



Guida per gli sviluppatori

Amazon Elastic Container Service



Amazon Elastic Container Service: Guida per gli sviluppatori

Copyright © 2026 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 dei rispettivi proprietari, che possono o meno essere affiliati, collegati o sponsorizzati da Amazon.

Table of Contents

Cos'è Amazon ECS?	1
Terminologia e componenti	1
Funzionalità	3
Provisioning	4
Prezzi	4
Servizi correlati	4
Nozioni di base	6
Configurazione	6
Console di gestione AWS	6
Iscriviti a un Account AWS	7
Crea un utente con accesso amministrativo	7
Crea un cloud privato virtuale	9
Creare un gruppo di sicurezza	10
Crea le credenziali per connetterti alla tua istanza EC2	13
Installa il AWS CLI	14
Fasi successive per l'uso di Amazon ECS	15
Creazione di un'immagine di container	15
Prerequisiti	16
Creazione di un'immagine Docker	18
Invia l'immagine al registro del container di Amazon Elastic	20
Eliminazione	22
Fasi successive	22
Ulteriori informazioni su come creare un'attività per Amazon ECS Managed Instances	22
Prerequisiti	22
Fase 1: creazione di un cluster	23
Fase 2: Creazione di una definizione di attività	24
Fase 3: creazione di un servizio	25
Fase 4: visualizzazione del servizio	26
Fase 5: rimozione	26
Scopri come creare un'attività per Amazon ECS Managed Instances con AWS CLI	27
Prerequisiti	27
Fase 1: creazione di un cluster	28
Fase 2: creazione di un provider di capacità di Amazon ECS Managed Instances	30
Fase 3: configurazione di una strategia predefinita del provider di capacità per il cluster	31

Fase 4: registrazione di una definizione delle attività di Linux	31
Fase 5: elenco delle definizioni delle attività	33
Fase 6: creazione di un servizio	33
Fase 7: elenco dei servizi	34
Fase 8: descrizione del servizio in esecuzione	34
Fase 9: test	35
Fase 10: Pulizia	38
Ulteriori informazioni su come creare un'attività di Linux per Fargate	39
Prerequisiti	39
Fase 1: Creazione del cluster	40
Fase 2: Creazione di una definizione di attività	41
Fase 3: Creazione del servizio	42
Fase 4: visualizzazione del servizio	42
Fase 5: rimozione	43
Ulteriori informazioni su come creare un'attività di Windows per Fargate	43
Prerequisiti	44
Fase 1: creazione di un cluster	44
Fase 2: Registrazione di una definizione di attività di Windows	45
Fase 3: creazione di un servizio con la definizione di attività	46
Fase 4: visualizzazione del servizio	47
Fase 5: pulizia	48
Scopri come creare un'attività Windows per EC2	48
Prerequisiti	48
Fase 1: creazione di un cluster	49
Fase 2: Registrazione di una definizione di attività	51
Fase 3: creazione di un servizio	52
Fase 4: visualizzazione del servizio	53
Fase 5: pulizia	54
Utilizzando il AWS CDK	54
Passaggio 1: configura il tuo progetto AWS CDK	55
Passaggio 2: utilizzare il AWS CDK per definire un server Web containerizzato su Fargate ...	58
Fase 3: Test del server Web	65
Fase 4: pulizia	66
Fasi successive	66
Creazione di risorse utilizzando la CLI di AWS Copilot	67
Installazione della CLI di AWS Copilot	68

Distribuzione di un'applicazione Amazon ECS di esempio utilizzando la AWS CLI Copilot	76
Amazon ECS con CloudFormation	79
Creazione di risorse Amazon ECS utilizzando la console CloudFormation	80
Prerequisiti	80
Fase 1: crea un modello di stack	80
Fase 2: crea uno stack per le risorse Amazon ECS	92
Fase 3: verifica	93
Fase 4: Eliminazione delle risorse	94
Creazione di risorse Amazon ECS utilizzando AWS CLI comandi per CloudFormation	94
Prerequisiti	94
Fase 1: creazione di uno stack	95
Fase 2: verifica della creazione di risorse Amazon ECS	107
Fase 3: pulizia	109
Modelli di esempio CloudFormation	109
Definizioni di processi	110
Fornitori di capacità	122
Cluster	124
Servizi	137
Ruoli IAM per Amazon ECS	211
Best practice	214
Progetta la tua soluzione per Amazon ECS	218
Capacity	219
Endpoint di servizio	220
Rete	220
Accesso alle funzionalità	221
Ruoli IAM	222
Registrazione dei log	222
Best Practice per l'immagine dei container	223
Tipi di lancio e fornitori di capacità	223
Best practice	223
Mutabilità del servizio	224
Utilizzo degli endpoint dual-stack	226
Utilizzo degli endpoint dual-stack (AWS CLI)	227
Utilizzo degli endpoint dual-stack (AWS SDKs)	227
Utilizzo degli endpoint dual-stack dall'API REST	228
Applicazioni in sottoreti condivise, zone locali e zone Wavelength	229

Sottoreti condivise	229
Local Zones	230
Zone Wavelength	231
Amazon Elastic Container Service su AWS Outposts	232
Considerazioni	232
Prerequisiti	232
Panoramica della creazione di cluster su AWS Outposts	233
Best practice per le immagini dei container	235
Best practice per la rete	237
Connetti le applicazioni a Internet	238
Best practice per ricevere connessioni in entrata ad Amazon ECS	242
Le migliori pratiche per la connessione ai servizi AWS	247
Best practice per la connessione dei servizi	250
Le migliori pratiche per i servizi di rete tra AWS account e VPCs	255
AWS servizi per la risoluzione dei problemi di rete	256
Ottimizza i tempi di avvio attività	257
Utilizzo su larga scala	258
Quote di servizio di Amazon ECS e limiti di limitazione delle API	259
Gestisci i problemi di limitazione	264
Best practice di dimensionamento automatico e gestione della capacità	267
Determinazione delle dimensioni dell'attività	268
Ottimizzazione del dimensionamento automatico del servizio	269
Capacità e disponibilità	275
Capacità dei cluster	280
Scelta delle dimensioni delle attività Fargate	281
Accelerazione del dimensionamento automatico dei cluster	281
Accedi alle funzionalità con le impostazioni dell'account	283
Nomi di risorse Amazon (ARNs) e IDs	284
Sequenza temporale formato ARN e ID risorsa	287
Container Insights	287
AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)	289
Autorizzazione all'assegnazione di tag	290
Cronologia dell'autorizzazione all'assegnazione di tag	291
AWS Fargate tempo di attesa per il ritiro dell'attività	292
AWS Fargate ritiri di attività utilizzando finestre di EC2 eventi	293
Incrementa le interfacce di rete per istanze di container Linux	294

Monitoraggio del runtime (GuardDuty integrazione con Amazon)	295
VPC a doppio stack IPv6	295
Modalità driver log predefinita	296
Visualizzazione delle impostazioni dell'account con la console	297
Modifica delle impostazioni dell'account	297
Tornare alle impostazioni predefinite dell'account	299
Gestione delle impostazioni dell'account utilizzando il AWS CLI	299
Ruoli IAM per Amazon ECS	301
Istanze gestite da Amazon ECS	305
Nozioni di base	305
fornitori di capacità	306
Selezione e ottimizzazione delle istanze	306
Definizioni di processi	307
Architettura del sistema operativo e della CPU	307
Funzionalità principali	308
Ruoli IAM	308
Conformità e sicurezza	308
Rete	309
Archiviazione dell'istanza	309
Bilanciamento del carico nel servizio	311
Monitoraggio e osservabilità	311
Ottimizzazione dei prezzi e dei costi	312
Service Quotas	312
Considerazioni sulla migrazione	312
Considerazioni e limitazioni	313
Controlli organizzativi	313
Esempio di SCP per bloccare tutte le istanze AMIs ad eccezione di Amazon ECS Managed Instances AMIs	314
Account AMI Amazon ECS Managed Instances	314
Tipi di Istanze gestite da Amazon ECS	316
Famiglie di istanze delle Istanze gestite da Amazon ECS	316
Metodi di selezione delle istanze	318
Selezione del tipo di istanza specifico	318
Selezione del tipo di istanza basata su attributi	318
Ottimizzazione dei costi	319
Best practice per la selezione delle istanze per le Istanze gestite da Amazon ECS	319

