monitora i contenitori Amazon ECS utilizzando Container Insights con osservabilità migliorata</a> .                                                                                                                                                                                                                                                                                                                                                                 | 9 luglio 2019  |
| Autorizzazioni a livello di risorsa per servizi e set di processi Amazon ECS | Amazon ECS ha esteso il supporto per le autorizzazioni a livello di risorsa per i servizi e i processi di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Come funziona Amazon Elastic Container Service con IAM</a> .                                                                                                                                                                                                                                                                                                                                                        | 27 giugno 2019 |
| Nuova AMI ottimizzata per Amazon ECS con patch per -2019-005 AWS             | Amazon ECS ha aggiornato l'AMI ottimizzata per Amazon ECS per fare fronte alle vulnerabilità descritte in <a href="#">AWS-2019-005</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 17 giugno 2019 |
| Trunking dell'interfaccia di rete elastica                                   | Amazon ECS introduce il supporto per il lancio di istanze di container utilizzando tipi di EC2 istanze Amazon supportati con una maggiore densità di elastic network interface (ENI). L'uso di questi tipi di istanze e il consenso esplicito all'impostazione dell'account <code>aws-vpc-trunking</code> fornisce una densità ENI aumentata su nuove istanze di container avviate che consente di posizionare più attività su ogni istanza di container. Per ulteriori informazioni, consulta <a href="#">Aumento delle interfacce di rete di istanze di container Amazon ECS Linux</a> . | 6 giugno 2019  |
| AWS Aggiornamento della piattaforma Fargate versione 1.3.0                   | A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log <code>sp1unk</code> in aggiunta ai driver di log <code>awslogs</code> . Per ulteriori informazioni, consulta <a href="#">Archiviazione e registrazione</a> .                                                                                                                                                                                                                                                                                                                           | 1 maggio 2019  |



| Modifica                                                   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Data          |
|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| AWS Aggiornamento della piattaforma Fargate versione 1.3.0 | A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta il riferimento ai dati sensibili nella configurazione dei log di un container utilizzando il parametro di definizione del container <code>secretOptions</code> . Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a> .                                                                                                 | 1 maggio 2019 |
| AWS Aggiornamento della piattaforma Fargate versione 1.3.0 | A partire dal 2 aprile 2019, qualsiasi nuova attività di Fargate che verrà lanciata supporta l'iniezione di dati sensibili nei contenitori archiviando i dati sensibili in AWS Secrets Manager segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a> .                       | 2 aprile 2019 |
| AWS Aggiornamento della piattaforma Fargate versione 1.3.0 | A partire dal 27 marzo 2019, i nuovi processi Fargate avviati possono utilizzare ulteriori parametri di definizione dei processi che consentono di definire una configurazione proxy, le dipendenze per l'avvio e l'arresto di container, nonché un relativo valore di timeout per container. Per ulteriori informazioni, consulta <a href="#">Configurazione del proxy</a> , <a href="#">Dipendenze per i container</a> e <a href="#">Timeout del container</a> . | 27 marzo 2019 |
| Amazon ECS presenta il tipo di implementazione esterna     | Il tipo di implementazione esterna consente di usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implementazione per un servizio Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Implementa i servizi Amazon ECS utilizzando un controller di terze parti</a> .                                                                                                                                  | 27 marzo 2019 |

| Modifica                                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data            |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| AWS Contenitori di Deep Learning su Amazon ECS                               | AWS I Deep Learning Containers sono un set di immagini Docker per addestrare e servire modelli TensorFlow su Amazon Elastic Container Service (Amazon ECS). I Deep Learning Containers forniscono ambienti ottimizzati con TensorFlow librerie Nvidia CUDA (per istanze GPU) e Intel MKL (per istanze CPU) e sono disponibili in Amazon ECR. Per ulteriori informazioni, consulta <a href="#">Utilizzo di AWS Deep Learning Containers su Amazon ECS</a> . | 27 marzo 2019   |
| Amazon ECS presenta la gestione ottimizzata delle dipendenze per i container | Amazon ECS presenta parametri aggiuntivi di definizione dei processi che consentono di definire le dipendenze per l'avvio e l'arresto di container, nonché un relativo valore di timeout per container. Per ulteriori informazioni, consulta <a href="#">Dipendenze per i container</a> .                                                                                                                                                                  | 7 marzo 2019    |
| Amazon ECS presenta l'API PutAccountSettingDefault                           | Amazon ECS introduce l'API PutAccountSettingDefault che consente a un utente di impostare lo stato di attivazione del formato ARN/ID predefinito per tutti gli utenti e i ruoli dell'account. In precedenza, l'impostazione dello stato di attivazione predefinito dell'account richiedeva l'uso del proprietario dell'account.<br><br>Per ulteriori informazioni, consulta <a href="#">Nomi di risorse Amazon (ARNs) e IDs</a> .                          | 8 febbraio 2019 |
| Amazon ECS supporta carichi di lavoro per GPU                                | Amazon ECS introduce il supporto per i carichi di lavoro della GPU consentendo di creare cluster con istanze di container abilitate per GPU. In una definizione di attività è possibile specificare il numero di operazioni richieste GPUs e l'agente ECS aggiungere a quelle fisiche al contenitore. GPUs<br><br>Per ulteriori informazioni, consulta <a href="#">Definizioni di attività Amazon ECS per carichi di lavoro GPU</a> .                      | 4 febbraio 2019 |