Comportamento delle istanze gestite di Amazon ECS	321
Applicazione di patch nelle Istanze gestite da Amazon ECS	321
Ciclo di vita dell'istanza	321
Finestre degli eventi e pianificazione della manutenzione	322
Sequenza di manutenzione	322
Considerazioni sulla sicurezza per le Istanze gestite da Amazon ECS	308
Modello di sicurezza	323
Funzionalità di sicurezza	323
Conformità e supporto normativo	325
Considerazioni FIPS-140 sulle istanze gestite di Amazon ECS	325
Disattiva FIPS sulle istanze gestite di Amazon ECS	326
Considerazioni relative alla sicurezza	327
Best practice di sicurezza	328
Abilitazione del controllo della crittografia VPC per le istanze gestite di Amazon ECS	328
Prerequisiti	329
Identifica i tipi di istanze compatibili	329
Crea un cluster con supporto per la crittografia VPC	329
Considerazioni	330
Risoluzione dei problemi	330
Ottimizzazione dell'infrastruttura	330
Rilevamento delle istanze inattive	331
Rilevamento delle istanze sottoutilizzate	331
ScaleInAfter	332
Eseguire la migrazione da Fargate alle Istanze gestite da Amazon ECS	333
Considerazioni sulla migrazione	333
Prerequisiti	334
Passaggio 1: aggiornare un cluster per usare le Istanze gestite da Amazon ECS	335
Passaggio 2: aggiornare la definizione dell'attività per disporre della funzionalità delle Istanze gestite da Amazon ECS	335
Passaggio 3: aggiornare il servizio per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS	335
Passaggio 4: eseguire la migrazione delle attività autonome	335
Esegui la migrazione EC2 da istanze gestite di Amazon ECS	336
Considerazioni sulla migrazione	336
Prerequisiti	337
Passaggio 1: aggiornare un cluster per usare le Istanze gestite da Amazon ECS	337

Passaggio 2: aggiornare la definizione dell'attività per disporre della funzionalità delle Istanze gestite da Amazon ECS	337
Passaggio 3: aggiornare il servizio per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS	338
Passaggio 4: eseguire la migrazione delle attività autonome	338
AWS Fargate	339
Procedure guidate	339
fornitori di capacità	340
Definizioni di processi	340
Versioni della piattaforma	340
Bilanciamento del carico nel servizio	341
Parametri di utilizzo	341
Considerazioni sulla sicurezza relative all'utilizzo di Fargate	342
Best practice sulla sicurezza di Fargate	342
Utilizzato AWS KMS per crittografare lo storage temporaneo per Fargate	342
Funzionalità SYS_PTRACE per il tracciamento di syscall del kernel con Fargate	343
Usa Amazon GuardDuty con Fargate Runtime Monitoring	343
Considerazioni relative alla sicurezza di Fargate	343
Versioni della piattaforma Fargate	345
Migrazione alla versione 1.4.0 della piattaforma Linux	346
Log delle modifiche della versione della piattaforma Linux	347
Definizione della versione della piattaforma Linux come obsoleta	349
Log delle modifiche della versione della piattaforma Windows	353
Considerazioni relative ai container Windows su Fargate per Amazon ECS	353
Comportamento dell'estrazione delle immagini dei container Linux sui container Fargate	355
Comportamento dell'estrazione delle immagini dei container Windows sui container Fargate ...	356
Archiviazione temporanea delle attività Fargate	357
Versioni della piattaforma container Linux Fargate	357
Versioni della piattaforma container Windows Fargate	358
Chiavi gestite dal cliente per l'archiviazione temporanea di AWS Fargate	359
Ritiro e manutenzione delle attività	372
Panoramica sull'avviso di ritiro delle attività	373
Posso disabilitare il ritiro delle attività?	377
Posso ricevere notifiche di ritiro delle attività tramite altri AWS servizi?	377
Posso modificare il ritiro di un'attività dopo averla pianificata?	378
In che modo Amazon ECS gestisce le attività che fanno parte di un servizio?	378

Amazon ECS può gestire automaticamente le attività autonome?	379
Risoluzione dei problemi relativi alla disponibilità del servizio durante il ritiro delle attività	379
Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS	379
AWS Regioni di Fargate	385
Contenitori Linux su AWS Fargate	385
Contenitori Windows su AWS Fargate	387
EC2	390
Comportamento dell'immagine del contenitore per istanze esterne EC2 e per istanze esterne .	391
Amazon ECS Anywhere	393
Modalità Amazon ECS Express	394
Vantaggi dei servizi Amazon ECS Express Mode	394
Prezzi	395
Disponibilità	395
Creazione di un servizio in modalità Express	395
Prerequisiti	396
Procedure guidate	396
Personalizzazione	396
Crea il tuo primo servizio Express Mode nella console	397
Procedura	397
Fasi successive	398
Personalizzazione	399
Crea il tuo primo servizio Express Mode utilizzando il AWS CLI	401
Prerequisiti	401
Fase 1: creazione di ruoli IAM	401
Fase 2: Creare la prima applicazione di servizio Express Mode	402
Fase 3: Creare un'applicazione di servizio Express Mode con impostazioni personalizzate .	396
Fase 4: Monitora la distribuzione	404
Fase 5: Accedere all'applicazione	404
Risorse create dai servizi Express Mode	405
Impostazioni predefinite del cluster	406
Impostazioni predefinite per la definizione delle attività	406
Impostazioni predefinite per la definizione del contenitore	407
Impostazioni predefinite di registrazione	407
Impostazioni predefinite del servizio	408
Impostazioni predefinite della configurazione di rete	410
Impostazioni predefinite dei ruoli IAM	411

Impostazioni predefinite di Application Auto Scaling	412
Impostazioni predefinite del target di scalabilità delle applicazioni	412
Impostazioni predefinite dei criteri di scalabilità delle applicazioni	413
Impostazioni predefinite di Application Load Balancer	413
Impostazioni predefinite del gruppo target	414
Condivisione delle risorse e ottimizzazione dei costi	415
Visualizzazione dei dettagli di un servizio in modalità Express	415
Aggiornamento di un servizio in modalità Amazon ECS Express	417
Comportamento di distribuzione	417
Procedura	417
Elimina i servizi in modalità Express	419
Differenza tra l'eliminazione di un servizio Express Mode e di un servizio	420
Procedura	420
Best practice per i servizi in modalità Amazon ECS Express	421
Best practice di sicurezza	421
Ottimizzazione delle prestazioni e del calcolo	423
Migliori pratiche operative	425
Risoluzione dei problemi	426
Problemi relativi alla distribuzione	426
Problemi di connettività	428
Problemi di prestazioni	428
Strumenti di monitoraggio e debug	429
Aggiornamento delle risorse al di fuori della modalità Express	430
Comprensione del modello di responsabilità condivisa	430
Devo fare qualcosa prima di poter sfruttare le funzionalità di Amazon ECS nel mio servizio Express Mode?	431
Esempi di aggiornamenti al di fuori della modalità Express	431
Definizioni di processi	434
Stati delle definizioni di attività	435
Risorse Amazon ECS in grado di bloccare un'eliminazione	436
Progetta l'applicazione	437
Best practice per le dimensioni delle attività	439
Rete di attività per le istanze gestite Amazon ECS	440
Task networking per EC2	446
Attività di rete per Fargate	461
Opzioni di archiviazione per le attività	468

Gestione dello spazio di memoria di swap del container	562
Differenze nella definizione dell'attività per istanze gestite da Amazon ECS	564
Differenze nella definizione delle attività per Fargate	566
Differenze nella definizione delle attività per le EC2 istanze che eseguono Windows	575
Creazione di una definizione di attività attraverso la nuova console	576
Convalida JSON	577
CloudFormation pile	577
Procedura	577
Utilizzo di Amazon Q Developer per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS	609
Prerequisiti	609
Procedura	610
Aggiornamento di una definizione di attività attraverso la console	612
Convalida JSON	613
Procedura	613
Annullamento della registrazione di una revisione di definizione dell'attività con la console	614
CloudFormation pile	577
Procedura	615
Eliminazione di una revisione di definizione di attività attraverso la console	615
Risorse Amazon ECS in grado di bloccare un'eliminazione	436
Procedura	617
Casi d'uso delle definizioni di attività	617
Definizioni di attività per carichi di lavoro GPU	618
Definizioni di attività per carichi di lavoro di transcodifica video	631
Definizioni delle attività per i carichi di lavoro di machine learning di AWS Neuron	644
Definizioni delle attività per le istanze di deep learning	653
Definizioni di attività per carichi di lavoro ARM a 64 bit	656
Invia i log a CloudWatch	658
Inviare i log a un servizio o AWS AWS Partner	662
Utilizzo di immagini non AWS containerizzate	676
Riavviare i singoli container nelle attività	680
Invio di dati sensibili a un container	683
Parametri di definizione dell'attività per Istanze gestite da Amazon ECS	707
Family	707
Capacity	708
Ruolo del processo	708