| Modifica                                      | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                    | Data             |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Amazon ECS ha esteso il supporto dei segreti  | <p>Amazon ECS ha esteso il supporto per l'utilizzo di AWS Secrets Manager segreti direttamente nelle definizioni delle attività per iniettare dati sensibili nei contenitori.</p> <p>Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a>.</p>                                                                                                          | 21 gennaio 2019  |
| Endpoint VPC di interfaccia (AWS PrivateLink) | <p>Aggiunto il supporto per la configurazione degli endpoint VPC dell'interfaccia con tecnologia AWS PrivateLink. Ciò consente di creare una connessione privata tra il VPC e Amazon ECS senza richiedere l'accesso tramite Internet, un'istanza NAT, una connessione VPN o AWS Direct Connect.</p> <p>Per ulteriori informazioni, consulta <a href="#">Endpoint VPC di interfaccia (AWS PrivateLink)</a>.</p> | 26 dicembre 2018 |

| Modifica                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Data             |
|----------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AWS Piattaforma Fargate versione 1.3.0 | <p>Rilasciata la nuova versione della piattaforma AWS Fargate, che contiene:</p> <ul style="list-style-type: none"><li>• È stato aggiunto il supporto per l'utilizzo AWS Systems Manager dei parametri Parameter Store per iniettare dati sensibili nei contenitori.</li></ul> <p>Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a>.</p> <li>• Aggiunta la funzione di riavvio dei processi per i processi Fargate, che corrisponde al processo di aggiornamento dei processi che sono parte di un servizio Amazon ECS.</li> <p>Per ulteriori informazioni, consulta la sezione <a href="#">Ritiro e manutenzione delle attività AWS Fargate su Amazon ECS</a>.</p> <p>Per ulteriori informazioni, consulta <a href="#">Versioni della piattaforma Fargate per Amazon ECS</a>.</p> | 17 dicembre 2018 |

| Modifica                                                         | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data             |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Restrizioni dei servizi aggiornate                               | <p>Le seguenti restrizioni dei servizi sono state aggiornate:</p> <ul style="list-style-type: none"> <li>• Il numero di cluster per regione, per account è stato calcolato tra 1000 e 2000.</li> <li>• Il numero di istanze di container per cluster è stato calcolato tra 1000 e 2000.</li> <li>• Il numero di servizi per cluster è stato calcolato tra 500 e 1000.</li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Service Quotas di Amazon ECS</a>.</p> | 14 dicembre 2018 |
| AWS Espansione della regione di Fargate                          | <p>AWS Fargate con Amazon ECS si è estesa alle regioni di Asia Pacifico (Mumbai) e Canada (Centrale).</p> <p>Per ulteriori informazioni, consulta <a href="#">Regioni supportate per Amazon ECS su AWS Fargate</a>.</p>                                                                                                                                                                                                                                                    | 7 dicembre 2018  |
| Implementazioni blu/verde di Amazon ECS                          | <p>Amazon ECS ha aggiunto il supporto per le distribuzioni blu/green utilizzando CodeDeploy. Questo tipo di distribuzione consente di verificare una nuova distribuzione di un servizio prima che questo riceva traffico di produzione.</p> <p>Per ulteriori informazioni, consulta <a href="#">Convalida lo stato di un servizio Amazon ECS prima della distribuzione</a>.</p>                                                                                            | 27 novembre 2018 |
| AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS rilasciata | <p>Amazon ECS ha rilasciato un Amazon Linux AMIs 2 ottimizzato per Amazon ECS per l'architettura arm64.</p> <p>Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a>.</p>                                                                                                                                                                                                                                                             | 26 novembre 2018 |