Ruolo per l'esecuzione del processo	708
Modalità di rete	709
Piattaforma di runtime	710
Dimensioni processo	711
Altri parametri di definizione di attività	712
Definizioni del container	718
Parametri di definizione di attività per Fargate	763
Family	763
Capacity	763
Ruolo del processo	764
Ruolo per l'esecuzione del processo	764
Modalità di rete	765
Piattaforma di runtime	765
Dimensioni processo	766
Definizioni del container	770
Nome dell'acceleratore Elastic Inference (obsoleto)	816
Configurazione del proxy	817
Volumi	819
Tag	826
Altri parametri di definizione di attività	827
Parametri di definizione delle attività per EC2	829
Family	830
Capacity	830
Ruolo del processo	831
Ruolo per l'esecuzione del processo	831
Modalità di rete	831
Piattaforma di runtime	833
Dimensioni processo	834
Definizioni del container	835
Nome dell'acceleratore Elastic Inference (obsoleto)	886
Vincoli di posizionamento delle attività	886
Configurazione del proxy	887
Volumi	889
Tag	896
Altri parametri di definizione di attività	897
Modello di definizione di attività	899

Esempi di definizioni di attività	910
Server Web	911
Driver di log splunk	913
Driver di log fluentd	914
Driver di log gelf	914
Carichi di lavoro su istanze esterne	915
Immagine Amazon ECR e ruolo IAM della definizione di attività	917
Punto di ingresso con comando	917
Dipendenze per i container	918
Volumi nelle definizioni di attività	920
Definizioni di attività di esempio di Windows	920
Cluster	922
Scegliere il tipo di cluster giusto	922
Componenti del cluster	923
Concetti relativi ai cluster	923
fornitori di capacità	925
Cluster per Istanze gestite da Amazon ECS	926
Creazione di un cluster per Istanze gestite da Amazon ECS	927
Aggiornare un cluster per usare Istanze gestite da Amazon ECS	932
Provider di capacità di Istanze gestite	934
Protezione scalabile delle attività	944
Cluster per Fargate	952
Avvisi di terminazione di Fargate Spot	953
Creazione di un cluster per i carichi di lavoro Fargate	955
Fornitori di capacità Amazon ECS per carichi di lavoro EC2	957
EC2 sicurezza delle istanze dei container	960
Creazione di un cluster per i EC2 carichi di lavoro Amazon	960
Scalabilità automatica del cluster	967
Istanze di EC2 container Amazon	1005
Cluster per istanze esterne	1164
Sistemi operativi e architetture di sistema supportati	1164
Considerazioni	1165
Creazione di un cluster per i carichi di lavoro delle istanze esterne	1169
Registrazione di un'istanza esterna in un cluster Amazon ECS	1171
Annullamento della registrazione di un'istanza esterna	1177
Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS	1182

Aggiornamento di un cluster	1187
Eliminazione di un cluster	1189
Annullamento della registrazione di un'istanza di container	1190
Procedura	1191
Esaurimento dell'istanza di container	1191
Comportamento di svuotamento per i servizi	1192
Comportamento di svuotamento per processi autonomi	1193
Comportamento di drenaggio per le istanze gestite da Amazon ECS	997
Procedura	1194
Modifica del tipo o della dimensione dell'istanza all'esterno del gruppo Auto Scaling	1195
Istanze Amazon EC2 Container	1196
Ciclo di vita	1196
Supporto Docker	1198
AMI ottimizzate per Amazon ECS	1198
Informazioni aggiuntive	1198
Configurazione dell'agente del container	1199
Installazione dell'agente del container Amazon ECS	1202
Parametri di configurazione del log dell'agente container	1208
Configurazione di istanze di container per immagini Docker private	1209
Pulisci attività e immagini	1214
Pianificare i container	1217
Opzioni di calcolo	1219
Ciclo di vita del processo	1220
Stati del ciclo di vita	1221
In che modo Amazon ECS colloca le attività sulle istanze dei container	1223
Istanze gestite da Amazon ECS	1223
EC2	1224
Fargate	1225
Dimensionamento automatico di Istanze gestite da Amazon ECS e posizionamento delle attività	1225
Usare le strategie per definire il posizionamento delle attività	1229
Attività relative al gruppo	1235
Definire quali istanze di container sono utilizzate per le attività	1236
Attività autonome	1247
Flusso di lavoro delle attività	1248
Esecuzione di un'applicazione come attività	1248

Utilizzo di Amazon EventBridge Scheduler per pianificare le attività	1260
Interruzione di un'attività	1267
Servizi	1269
Opzione di calcolo dell'infrastruttura	1270
Scalabilità automatica del servizio	1271
Bilanciamento del carico nel servizio	1271
Servizi di interconnessione	1273
Controller e strategie di implementazione dei servizi	1273
Implementazioni di servizi	1486
Revisioni del servizio	1498
Ribilanciamento della zona di disponibilità	1507
Usare il bilanciamento del carico per distribuire il traffico del servizio	1516
Scalabilità automatica del servizio	1531
Interconnessione dei servizi	1573
Protezione scalabile delle attività	1649
Iniezione di guasti con Amazon ECS e Fargate	1658
Aggiornare i parametri del servizio Amazon ECS	1667
Eliminazione di un servizio	1700
Migrazione di un servizio con un ARN corto	1701
Logica di limitazione del servizio	1707
Parametri di definizione del servizio	1709
Modello di definizione del servizio	1739
Applicazione di tag alle risorse	1747
Assegnazione di tag alle risorse	1748
Applicazione di tag alle risorse durante la creazione	1751
Restrizioni	1752
Tag gestiti da Amazon ECS	1752
Uso dei tag per la risoluzione dei problemi	1753
Aggiunta di tag alle risorse	1754
Tag per le istanze gestite da Amazon ECS	1757
Tag aggiunti da AWS	1757
Tag personalizzati	1758
Aggiungere tag a un'istanza di EC2 contenitore	1759
Aggiunta di tag a un'istanza di container esterna	1761
Report di utilizzo di	1761
Costi e utilizzo a livello di attività	1762

Monitoraggio	1764
Best practice per il monitoraggio di Amazon ECS	1765
Strumenti di monitoraggio	1765
Strumenti automatici	1765
Strumenti manuali	1767
Monitora Amazon ECS utilizzando CloudWatch	1768
Considerazioni	1769
Parametri consigliati	1769
Visualizzazione dei parametri Amazon ECS	1770
Metriche di Amazon ECS CloudWatch	1772
AWS Fargate metriche di utilizzo	1800
Parametri di prenotazione dei cluster di Amazon ECS	1802
Parametri di utilizzo dei cluster di Amazon ECS	1804
Parametri di utilizzo del servizio Amazon ECS	1806
Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge	1810
Eventi Amazon ECS	1811
Gestione degli eventi	1835
Monitora i container di Amazon ECS utilizzando Container Insights con osservabilità migliorata.	1839
Parametri aggiuntivi per le istanze gestite Amazon ECS	1840
Monitoraggio delle istanze gestite da Amazon ECS	1841
Monitoraggio di Container Insights	1841
Monitoraggio delle istanze	1842
Determina lo stato delle attività utilizzando i controlli dell'integrità dei container	1844
Come viene determinata l'integrità delle attività	1846
Controllo dell'integrità e disconnessioni degli agenti	1847
Visualizzazione dell'integrità dei container	1847
Monitora l'integrità delle istanze di container Amazon ECS	1848
Problemi di integrità delle istanze di container	1849
Individua le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni	1849
Autorizzazioni IAM richieste per AWS Distro per l'integrazione con OpenTelemetry AWS X- Ray	1849
Specificando il AWS Distro for OpenTelemetry sidecar da integrare nella definizione dell'attività AWS X-Ray	1850

Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni	1852
Esportazione dei parametri delle applicazioni su Amazon CloudWatch	1852
Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus	1857
Registrazione delle chiamate API Amazon ECS tramite AWS CloudTrail	1860
Eventi di gestione di Amazon ECS in CloudTrail	1862
Eventi relativi ai dati di Amazon ECS in CloudTrail	1862
Esempi di eventi Amazon ECS	1863
Visualizzazione dei CloudWatch log Live Tail per i servizi e le attività di Amazon ECS	1865
Autorizzazioni richieste	1865
Procedura	1866
AWS CLI riferimenti	1867
Monitora i carichi di lavoro utilizzando i metadati	1867
Variabili di ambiente	1868
File di metadati del container	1869
Metadati delle attività per le attività sulle istanze gestite da Amazon ECS	1875
Metadati delle attività disponibili per le attività di Amazon ECS su EC2	1887
Metadati delle attività disponibili per le attività su Fargate	1931
Introspezione del container	1955
Identifica i comportamenti non autorizzati utilizzando il Monitoraggio del runtime	1958
Come funziona il Monitoraggio del runtime con Amazon ECS	1959
Considerazioni	1960
Utilizzo delle risorse	1960
Monitoraggio del runtime per carichi di lavoro Fargate	1960
Monitoraggio del runtime per carichi di lavoro EC2	1964
Risoluzione di problemi di Monitoraggio del runtime	1969
Monitora i container Amazon ECS con ECS Exec	1973
Considerazioni	1973
Architecture	1976
Configurazione di ECS Exec	1976
Registrazione tramite ECS Exec	1978
Utilizzo delle policy IAM per limitare l'accesso a ECS Exec	1983
Risoluzione dei problemi relativi a ECS Exec	1987
Esecuzione di comandi tramite ECS Exec	1987
Suggerimenti sul Compute Optimizer	1989
Suggerimenti sulle dimensioni delle attività per Fargate	1990

Server MCP Amazon ECS	1991
Panoramica	1991
Nozioni di base	1992
Nozioni di base	1992
Prerequisiti	1992
Configurazione	1993
Configurazioni e best practice comuni	1998
Configurazioni degli strumenti	2000
Configurazioni degli strumenti	2000
Variabili di ambiente	2000
Arguments (Argomenti)	2000
Tools (Strumenti)	2001
Risoluzione dei problemi	2006
Risolvi gli errori relativi alle attività interrotte	2010
Interruzione dell'aggiornamento dei messaggi di errore relativi alle attività	2011
Visualizzazione degli errori delle attività interrotte	2016
Messaggi di errore relativi alle attività interrotti	2018
Verifica della connettività delle attività	2038
Visualizzare le richieste di ruolo IAM	2043
Visualizzare i messaggi relativi agli eventi del servizio	2044
Messaggi relativi agli eventi del servizio Amazon ECS	2046
Messaggi degli eventi relativi a malfunzionamenti del servizio Amazon ECS	2059
Messaggi relativi agli eventi del servizio di ribilanciamento del servizio della zona di disponibilità di Amazon ECS	2060
Risoluzione dei problemi dei bilanciatori del carico per il servizio in Amazon ECS	2062
Risoluzione dei problemi di dimensionamento automatico del servizio in Amazon ECS	2064
Acquisizione dell'evento	2064
Come funziona	2065
Event types (Tipi di evento)	2065
Configurazione dei gruppi di log	2066
Considerazioni	2066
Risoluzione dei problemi basata sugli eventi	2067
Attiva l'acquisizione degli eventi per un cluster esistente	2067
Visualizzazione degli eventi di modifica dello stato dei servizi e delle attività	2068
Risoluzione degli errori di CPU o memoria non valide nella definizione delle attività	2070
Visualizzare i log dell'agente container	2073

Raccolta log dei container con il raccoglitore di log di Amazon ECS	2074
Introspezione dell'agente container	2077
Diagnostica Docker in Amazon ECS	2080
Elenca i container Docker in Amazon ECS	2080
Visualizza i log Docker in Amazon ECS	2081
Ispeziona i container Docker in Amazon ECS	2082
Configurazione dell'output dettagliato dal daemon Docker in Amazon ECS	2083
Risolvi i problemi relativi a API error (500): devmapper di Docker in Amazon ECS	2085
Risoluzione dei problemi di ECS Exec	2086
Verifica tramite Exec Checker	2086
Errore durante la chiamata a execute-command	2086
Risoluzione dei problemi relativi ad Amazon ECS Anywhere	2087
Problemi di registrazione delle istanze esterne	2087
Problemi di rete delle istanze esterne	2088
Problema di esecuzione dei processi	2088
Risoluzione dei problemi relativi al caricamento delle classi Java su Fargate	2089
Caratteristiche	2089
Cause	2089
Risoluzione	2090
Prevenzione	2091
AWS Fargate limitazione delle quote	2091
Limitazione dell'API RunTask in Fargate	2092
Adeguamento delle quote di velocità in Fargate	2093
Risoluzione dei problemi relativi alle istanze gestite da Amazon ECS	2093
Prerequisiti	2093
Scenari per la risoluzione dei problemi comuni	2093
Analisi dei log nelle istanze gestite di Amazon ECS	2100
Usa il EC2 AWS CLI per ottenere l'output della console da un'istanza gestita di Amazon ECS	2101
Pulizia	2101
Risorse aggiuntive	2018
Risoluzione dei problemi relativi alle istanze gestite da Amazon ECS	2102
Errori delle istanze gestite da Amazon ECS	2106
Motivi degli errori dell'API	2110
Risoluzione dei problemi con Amazon Q Developer	2120
Autorizzazioni richieste	2120

Procedura	2121
Considerazioni	2122
Ulteriori informazioni	2122
Sicurezza	2123
Identity and Access Management	2124
Destinatari	2124
Autenticazione con identità	2125
Gestione dell'accesso tramite policy	2126
Come funziona Amazon Elastic Container Service con IAM	2128
Esempi di policy basate su identità	2138
AWS politiche gestite per Amazon ECS	2151
Uso di ruoli collegati ai servizi	2171
Ruoli IAM per Amazon ECS	2178
Autorizzazioni necessarie per la console Amazon ECS	2246
Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green	2257
Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi Amazon ECS ..	2258
Assegnazione di tag alle risorse al momento della creazione	2259
Risoluzione dei problemi	2264
Best practice di IAM	2266
Registrazione di log e monitoraggio	2269
Convalida della conformità	2271
Best practice di conformità e sicurezza	2271
AWS Fargate Conformità FIPS-140	2273
AWS Fargate Considerazioni sullo standard FIPS-140	2273
Uso di FIPS su Fargate	2274
Uso CloudTrail per il controllo FIPS-140 di Fargate	2274
Sicurezza dell'infrastruttura	2276
Endpoint VPC di interfaccia (AWS PrivateLink)	2277
Modello di responsabilità condivisa	2284
Fargate	2284
EC2	2285
Responsabilità condivisa delle istanze gestite da Amazon ECS	2286
AWS responsabilità	2286
Responsabilità del cliente	2287
Best practice sulla sicurezza di rete	2288
Crittografia dei dati in transito	2288

Reti di attività	2289
AWS PrivateLink e Amazon ECS	2290
Impostazioni dell'agente di container	2291
Raccomandazioni sulla sicurezza di rete	2291
Best practice per la sicurezza di attività e container	2293
Creazione di immagini di base o uso di immagini distroless	2293
Scansione delle immagini per individuare eventuali vulnerabilità	2294
Rimozione delle autorizzazioni speciali dalle immagini	2295
Creazione di un set di immagini curate	2296
Scansione dei pacchetti e delle librerie di applicazioni per individuare eventuali vulnerabilità	2294
Esecuzione dell'analisi statica del codice	2296
Esecuzione di container come utente non root	2297
Uso di un file system root di sola lettura	2297
Configura attività con limiti di CPU e memoria (Amazon EC2)	2297
Uso di tag immutabili con Amazon ECS	2298
Evita di utilizzare i container come privilegiati (Amazon EC2)	2298
Rimozione delle funzionalità di Linux non necessarie dal container	2299
Uso di una chiave gestita dal cliente (CMK) per la crittografia delle immagini inviate ad Amazon ECR	2299
Esercitazioni	2300
Creazione di un'attività Linux per Fargate con AWS CLI	2302
Prerequisiti	2303
Fase 1: creare un cluster	2304
Fase 2: Registra una definizione dei processi Linux	2305
Fase 3: Elenca le definizioni di attività	2306
Fase 4: Crea un servizio	2307
Fase 5: Elenca i servizi	2307
Fase 6: Descrivi il servizio in esecuzione	2308
Fase 7: Test	2310
Fase 8: elimina	2314
Creazione di un'attività Windows per Fargate con il AWS CLI	2314
Prerequisiti	2315
Fase 1: creare un cluster	2316
Fase 2: Registrazione di una definizione di attività di Windows	2317
Fase 3: Elenca le definizioni di attività	2318