| Modifica                                                                           | Descrizione                                                                                                                                                                                                                                                            | Data             |
|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Aggiunto supporto per contrassegni Docker aggiuntivi nelle definizioni di processo | Amazon ECS ha introdotto il supporto per i seguenti flag Docker nelle definizioni di processo: <ul style="list-style-type: none"><li>• <a href="#">Modalità IPC</a></li><li>• <a href="#">Modalità PID</a></li></ul>                                                   | 16 novembre 2018 |
| Supporto per i segreti di Amazon ECS                                               | Amazon ECS ha aggiunto il supporto per l'utilizzo AWS Systems Manager dei parametri Parameter Store per iniettare dati sensibili nei contenitori.<br><br>Per ulteriori informazioni, consulta <a href="#">Trasferisci dati sensibili a un contenitore Amazon ECS</a> . | 15 novembre 2018 |
| Aggiunta di tag alle risorse                                                       | Amazon ECS ha aggiunto il supporto per l'aggiunta di tag di metadati a servizi, definizioni di processi, cluster e istanze di container.<br><br>Per ulteriori informazioni, consulta <a href="#">Taggare le risorse Amazon ECS</a> .                                   | 15 novembre 2018 |
| AWS Espansione della regione di Fargate                                            | AWS Fargate con Amazon ECS si è estesa alle regioni Stati Uniti occidentali (California settentrionale) e Asia Pacifico (Seoul).<br><br>Per ulteriori informazioni, consulta <a href="#">AWS Fargate per Amazon ECS</a> .                                              | 7 novembre 2018  |

| Modifica                                                 | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Data            |
|----------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Restrizioni dei servizi aggiornate                       | <p>Le seguenti restrizioni dei servizi sono state aggiornate:</p> <ul style="list-style-type: none"> <li>Il numero di processi che utilizzano il tipo di avvio Fargate, per regione, per account è stato aumentato da 20 a 50.</li> <li>Il numero di indirizzi IP pubblici per i processi che utilizzano il tipo di avvio Fargate è stato aumentato da 20 a 50.</li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Service Quotas di Amazon ECS</a>.</p> | 31 ottobre 2018 |
| AWS Espansione della regione di Fargate                  | <p>AWS Fargate con Amazon ECS si è estesa alla regione Europa (Londra).</p> <p>Per ulteriori informazioni, consulta <a href="#">AWS Fargate per Amazon ECS</a>.</p>                                                                                                                                                                                                                                                                                                  | 26 ottobre 2018 |
| AMI Amazon Linux 2 ottimizzata per Amazon ECS rilasciata | <p>Amazon ECS offre Linux AMIs ottimizzati per il servizio in due varianti. La versione più recente e consigliata si basa su x;. Anche i server Amazon ECS si AMIs basano su, ma ti consigliamo di migrare i carichi di lavoro alla variante Amazon Linux 2, poiché il supporto per l'AMI Amazon Linux terminerà entro il 30 giugno 2020.</p> <p>Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a>.</p>                     | 18 ottobre 2018 |

| Modifica                                                                       | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Data              |
|--------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Endpoint dei metadati dei processi versione 3 di Amazon ECS                    | A partire dalla versione 1.21.0 dell'agente del container di Amazon ECS, l'agente inserisce una variabile di ambiente denominata <code>ECS_CONTAINER_METADATA_URI</code> in ogni container in un processo. Quando esegui una query sull'endpoint dei metadati dell'attività versione 3, sono disponibili diversi metadati delle attività e <a href="#">statistiche Docker</a> per le attività che utilizzano la modalità di rete <code>awsvpc</code> in un endpoint HTTP fornito dall'agente del container Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Monitora i carichi di lavoro utilizzando i metadati Amazon ECS</a> . | 18 ottobre 2018   |
| Espansione della regione per l'individuazione dei servizi di Amazon ECS        | L'individuazione dei servizi di Amazon ECS ha esteso il supporto alle regioni Canada (Centrale), Sud America (San Paolo), Asia Pacifico (Seoul), Asia Pacifico (Mumbai) ed Europa (Parigi).<br><br>Per ulteriori informazioni, consulta <a href="#">Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS</a> .                                                                                                                                                                                                                                                                                                           | 27 settembre 2018 |
| Supporto aggiunto per i flag Docker aggiuntivi nelle definizioni dei container | Amazon ECS ha introdotto il supporto per i seguenti flag Docker nelle definizioni del container: <ul style="list-style-type: none"> <li>• <a href="#">Controlli di sistema</a></li> <li>• <a href="#">Interactive</a></li> <li>• <a href="#">Pseudoterminale</a></li> </ul>                                                                                                                                                                                                                                                                                                                                                                  | 17 settembre 2018 |



| Modifica                                                                                                 | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Data              |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Supporto per l'autenticazione del registro privato per Amazon ECS utilizzando le attività di AWS Fargate | <p>Amazon ECS ha introdotto il supporto per i processi Fargate con l'autenticazione di registri privati tramite AWS Secrets Manager. Questa funzione permette di archiviare in modo sicuro le proprie credenziali e quindi consultarle nella definizione del container, consentendo quindi alle attività di utilizzare immagini private.</p> <p>Per ulteriori informazioni, consulta <a href="#">Utilizzo di immagini non AWS containerizzate in Amazon ECS</a>.</p> | 10 settembre 2018 |
| Espansione della regione per l'individuazione dei servizi di Amazon ECS                                  | <p>Il servizio di individuazione dei servizi di Amazon ECS ha esteso il supporto per le regioni Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) e Europa (Londra).</p> <p>Per ulteriori informazioni, consulta <a href="#">Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS</a>.</p>                                                                                                          | 30 agosto 2018    |
| Processi pianificati con supporto per i processi Fargate                                                 | <p>Amazon ECS ha introdotto il supporto per i processi pianificati per il tipo di avvio Fargate.</p>                                                                                                                                                                                                                                                                                                                                                                 | 28 agosto 2018    |
| Autenticazione del registro privato tramite supporto AWS Secrets Manager                                 | <p>Amazon ECS ha introdotto il supporto per l'autenticazione di registri privati tramite AWS Secrets Manager. Questa funzione permette di archiviare in modo sicuro le proprie credenziali e quindi consultarle nella definizione del container, consentendo quindi alle attività di utilizzare immagini private.</p> <p>Per ulteriori informazioni, consulta <a href="#">Utilizzo di immagini non AWS containerizzate in Amazon ECS</a>.</p>                        | 16 agosto 2018    |

| Modifica                                | Descrizione                                                                                                                                                                                                                  | Data           |
|-----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Aggiunta del supporto dei volumi Docker | <p>Amazon ECS ha introdotto il supporto per i volumi Docker.</p> <p>Per ulteriori informazioni, consulta <a href="#">Opzioni di storage per le attività di Amazon ECS</a>.</p>                                               | 9 agosto 2018  |
| AWS Espansione della regione di Fargate | <p>AWS Fargate con Amazon ECS si è estesa alle regioni Europa (Francoforte), Asia Pacifico (Singapore) e Asia Pacifico (Sydney).</p> <p>Per ulteriori informazioni, consulta <a href="#">AWS Fargate per Amazon ECS</a>.</p> | 19 luglio 2018 |

| Modifica                                                                                                                                                                               | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Strategie del pianificatore del servizio Amazon ECS aggiunte                                                                                                                           | <p>Amazon ECS ha introdotto il concetto di strategie del pianificatore di servizi.</p> <p>Sono disponibili due strategie del pianificatore del servizio:</p> <ul style="list-style-type: none"><li>• <b>REPLICA</b>: la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le decisioni riguardo al posizionamento delle attività. Per ulteriori informazioni, consulta <a href="#">Strategia di replica</a>.</li><li>• <b>DAEMON</b>: la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Quando si utilizza questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta <a href="#">Strategia daemon</a>.</li></ul> | 12 giugno 2018 |
| <div> <b>Note</b><br/>I processi Fargate non supportano la strategia di pianificazione DAEMON</div> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |
| .                                                                                                                                                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |

| Modifica                                                                                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Data           |
|---------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Agente del container Amazon ECS v1.18.0                                                                                               | <p>Rilascio della nuova versione dell'agente del container Amazon ECS, con aggiunta della seguente funzionalità:</p> <ul style="list-style-type: none"> <li>• Aggiunto il supporto per la personalizzazione del comportamento pull dell'immagine dell'agente del container utilizzando il parametro <code>ECS_IMAGE_PULL_BEHAVIOR</code>. Per ulteriori informazioni, consulta <a href="#">Configurazione dell'agente del container Amazon ECS</a>.</li> </ul> <p><a href="#">Per ulteriori informazioni, consulta <code>amazon-ecs-agent</code> github.</a></p> | 24 maggio 2018 |
| Aggiunto il supporto per le modalità di rete <code>bridge</code> e <code>host</code> quando si configura l'individuazione dei servizi | <p>Aggiunto il supporto per la configurazione dell'individuazione dei servizi per i servizi Amazon ECS tramite le definizioni di processi che specificano le modalità di rete <code>bridge</code> o <code>host</code>. Per ulteriori informazioni, consulta <a href="#">Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS</a>.</p>                                                                                                                                                                                                        | 22 maggio 2018 |
| Aggiunto il supporto per altri parametri di metadati dell'AMI ottimizzata per Amazon ECS                                              | <p>Aggiunti dei parametri secondari che consentono di recuperare in modo sistematico l'ID dell'AMI ottimizzata per Amazon ECS, il nome dell'immagine, il sistema operativo, la versione dell'agente del container e la versione di runtime. Esegui una query dei metadati utilizzando l'API Archivio parametri di Systems Manager. Per ulteriori informazioni, consulta <a href="#">Recupero di metadati AMI Linux ottimizzati per Amazon ECS</a>.</p>                                                                                                           | 9 maggio 2018  |