Fase 4: Crea un servizio	2318
Fase 5: Elenca i servizi	2319
Fase 6: Descrivi il servizio in esecuzione	2319
Fase 7: eliminare	2322
Creazione di un'attività per EC2 con AWS CLI	2322
Prerequisiti	2323
Creazione di un cluster	2323
Avvio di un'istanza di container con l'AMI Amazon ECS	2324
Elencare le istanze di container	2327
Descrivere l'istanza di container	2327
Registrare una definizione di attività	2330
Elencare le definizioni di attività	2332
Creazione di un servizio	2333
Elencare i servizi	2333
Descrivere il servizio	2334
Descrivere l'attività in esecuzione	2335
Testare il server web	2336
Eseguire la pulizia delle risorse	2337
Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events	2338
Prerequisito: configurazione di un cluster di verifica	2338
Fase 1: Creazione della funzione Lambda	2339
Fase 2: Registrazione di una regola di evento	2339
Fase 3: creazione di una definizione di attività	2340
Fase 4: Test della regola	2342
Invio di avvisi di Amazon Simple Notification Service per gli eventi di attività arrestate	2342
Prerequisito: configurazione di un cluster di verifica	2343
Prerequisito: configurazione delle autorizzazioni per Amazon SNS	2343
Fase 1: Creazione e sottoscrizione a un argomento Amazon SNS	2343
Fase 2: Registrazione di una regola di evento	2344
Fase 3: Test del tuo articolo	2345
Concatenazione di messaggi di log multilinea o di traccia dello stack	2346
Autorizzazioni IAM richieste	2347
Determinazione dei casi in cui utilizzare l'impostazione del log multilinea	2347
Opzioni di analisi e concatenazione	2349
Implementazione di Fluent Bit su container Windows	2368
Prerequisiti	2371

Fase 1: Creazione dei ruoli di accesso IAM	2372
Fase 2: Creazione di un'istanza di container Windows di Amazon ECS	2373
Fase 3: Configurazione di Fluent Bit	2374
Passaggio 4: Registrare una definizione di attività Windows Fluent Bit che indirizza i log a CloudWatch	2376
Fase 5: Esecuzione della definizione di attività <code>ecs-windows-fluent-bit</code> come servizio Amazon ECS utilizzando la strategia di pianificazione del daemon	2378
Fase 6: Registrazione di una definizione di attività di Windows che genera i log	2379
Fase 7: Esecuzione della definizione di attività <code>windows-app-task</code>	2381
Fase 8: Verificare gli accessi CloudWatch	2381
Fase 9: Pulizia	2382
Utilizzo gMSA per contenitori EC2 Linux	2383
Considerazioni	2384
Prerequisiti	2385
Configurazione	2386
CredSpec file	2393
Utilizzo del gMSA per i container Linux su Fargate	2394
Considerazioni	2394
Prerequisiti	2395
Configurazione	2395
CredSpec file	2398
Utilizzo di contenitori Windows con modalità <code>domainless</code> utilizzando il gMSA AWS CLI	2400
Prerequisiti	2401
Fase 1: creazione e configurazione dell'account gMSA su Active Directory Domain Services (AD DS)	2402
Fase 2: caricamento delle credenziali in Secrets Manager	2405
Fase3: modifica del codice JSON CredSpec per includere informazioni relative a gMSA senza dominio	2405
Fase 4: caricamento di CredSpec su Amazon S3	2407
Fase 5: creazione di un cluster Amazon ECS (facoltativo)	2407
Fase 6: creazione di un ruolo IAM per le istanze di container	2408
Fase 7: creazione di un ruolo di esecuzione dell'attività personalizzata	2408
Fase 8: creazione di un ruolo dell'attività per Amazon ECS Exec	2409
Fase 9: registrazione di una definizione di attività	2411
Fase 10: registrazione di un'istanza di container Windows	2413
Fase 11: verifica dell'istanza di container	2413

Fase 12: esecuzione di un'attività di Windows	2415
Fase 13: verifica che il container disponga delle credenziali gMSA	2415
Fase 14: pulizia	2416
Debug	2417
Scopri come usare g MSAs per i contenitori EC2 Windows	2418
Considerazioni	2419
Prerequisiti	2420
Configurazione	2421
Utilizzo di Image Builder per creare modelli personalizzati ottimizzati per Amazon ECS AMIs ..	2427
Pulizia delle AMI ottimizzate per Amazon ECS	2428
Utilizzo dell'ARN dell'immagine con infrastructure as code (IaC)	2429
Utilizzo dell'immagine ARN con CloudFormation	2432
Utilizzo dell'ARN dell'immagine con Terraform	2433
Utilizzo dei AWS Deep Learning Containers	2433
Tutorial: creazione di un servizio utilizzando una distribuzione blu/verde	2434
Prerequisiti	2434
Fase 1: Creazione di un Application Load Balancer	2435
Fase 2: creazione di un cluster Amazon ECS	2436
Fase 3: Registra una definizione di attività	2436
Fase 4: creazione di un servizio Amazon ECS	2438
Fase 5: Creazione delle risorse AWS CodeDeploy	2439
Fase 6: creare e monitorare una distribuzione CodeDeploy	2441
Fase 7: eliminare	2445
Service Quotas	2447
Gestione delle quote di servizio in Console di gestione AWS	2447
Gestione delle service quotas e dei limiti di limitazione delle API	2449
Bilanciamento del carico elastico	260
Interfacce di rete elastiche	261
AWS Cloud Map	263
Documentazione di riferimento dell'API Amazon ECS	2455
Cronologia dei documenti	2456
.....	mmdix

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

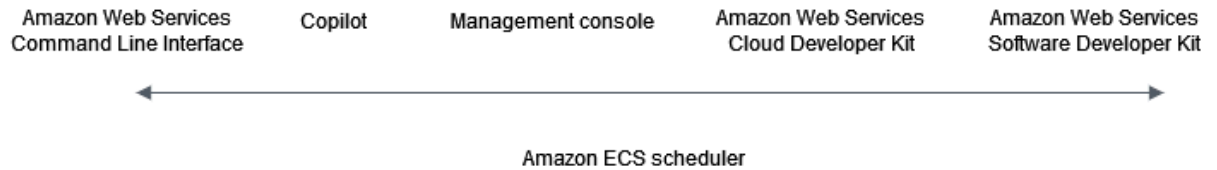
Amazon ECS si compone di tre livelli:

- 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 su cui vengono eseguiti i container. Di seguito è riportata una panoramica delle opzioni di capacità:

- Amazon ECS Managed Instances è un'opzione di elaborazione per Amazon ECS che ti consente di eseguire carichi di lavoro containerizzati su una gamma di tipi di istanze EC2 Amazon, affidando al contempo la gestione dell'infrastruttura a AWS. Con Amazon ECS Managed Instances, puoi accedere a funzionalità di elaborazione specifiche come accelerazione GPU, architetture CPU specifiche, prestazioni di rete elevate e tipi di istanze specializzate, AWS gestendo al contempo il provisioning, l'applicazione di patch, la scalabilità e la manutenzione dell'infrastruttura sottostante.
- EC2 Istanze Amazon nel cloud AWS

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

- Senza server 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 un'istanza esterna, come un server on-premises 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 blueprint per l'applicazione.

Cluster

L'infrastruttura su cui viene eseguita l'applicazione.

Processo

Un'applicazione, come un processo batch, che esegue operazioni 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.

Dimensionamento automatico del servizio

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

Provisioning

Esistono diverse opzioni per il provisioning di Amazon ECS:

- **Console di gestione AWS:** 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.

Prezzi

I prezzi di Amazon ECS dipendono dall'opzione di capacità scelta per i container.

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

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](#)

Monitoraggio dei servizi e delle attività.

[Amazon Elastic Container Registry](#)

Archiviazione e gestione delle immagini di container.

[Elastic Load Balancing](#)

Distribuzione automatica del traffico in entrata del servizio.

[Amazon GuardDuty](#)

Rilevazione dell'uso potenzialmente non autorizzato o dannoso di istanze e carichi di lavoro dei container.

Ulteriori informazioni su come creare e utilizzare le risorse di Amazon ECS

Le seguenti guide forniscono un'introduzione agli strumenti disponibili per accedere ad Amazon ECS e procedure introduttive per l'esecuzione dei 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 Console di gestione AWS 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](#)
- [Ulteriori informazioni su come creare un'attività per Amazon ECS Managed Instances](#)
- [Scopri come creare un'attività per Amazon ECS Managed Instances con AWS CLI](#)
- [Ulteriori informazioni su come creare un'attività di Amazon ECS Linux per Fargate](#)
- [Ulteriori informazioni su come creare un'attività di Amazon ECS Windows per Fargate](#)
- [Scopri come creare un'attività Amazon ECS Windows per EC2](#)
- [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.

Console di gestione AWS

Console di gestione AWS 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 [Ulteriori informazioni su come creare e utilizzare le risorse di 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 Console di gestione AWS, possono gestire facilmente le risorse correlate come i sistemi di bilanciamento del carico e le istanze Amazon EC2 .

Inizia con. Console di gestione AWS

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 o un messaggio di testo e ti verrà chiesto di 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 [Console di gestione AWS](#) 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 [Accedere come utente root](#) nella 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 il Centro identità IAM.