| Modifica                                                  | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data           |
|-----------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| AWS Espansione della regione di Fargate                   | <p>AWS Fargate con Amazon ECS si è estesa alle regioni Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon) e UE occidentale (Irlanda).</p> <p>Per ulteriori informazioni, consulta <a href="#">AWS Fargate per Amazon ECS</a>.</p>                                                                                                                                                                                                                                                                                                                       | 26 Aprile 2018 |
| Recupero dei metadati dell'AMI ottimizzata per Amazon ECS | <p>Aggiunta la possibilità di recuperare a livello di programmazione i metadati dell'AMI ottimizzata per Amazon ECS tramite l'API Archivio parametri di Systems Manager. Per ulteriori informazioni, consulta <a href="#">Recupero di metadati AMI Linux ottimizzati per Amazon ECS</a>.</p>                                                                                                                                                                                                                                                                        | 10 aprile 2018 |
| AWS Versione della piattaforma Fargate                    | <p>Rilasciata la nuova versione della piattaforma AWS Fargate, che contiene:</p> <ul style="list-style-type: none"> <li>• Aggiunta del supporto per <a href="#">Monitora i carichi di lavoro utilizzando i metadati Amazon ECS</a>.</li> <li>• Aggiunta del supporto per <a href="#">Controllo dello stato</a>.</li> <li>• Aggiunto supporto per <a href="#">Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS</a></li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Versioni della piattaforma Fargate per Amazon ECS</a>.</p> | 26 marzo 2018  |
| Individuazione dei servizi di Amazon ECS                  | <p>Aggiunta dell'integrazione con Route 53 per supportar e l'individuazione dei servizi di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS</a>.</p>                                                                                                                                                                                                                                                                                                                           | 22 marzo 2018  |

| Modifica                                                                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Data             |
|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per i parametri Docker shm-size e tmpfs                                    | <p>Aggiunto il supporto per i parametri Docker shm-size e tmpfs nelle definizioni di processo di Amazon ECS.</p> <p>Per ulteriori informazioni sulla sintassi della CLI ECS aggiornata, consulta <a href="#">Parametri Linux</a>.</p>                                                                                                                                                                                                                                                               | 20 marzo 2018    |
| Controlli dell'integrità dei container                                              | <p>Aggiunta del supporto per i controlli dello stato Docker nelle definizioni dei container. Per ulteriori informazioni, consulta <a href="#">Controllo dello stato</a>.</p>                                                                                                                                                                                                                                                                                                                        | 8 marzo 2018     |
| AWS Fargate                                                                         | <p>È stata aggiunta una panoramica per Amazon ECS con AWS Fargate. Per ulteriori informazioni, consulta <a href="#">AWS Fargate per Amazon ECS</a>.</p>                                                                                                                                                                                                                                                                                                                                             | 22 febbraio 2018 |
| Endpoint di metadati dei processi Amazon ECS                                        | <p>A partire dalla versione 1.17.0 dell'agente del container Amazon ECS, sono disponibili diversi metadati dei processi e <a href="#">statistiche Docker</a> per i processi che utilizzano la modalità di rete awsvpc in un endpoint HTTP fornito dall'agente del container Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Monitora i carichi di lavoro utilizzando i metadati Amazon ECS</a>.</p>                                                                                    | 8 febbraio 2018  |
| Auto Scaling del servizio Amazon ECS tramite policy di monitoraggio degli obiettivi | <p>Aggiunta del supporto per la funzionalità Auto Scaling del servizio ECS con utilizzo di policy di monitoraggio degli obiettivi nella console Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Usa una metrica target per scalare i servizi Amazon ECS</a>.</p> <p>Rimozione del tutorial precedente per il dimensionamento per fasi nella procedura guidata per la prima esecuzione di ECS. Il tutorial è stato sostituito con il nuovo tutorial per il monitoraggio dei target.</p> | 8 febbraio 2018  |
| Supporto Docker 17.09                                                               | <p>Supporto aggiunto per Docker 17.09. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a>.</p>                                                                                                                                                                                                                                                                                                                                                              | 18 gennaio 2018  |