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

2. Nel Centro identità IAM, 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 come utente del Centro identità IAM, utilizza l'URL di accesso che è stato inviato al tuo indirizzo e-mail quando hai creato l'utente del Centro identità IAM.

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.

Assegnazione dell'accesso ad altri utenti

1. Nel Centro identità IAM, crea un set di autorizzazioni conforme alla best practice per l'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, [Creare un gruppo di sicurezza](#). Per determinare se disponi di un VPC predefinito, consulta [Work with your default VPC and default subnets](#) nella Guida per l'utente di Amazon VPC. 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 Guida per l'utente di Amazon VPC e utilizza la tabella seguente per determinare quali opzioni selezionare.

Opzione	Valore	
Risorse da creare	Solo VPC	
Name	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.

Creare 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	
Region	La stessa regione in cui è stata creata la coppia di chiavi.	

Opzione	Valore	
Name	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 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 [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)	

Opzione	Valore	
	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 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 EC2 utilizzarla.

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.

Come 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

Console di gestione AWS 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 sanno 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 [Guida per l'utente AWS Tools for Windows PowerShell](#).

Fasi successive per l'uso 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 del container](#) con Docker e inviala ad Amazon ECR per utilizzarla nelle definizioni delle attività di Amazon ECS.
- [Ulteriori informazioni su come creare un'attività di Amazon ECS Linux per Fargate](#).
- [Ulteriori informazioni su come creare un'attività di Amazon ECS Windows per Fargate](#).
- [Scopri come creare un'attività Amazon ECS Windows per EC2](#).
- 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.

Amazon ECS pianifica le applicazioni containerizzate su istanze di container o su Fargate. AWS Le applicazioni containerizzate sono pacchettizzate come immagini del container. Questo esempio crea un'immagine del container per un server web.

Puoi creare la tua prima immagine Docker, quindi inviarla ad Amazon ECR, che è un registro del container, per utilizzarla nelle definizioni delle attività di Amazon ECS. Questa procedura guidata presuppone una conoscenza di base di Docker e del suo funzionamento. Per ulteriori informazioni su Docker, consulta [What is Docker?](#) e la [documentazione di 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 [Moving an image through its lifecycle in Amazon ECR](#) nella Guida per l'utente di Amazon Elastic Container Registry.
- 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 2023, consulta [Installazione di Docker su 023 AL2](#). Per tutti gli altri sistemi operativi, consulta la documentazione su Docker all'indirizzo [Panoramica di Docker Desktop](#).
- Hai installato e configurato AWS CLI. Per ulteriori informazioni, consulta [Installing or updating to the latest version of the AWS CLI](#) nella Guida per l'utente dell'AWS Command Line Interface.

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 2023 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 delle attività di Amazon ECS utilizzano le immagini dei container per avviare questi ultimi 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 2023 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'istruzione EXPOSE indica che la porta 80 nel container è in ascolto, mentre l'istruzione CMD 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 un sistema basato su ARM, come [Apple Silicon](#), utilizza l'opzione --platform "--platform linux/amd64".

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

4. Elenca l'immagine del container.

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

Output:

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

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 utilizzi 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/>.

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 di immagini AWS 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.

`region` Sostituiscilo, 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, consulta [Installing or updating to the latest version of the AWS CLI](#) nella Guida per l'utente dell'AWS Command Line Interface .

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 [Fasi successive](#). 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
```

Fasi successive

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 container ad Amazon ECR, puoi utilizzare l'immagine in una definizione delle attività. Per ulteriori informazioni, consultare uno dei seguenti argomenti:

- [the section called “Ulteriori informazioni su come creare un'attività di Linux per Fargate”](#)
- [the section called “Ulteriori informazioni su come creare un'attività di Windows per Fargate”](#)
- [Creazione di un'attività Amazon ECS Linux per Fargate con AWS CLI](#)

Ulteriori informazioni su come creare un'attività per Amazon ECS Managed Instances

Ulteriori informazioni su come utilizzare Amazon ECS con Amazon ECS Managed Instances per eseguire un'applicazione containerizzata.

Prerequisiti

Prima di iniziare questo tutorial, completa le seguenti operazioni:

- Esegui i passaggi in [Configurazione dell'utilizzo di Amazon ECS](#).
- Hai completato le fasi descritte in [the section called “Configurazione”](#).
- Disponi dei ruoli IAM richiesti per Istanze gestite da Amazon ECS. Questo include:
 - Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Il nome del ruolo dell'istanza deve essere incluso `ecsInstanceRole` come prefisso in modo che corrisponda all'`iam:PassRole` azione nel ruolo dell'infrastruttura.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine del container ospitata su Amazon ECR Public, quindi le istanze devono avere accesso a internet. Per assegnare alle tue istanze 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 alle istanze.

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: creazione di un cluster

1. [Apri la console Amazon ECS 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, inserisci un nome univoco per il cluster in Nome del cluster.
6. In Infrastruttura, scegli Fargate e Managed. EC2
7. Configura le impostazioni di Managed Instances:
 - a. In Ruolo dell'infrastruttura, seleziona il ruolo IAM che hai creato per la gestione dell'infrastruttura di Managed Instances.

- b. In Profilo dell'istanza, seleziona il `ecsInstanceRole` creato.
 - c. In Attributi dell'istanza, scegli Utilizza i valori predefiniti di ECS.
8. In Rete, configura il VPC e le sottoreti per Managed Instances:
 - a. In VPC, scegli il VPC che ospita Managed Instances.
 - b. In Sottoreti, seleziona una o più sottoreti in cui verrà avviato Managed Instances.
 - c. In Security groups (Gruppi di sicurezza), selezionare uno o più gruppi di sicurezza.
9. (Facoltativo) Per aggiungere tag al tuo cluster, espandi la sezione Tag, quindi configura i tag.
10. Scegli Create (Crea).

Fase 2: Creazione di una definizione di attività

Una definizione di attività è un blueprint dell'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. Per creare una definizione delle attività, segui questi passaggi:

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 il seguente JSON nell'editor per sostituire il JSON precompilato:

Sostituiscilo **account-id** con l'ID del tuo AWS account e **region** con la regione che stai utilizzando.

```
{
  "family": "managed-instance-tutorial",
  "networkMode": "awsvpc",
  "executionRoleArn": "arn:aws:iam::account-id:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "name": "sample-app",
      "image": "public.ecr.aws/docker/library/httpd:latest",
      "essential": true,
      "portMappings": [
        {
          "containerPort": 80,
          "hostPort": 80,
```