| Modifica                                                                                        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | Data             |
|-------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Nuovo comportamento del pianificatore del servizio                                              | Aggiornamento delle informazioni sul comportam<br>ento per le attività di servizio che non è stato possibile avviare. Nuovo messaggio documentato relativo a un evento di servizio che si attiva quando un'attività di servizio presenta errori consecutivi.                                                                                                                                                                                                                                                                                                                                                                         | 11 gennaio 2018  |
| Periodo di attesa per l'inizializzazione del controllo dell'integrità di Elastic Load Balancing | Aggiunta della possibilità di specificare un periodo di attesa per i controlli dello stato.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 27 dicembre 2017 |
| CPU e memoria a livello di attività                                                             | Aggiunta del supporto per la specifica di CPU e memoria a livello di attività nelle definizioni di processo. Per ulteriori informazioni, consulta <a href="#">TaskDefinition</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 12 dicembre 2017 |
| Ruolo di esecuzione di attività                                                                 | <p>L'agente del container Amazon ECS effettua chiamate alle operazioni API di Amazon ECS per tuo conto, pertanto richiede una policy e un ruolo IAM che consentano al servizio di rilevare che l'agente appartiene a te. Le seguenti operazioni sono coperte dal ruolo di esecuzione delle attività:</p> <ul style="list-style-type: none"> <li>• Chiamate ad Amazon ECR per recuperare l'immagin e del container</li> <li>• Chiamate per CloudWatch archiviare i registri delle applicazioni del contenitore</li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Ruolo IAM di esecuzione di attività Amazon ECS</a>.</p> | 7 dicembre 2017  |
| Disponibilità a livello generale del supporto dei container Windows                             | Aggiunto il supporto per i container Windows Server 2016. Per ulteriori informazioni, consulta <a href="#">Varianti AMI ottimizzate per Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                             | 5 dicembre 2017  |

| Modifica                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Data             |
|-------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| AWS Fargate (Georgia)               | Supporto aggiunto per l'avvio dei servizi Amazon ECS tramite il tipo di avvio Fargate. Per ulteriori informazioni, consulta <a href="#">Tipi di avvio di Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 29 novembre 2017 |
| Modifica del nome Amazon ECS        | Amazon Elastic Container Service viene rinominato (in precedenza Amazon EC2 Container Service).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | 21 Novembre 2017 |
| Reti di attività                    | Le funzionalità di task networking fornite dalla modalità di rete awsvpc conferiscono alle attività di Amazon ECS le stesse proprietà di rete delle istanze Amazon. EC2 Quando utilizzi la modalità di rete awsvpc nelle definizioni di processo, ogni attività avviata da tale definizione di attività ottiene la propria interfaccia di rete elastica, un indirizzo IP privato primario e un hostname DNS interno. La funzionalità di task networking semplifica il networking in container e ti offre un maggiore controllo sul modo in cui le applicazioni containerizzate comunicano tra loro e con gli altri servizi all'interno dell'azienda. VPCs Per ulteriori informazioni, consulta <a href="#">Opzioni di task networking di Amazon ECS per il tipo di EC2 avvio</a> . | 14 Novembre 2017 |
| Metadati di container di Amazon ECS | I contenitori Amazon ECS sono ora in grado di accedere a metadati come il contenitore Docker o l'ID immagine, la configurazione di rete o Amazon. ARNs Per ulteriori informazioni, consulta <a href="#">File di metadati di container di Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          | 2 Novembre 2017  |
| Supporto Docker 17.06               | Supporto aggiunto per Docker 17.06. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 2 Novembre 2017  |



| Modifica                                                                      | Descrizione                                                                                                                                                                                                                                                                                                                                                                           | Data                    |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------|
| Supporto per i flag Docker: dispositivo e init                                | Aggiunta del supporto per il dispositivo Docker e le funzionalità di init nelle definizioni di processo mediante il parametro <code>LinuxParameters</code> ( <code>devices</code> e <code>initProcessEnabled</code> ). Per ulteriori informazioni, consulta <a href="#">LinuxParameters</a> .                                                                                         | 2<br>Novembre<br>2017   |
| Supporto per flag Docker: cap-add e cap-drop                                  | Aggiunta del supporto per le funzionalità Docker cap-add e cap-drop nelle definizioni di processo mediante il parametro <code>LinuxParameters</code> ( <code>capabilities</code> ). Per ulteriori informazioni, consulta <a href="#">LinuxParameters</a> .                                                                                                                            | 22<br>settembre<br>2017 |
| Supporto per Network Load Balancer                                            | Amazon ECS ha aggiunto il supporto per i Network Load Balancer nella console Amazon ECS.                                                                                                                                                                                                                                                                                              | 7<br>settembre<br>2017  |
| RunTask sostituisce                                                           | Aggiunta del supporto per le sostituzioni della definizione di attività quando si esegue un'attività. Ciò consente di eseguire un'attività mentre si modifica una definizione di attività, senza necessità di creare una nuova revisione della definizione di attività. Per ulteriori informazioni, consulta <a href="#">Esecuzione di un'applicazione come attività Amazon ECS</a> . | 27 giugno<br>2017       |
| Processi pianificati di Amazon ECS                                            | Aggiunta del supporto per le attività di pianificazione tramite cron.                                                                                                                                                                                                                                                                                                                 | 7 giugno<br>2017        |
| Istanze Spot nella console Amazon ECS                                         | Aggiunto il supporto per la creazione della serie di istanze Spot del container all'interno della console Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Avvio di un'istanza di container Linux di Amazon ECS</a> .                                                                                                                                                     | 6 giugno<br>2017        |
| Notifica di Amazon SNS per nuove versioni dell'AMI ottimizzata per Amazon ECS | Aggiunta la possibilità di sottoscrivere le notifiche SNS relative alle nuove versioni dell'AMI ottimizzata per Amazon ECS.                                                                                                                                                                                                                                                           | 23 marzo<br>2017        |

| Modifica                                             | Descrizione                                                                                                                                                                                                                                                                                                      | Data             |
|------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Microservizi e attività in batch                     | Aggiunta la documentazione per due casi di utilizzo comune per Amazon ECS: microservizi e processi batch. Per ulteriori informazioni, consulta <a href="#">Informazioni relative ad Amazon ECS</a> .                                                                                                             | Febbraio 2017    |
| Esaurimento dell'istanza di container                | Aggiunta del supporto per l'esaurimento dell'istanza di container, che fornisce un metodo per la rimozione di istanze di container da un cluster. Per ulteriori informazioni, consulta <a href="#">Drenaggio delle istanze di container Amazon ECS</a> .                                                         | 24 gennaio 2017  |
| Supporto Docker 1.12                                 | Supporto aggiunto per Docker 1.12. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                  | 24 gennaio 2017  |
| Nuove strategie per il posizionamento delle attività | Aggiunta del supporto per strategie di posizionamento delle attività: posizionamento basato su attributi, bin packing, distribuzione per zona di disponibilità e per ogni host. Per ulteriori informazioni, consulta <a href="#">Usa strategie per definire il posizionamento delle attività di Amazon ECS</a> . | 29 dicembre 2016 |
| Supporto del container Windows in versione beta      | Aggiunto il supporto per i container Windows 2016 (beta). Per ulteriori informazioni, consulta <a href="#">Varianti AMI ottimizzate per Amazon ECS</a> .                                                                                                                                                         | 20 dicembre 2016 |
| Supporto per Blox OSS                                | Aggiunta del supporto per Blox OSS, che consente l'utilizzo di pianificatori di attività personalizzate. Per ulteriori informazioni, consulta <a href="#">Pianifica i tuoi contenuti su Amazon ECS</a> .                                                                                                         | 1 dicembre 2016  |
| Flusso di eventi Amazon ECS per CloudWatch eventi    | Amazon ECS ora invia le modifiche allo stato dell'istanza del contenitore e dell'attività a CloudWatch Events. Per ulteriori informazioni, consulta <a href="#">Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge</a> .                                                                  | 21 novembre 2016 |

| Modifica                                                                   | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                       | Data              |
|----------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Registrazione dei container Amazon ECS su Logs CloudWatch                  | È stato aggiunto il supporto per il driver awslogs per inviare flussi di log dei container a Logs. CloudWatch. Per ulteriori informazioni, consulta <a href="#">Invia i log di Amazon ECS a CloudWatch</a> .                                                                                                                                                                                                                                                                      | 12 settembre 2016 |
| Servizi Amazon ECS con supporto Elastic Load Balancing per porte dinamiche | Aggiunta del supporto per un load balancer per supportare più combinazioni di istanza/porta per listener, con aumento della flessibilità per i container. Ora è possibile consentire a Docker di definire in modo dinamico la porta host del container; il pianificatore ECS registra la combinazione istanza/porta nel load balancer. Per ulteriori informazioni, consulta <a href="#">Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS</a> . | 11 agosto 2016    |
| Ruoli IAM per processi Amazon ECS                                          | Aggiunto il supporto per l'associazione di ruoli IAM a un processo. Ciò fornisce autorizzazioni più granulari ai container rispetto a un singolo ruolo per un'intera istanza di container. Per ulteriori informazioni, consulta <a href="#">Ruolo IAM dell'attività Amazon ECS</a> .                                                                                                                                                                                              | 13 luglio 2016    |
| Supporto Docker 1.11                                                       | Supporto aggiunto per Docker 1.11. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                                                                                                                                                                                   | 31 maggio 2016    |
| Scalabilità automatica attività                                            | Amazon ECS ha aggiunto il supporto per il dimensionamento automatico dei processi eseguiti da un servizio. Per ulteriori informazioni, consulta <a href="#">Ridimensiona automaticamente il tuo servizio Amazon ECS</a> .                                                                                                                                                                                                                                                         | 18 maggio 2016    |
| Filtro delle definizioni di processo nella famiglia di attività            | Aggiunta del supporto per il filtro in un elenco di definizioni di processo in base alla famiglia di definizione attività. Per ulteriori informazioni, consulta <a href="#">ListTaskDefinitions</a> .                                                                                                                                                                                                                                                                             | 17 maggio 2016    |

| Modifica                                                                      | Descrizione                                                                                                                                                                                                                                                                                                                                                                                       | Data             |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Container Docker e registrazione dell'agente Amazon ECS                       | Amazon ECS ha aggiunto la possibilità di inviare log di agenti ECS e container Docker da istanze di container a Logs per semplificare CloudWatch la risoluzione dei problemi.                                                                                                                                                                                                                     | 5 maggio 2016    |
| L'AMI ottimizzata per ECS ora supporta Amazon Linux 2016.03.                  | L'AMI ottimizzata per ECS ha aggiunto il supporto per Amazon Linux 2016.03. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                                                          | 5 aprile 2016    |
| Supporto Docker 1.9                                                           | Supporto aggiunto per Docker 1.9. Per ulteriori informazioni, consulta <a href="#">Linux ottimizzato per Amazon ECS AMIs</a> .                                                                                                                                                                                                                                                                    | 22 dicembre 2015 |
| CloudWatch metriche per la prenotazione della CPU e della memoria del cluster | Amazon ECS ha aggiunto CloudWatch parametri personalizzati per la prenotazione di CPU e memoria.                                                                                                                                                                                                                                                                                                  | 22 dicembre 2015 |
| Nuova esperienza per la prima esecuzione di Amazon ECS                        | Nell'esperienza per la prima esecuzione della console Amazon ECS è stata aggiunta la creazione di ruoli con zero clic.                                                                                                                                                                                                                                                                            | 23 novembre 2015 |
| Posizionamento delle attività nelle zone di disponibilità                     | Nel pianificatore del servizio Amazon ECS è stato aggiunto il supporto per il posizionamento dei processi tra le zone di disponibilità.                                                                                                                                                                                                                                                           | 8 ottobre 2015   |
| CloudWatch metriche per cluster e servizi Amazon ECS                          | Amazon ECS ha aggiunto CloudWatch parametri personalizzati per l'utilizzo della CPU e della memoria per ogni istanza di container, servizio e famiglia di definizione delle attività in un cluster. Queste nuove metriche possono essere utilizzate per ridimensionare le istanze di container in un cluster utilizzando i gruppi di Auto Scaling o per creare allarmi personalizzati. CloudWatch | 17 agosto 2015   |

| Modifica                                                                        | Descrizione                                                                                                                                                                                               | Data           |
|---------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Supporto per la porta UDP                                                       | Aggiunto il supporto per porte UDP nelle definizioni di processo.                                                                                                                                         | 7 luglio 2015  |
| Sostituzioni delle variabili di ambiente                                        | È stato aggiunto il supporto deregisterTaskDefinition e l'override delle variabili di ambiente per RunTask.                                                                                               | 18 giugno 2015 |
| Aggiornamenti automatici dell'agente Amazon ECS                                 | Aggiunta della possibilità di visualizzare la versione dell'agente ECS in esecuzione in un'istanza di container. Inoltre, è in grado di aggiornare l'agente ECS da, e SDK. AWS Management Console AWS CLI | 11 giugno 2015 |
| Integrazione del pianificatore del servizio Amazon ECS e Elastic Load Balancing | Aggiunta della possibilità di definire un servizio e associare tale servizio a un load balancer di Elastic Load Balancing.                                                                                | 9 aprile 2015  |
| GA di Amazon ECS                                                                | Disponibilità generale di Amazon ECS nelle regioni Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda).                          | 9 aprile 2015  |

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.