```
        "protocol": "tcp"
      }
    ],
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-group": "/ecs/managed-instance-tutorial",
        "awslogs-region": "region",
        "awslogs-stream-prefix": "ecs"
      }
    }
  },
  "requiresCompatibilities": [
    "MANAGED_INSTANCES"
  ],
  "cpu": "1024",
  "memory": "2048"
}
```

4. Scegli Create (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 processo in un cluster Amazon ECS. Segui la procedura indicata di seguito per creare un servizio:

1. Nel pannello di navigazione, scegli Cluster, quindi seleziona il cluster **managed-instance-tutorial**.
2. Nella scheda Servizi, scegliere Crea.
3. Per Task definition family, scegli managed-instance-tutorial.
4. Per Nome servizio, inserisci **managed-instance-tutorial-service**.
5. In Ambiente, scegli la strategia del fornitore di capacità.
6. In Rete, configura le seguenti impostazioni:
 - a. Scegli un VPC esistente o creane uno nuovo.
 - b. Per le sottoreti, scegli le sottoreti da usare.

- c. Per i gruppi di sicurezza, scegli un gruppo di sicurezza esistente o creane uno nuovo che consenta il traffico in entrata sulla porta 80.

7. Scegli Create (Crea).

Fase 4: visualizzazione del servizio

Una volta avviato il servizio, puoi visualizzarlo per ottenere ulteriori informazioni e testarlo.

1. Scegli il servizio **managed-instance-tutorial-service**.
2. Dalla scheda Attività, scegli l'ID dell'attività in esecuzione.
3. In Rete, in IP pubblico, scegli Apri indirizzo.
4. Dovresti vedere la pagina di test del server HTTP Apache, che conferma che il server web funziona correttamente.

Fase 5: rimozione

Una volta terminato questo tutorial, devi pulire le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

1. Nel pannello di navigazione scegliere Cluster.
2. Nella pagina Cluster seleziona il cluster **managed-instance-tutorial**.
3. Scegli la scheda Servizi.
4. Seleziona il servizio **managed-instance-tutorial-service**, quindi scegli Elimina.
5. Al prompt di conferma, inserisci **delete**, quindi scegli Elimina.
6. Dopo aver eliminato il servizio, scegli Cluster nel pannello di navigazione.
7. Nella pagina Cluster, seleziona il cluster **managed-instance-tutorial**, quindi scegli Elimina cluster.
8. Al prompt di conferma, inserisci **delete managed-instance-tutorial**, quindi scegli Elimina.

Scopri come creare un'attività per Amazon ECS Managed Instances con AWS CLI

La procedura seguente illustra come configurare un cluster, creare un provider di capacità, registrare una definizione delle attività, eseguire un'attività di Linux e come eseguire altri scenari comuni in Amazon ECS con l' AWS CLI. Usa la versione più recente dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the 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 [the section called “Utilizzo degli endpoint dual-stack”](#).

Argomenti

- [the section called “Prerequisiti”](#)
- [the section called “Fase 1: creazione di un cluster”](#)
- [the section called “Fase 2: creazione di un provider di capacità di Amazon ECS Managed Instances”](#)
- [the section called “Fase 3: configurazione di una strategia predefinita del provider di capacità per il cluster”](#)
- [the section called “Fase 4: registrazione di una definizione delle attività di Linux”](#)
- [the section called “Fase 5: elenco delle definizioni delle attività”](#)
- [the section called “Fase 6: creazione di un servizio”](#)
- [the section called “Fase 7: elenco dei servizi”](#)
- [the section called “Fase 8: descrizione del servizio in esecuzione”](#)
- [the section called “Fase 9: test”](#)
- [the section called “Fase 10: Pulizia”](#)

Prerequisiti

Prima di iniziare questo tutorial, completa le seguenti operazioni:

- Esegui i passaggi in [Configurazione dell'utilizzo di Amazon ECS](#).
- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, consulta [Installazione o aggiornamento all'ultima versione della AWS CLI nella Guida per l'utente](#).AWS Command Line Interface
- Hai completato le fasi descritte in [the section called “Configurazione”](#).
- Disponi dei ruoli IAM richiesti per Istanze gestite da Amazon ECS. Questo include:
 - Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Il nome del ruolo dell'istanza deve essere incluso `ecsInstanceRole` come prefisso in modo che corrisponda all'`iam:PassRole`azione nel ruolo dell'infrastruttura.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine del container ospitata su Amazon ECR Public, quindi le istanze devono avere accesso a internet. Per assegnare alle tue istanze 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 alle istanze.

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: creazione di 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 managed-instances-cluster
```

Output:

```
{
  "cluster": {
    "status": "ACTIVE",
    "defaultCapacityProviderStrategy": [],
    "statistics": [],
    "capacityProviders": [],
    "tags": [],
    "clusterName": "managed-instances-cluster",
    "settings": [
      {
        "name": "containerInsights",
        "value": "disabled"
      }
    ],
    "registeredContainerInstancesCount": 0,
    "pendingTasksCount": 0,
    "runningTasksCount": 0,
    "activeServicesCount": 0,
    "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/managed-instances-cluster"
  }
}
```

Fase 2: creazione di un provider di capacità di Amazon ECS Managed Instances

Prima di poter eseguire attività utilizzando Amazon ECS Managed Instances, devi creare un provider di capacità che definisca la configurazione dell'infrastruttura. Il provider di capacità specifica i ruoli IAM, la configurazione di rete e altre impostazioni per le istanze gestite.

Crea un file JSON con la configurazione del provider di capacità. Sostituisci i valori segnaposto con gli identificatori delle risorse effettivi:

```
{
  "name": "managed-instances-cp",
  "cluster": "managed-instances-cluster",
  "managedInstancesProvider": {
    "infrastructureRoleArn":
"arn:aws:iam::aws_account_id:role/ecsInfrastructureRole",
    "instanceLaunchTemplate": {
      "ec2InstanceProfileArn": "arn:aws:iam::aws_account_id:instance-
profile/ecsInstanceRole",
      "networkConfiguration": {
        "subnets": [
          "subnet-abcdef01234567890",
          "subnet-1234567890abcdef0"
        ],
        "securityGroups": [
          "sg-0123456789abcdef0"
        ]
      },
      "storageConfiguration": {
        "storageSizeGiB": 100
      },
      "monitoring": "basic"
    }
  }
}
```

Salva questa configurazione come `managed-instances-cp.json` e crea il provider di capacità:

```
aws ecs create-capacity-provider --cli-input-json file://managed-instances-cp.json
```

Il comando restituisce una descrizione del provider di capacità dopo che ne ha completato la creazione.

Fase 3: configurazione di una strategia predefinita del provider di capacità per il cluster

Aggiorna il cluster per utilizzare il provider di capacità Amazon ECS Managed Instances come strategia predefinita per il provider di capacità. Ciò consente alle attività e ai servizi di utilizzare automaticamente Amazon ECS Managed Instances senza specificare esplicitamente il provider di capacità.

Crea un file JSON con la configurazione del provider di capacità del cluster:

```
{
  "cluster": "managed-instances-cluster",
  "capacityProviders": [
    "managed-instances-cp"
  ],
  "defaultCapacityProviderStrategy": [
    {
      "capacityProvider": "managed-instances-cp",
      "weight": 1
    }
  ]
}
```

Salva questa configurazione come `cluster-cp-strategy.json` e aggiorna il cluster:

```
aws ecs put-cluster-capacity-providers --cli-input-json file://cluster-cp-strategy.json
```

Fase 4: registrazione di una definizione delle attività di Linux

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 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 [the section called "Parametri di definizione di attività per Fargate"](#).

```
{
```

```

"family": "sample-managed-instances",
"networkMode": "awsvpc",
"containerDefinitions": [
  {
    "name": "managed-instances-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 using Amazon ECS Managed Instances.</p></div></body></html>' > /usr/local/apache2/htdocs/index.html && httpd-foreground\""
    ]
  }
],
"requiresCompatibilities": [
  "MANAGED_INSTANCES"
],
"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/managed-instances-task.json
```

Il comando `register-task-definition` restituisce una descrizione della definizione di attività una volta completata la registrazione.

Fase 5: elenco delle definizioni delle 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-managed-instances:1"
  ]
}
```

Fase 6: creazione di 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-managed-instances: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 managed-instances-cluster --service-name managed-
instances-service --task-definition sample-managed-instances:1 --desired-
count 1 --network-configuration "awsvpcConfiguration={subnets=[subnet-
abcd1234],securityGroups=[sg-abcd1234]}"
```

Esempio di utilizzo di una sottorete pubblica:

```
aws ecs create-service --cluster managed-instances-cluster --service-name managed-
instances-service --task-definition sample-managed-instances:1 --desired-
```

```
count 1 --network-configuration "awsvpcConfiguration={subnets=[subnet-  
abcd1234],securityGroups=[sg-abcd1234],assignPublicIp=ENABLED}"
```

Il comando `create-service` restituisce una descrizione del servizio dopo il completamento della creazione.

Fase 7: elenco dei 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 managed-instances-cluster
```

Output:

```
{  
  "serviceArns": [  
    "arn:aws:ecs:region:aws_account_id:service/managed-instances-cluster/managed-  
instances-service"  
  ]  
}
```

Fase 8: descrizione del 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 managed-instances-cluster --services managed-  
instances-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 `Errors`, sono disponibili informazioni sugli eventuali errori associati alla chiamata.

L'output mostrerà che il servizio utilizza il provider di capacità Amazon ECS Managed Instances:

```
{
```

```

"services": [
  {
    "capacityProviderStrategy": [
      {
        "capacityProvider": "managed-instances-cp",
        "weight": 1,
        "base": 0
      }
    ],
    "networkConfiguration": {
      "awsvpcConfiguration": {
        "subnets": [
          "subnet-abcd1234"
        ],
        "securityGroups": [
          "sg-abcd1234"
        ],
        "assignPublicIp": "ENABLED"
      }
    },
    "enableECSTags": false,
    "loadBalancers": [],
    "deploymentController": {
      "type": "ECS"
    },
    "desiredCount": 1,
    "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/managed-instances-cluster",
    "serviceArn": "arn:aws:ecs:region:aws_account_id:service/managed-instances-service",
    "serviceName": "managed-instances-service",
    "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/sample-managed-instances:1"
  },
  {
    "failures": []
  }
]

```

Fase 9: test

Per testare l'implementazione, devi trovare l'indirizzo IP pubblico dell'istanza gestita che esegue l'attività.

Test di un'attività implementata utilizzando una sottorete pubblica

Per prima cosa, ottieni l'ARN dell'attività dal tuo servizio:

```
aws ecs list-tasks --cluster managed-instances-cluster --service managed-instances-service
```

L'output contiene l'ARN dell'attività:

```
{
  "taskArns": [
    "arn:aws:ecs:region:aws_account_id:task/managed-instances-cluster/EXAMPLE"
  ]
}
```

Descrivi l'attività per ottenere l'ARN dell'istanza di container. Utilizza l'ARN dell'attività per il parametro `tasks`:

```
aws ecs describe-tasks --cluster managed-instances-cluster --tasks
arn:aws:ecs:region:aws_account_id:task/managed-instances-cluster/EXAMPLE
```

L'output mostra che l'attività è in esecuzione su Amazon ECS Managed Instances e include l'ARN dell'istanza di container:

```
{
  "tasks": [
    {
      "launchType": "MANAGED_INSTANCES",
      "capacityProviderName": "managed-instances-cp",
      "containerInstanceArn": "arn:aws:ecs:region:aws_account_id:container-instance/managed-instances-cluster/CONTAINER_INSTANCE_ID",
      "taskArn": "arn:aws:ecs:region:aws_account_id:task/managed-instances-cluster/EXAMPLE",
      "taskDefinitionArn": "arn:aws:ecs:region:aws_account_id:task-definition/sample-managed-instances:1"
    }
  ]
}
```

Descrivi l'istanza del contenitore per ottenere l'ID dell' EC2 istanza:

```
aws ecs describe-container-instances --cluster managed-instances-cluster --container-instances CONTAINER_INSTANCE_ID
```

L'output include l'ID dell' EC2 istanza:

```
{
  "containerInstances": [
    {
      "ec2InstanceId": "i-1234567890abcdef0",
      "capacityProviderName": "managed-instances-cp",
      "containerInstanceArn": "arn:aws:ecs:region:aws_account_id:container-instance/managed-instances-cluster/CONTAINER_INSTANCE_ID"
    }
  ]
}
```

Descrivi l' EC2 istanza per ottenere l'indirizzo IP pubblico:

```
aws ec2 describe-instances --instance-ids i-1234567890abcdef0
```

L'indirizzo IP pubblico è indicato nell'output:

```
{
  "Reservations": [
    {
      "Instances": [
        {
          "PublicIpAddress": "198.51.100.2",
          "InstanceId": "i-1234567890abcdef0"
        }
      ]
    }
  ]
}
```

Inserisci l'indirizzo IP pubblico nel browser web; dovresti visualizzare una pagina web che mostra l'applicazione Amazon ECS di esempio in esecuzione su Amazon ECS Managed Instances.

Test di un'attività implementata utilizzando una sottorete privata

Per le attività implementate in sottoreti private, puoi utilizzare Amazon ECS Exec per connetterti al container e testare l'implementazione dall'interno dell'istanza. Segui gli stessi passaggi indicati in precedenza per ottenere l'ARN dell'attività, quindi utilizza ECS Exec:

```
aws ecs execute-command --cluster managed-instances-cluster \  
  --task arn:aws:ecs:region:aws_account_id:task/managed-instances-cluster/EXAMPLE \  
  --container managed-instances-app \  
  --interactive \  
  --command "/bin/sh"
```

Dopo l'esecuzione della shell, puoi testare il server web:

```
curl localhost
```

Dovresti visualizzare l'equivalente HTML della pagina Web che mostra l'applicazione di esempio di Amazon ECS.

Fase 10: Pulizia

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 managed-instances-cluster --service managed-instances-  
service --force
```

Attendi l'eliminazione del servizio e l'arresto di tutte le attività, quindi elimina il provider di capacità:

```
aws ecs delete-capacity-provider --capacity-provider managed-instances-cp
```

Elimina il cluster :

```
aws ecs delete-cluster --cluster managed-instances-cluster
```

 Note

Le istanze gestite vengono terminate automaticamente quando il provider di capacità viene eliminato. Non è necessario terminare manualmente le EC2 istanze.

Ulteriori informazioni su come creare un'attività di Amazon ECS Linux per 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, consulta [Progettazione per AWS Fargate per Amazon ECS](#).

Inizia a usare Amazon ECS usando Fargate per le tue attività nelle regioni in cui Amazon ECS AWS supporta Fargate. AWS Fargate

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

Prerequisiti

Prima di iniziare, completa le fasi illustrate in [Configurazione per l'uso di Amazon ECS](#) e verifica che l'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 blueprint 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 Servizi, scegliere Crea.
3. In Dettagli del servizio, specifica come viene distribuita l'applicazione.
 - a. Per Famiglia di definizioni di attività, scegli la definizione di attività in [Fase 2: Creazione di una definizione di attività](#) cui hai creato.
 - b. In Nome servizio, specificare un nome per il servizio.
4. In Ambiente, scegli Tipo di avvio, quindi scegli FARGATE.
5. In Configurazione di distribuzione, per Attività desiderate, inserisci 1.
6. 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](#).
7. Scegli Create (Crea).

Fase 4: visualizzazione del servizio

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

2. Nel pannello di navigazione scegliere 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. CloudFormation

1. Nel pannello di navigazione scegliere 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. Al prompt di conferma, immetti delete (elimina), quindi scegli Delete (Elimina). In alternativa, puoi utilizzare l'opzione Force delete che consente ad Amazon ECS di dimensionare il servizio per tuo conto prima di eliminarlo.

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.

Ulteriori informazioni su come creare un'attività di Amazon ECS Windows per Fargate

Inizia a usare Amazon ECS usando Fargate per le tue attività nelle regioni in cui Amazon ECS AWS supporta Fargate. AWS Fargate

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

Prerequisiti

Prima di iniziare, completa le fasi illustrate in [Configurazione per l'uso di Amazon ECS](#) e verifica che l'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 Console di gestione AWS

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 Console di gestione AWS

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 Nome servizio, specificare 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 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. CloudFormation

1. Nel pannello di navigazione scegliere 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. Al prompt 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 EC2

Inizia a usare Amazon ECS EC2 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 le fasi illustrate in [Configurazione per l'uso di Amazon ECS](#) e verifica che l'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

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 le EC2istanze 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 Console di gestione AWS

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

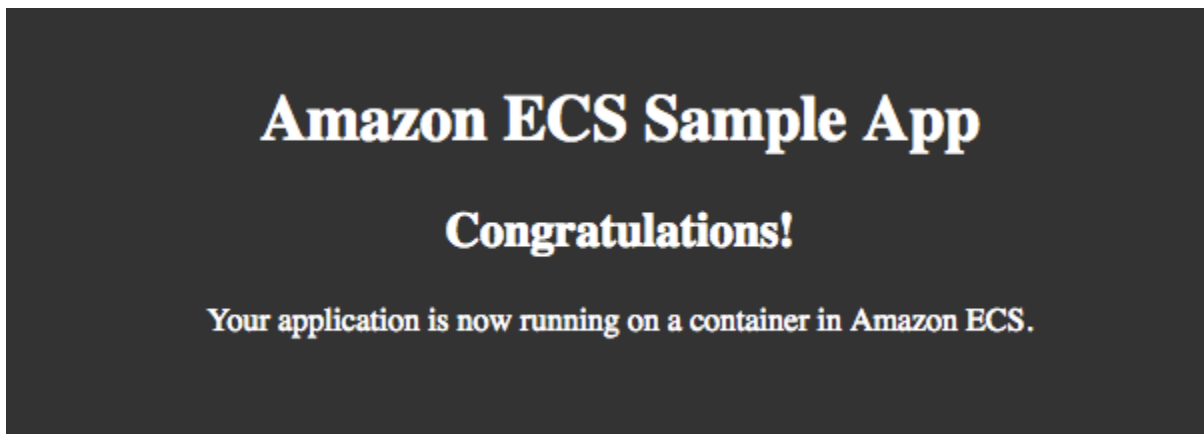
1. Nel pannello di navigazione scegliere Cluster.
2. Seleziona il cluster creato in [Fase 1: creazione di un cluster](#).
3. Nella scheda Servizi, scegli Crea.
4. Nella sezione Environment (Ambiente), completa le operazioni descritte di seguito.
 - a. Per Compute options (Opzioni di calcolo), scegli 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 Nome servizio, specificare 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 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. Lo screenshot seguente mostra 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. CloudFormation

1. Nel pannello di navigazione scegliere 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. Al prompt 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 di 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 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

- [Passaggio 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](#)
- [Fasi successive](#)

Passaggio 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 inizi 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

Utilizza l'immagine del container [amazon-ecs-sample](#). Questa immagine contiene un'app web PHP in esecuzione 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 dei seguenti esempi.

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() *awsdk.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 &awsdk.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 container è stata ottenuta dalla Galleria pubblica di Amazon ECR: `amazon/amazon-ecs-sample`.

- Altre informazioni pertinenti, ad esempio il fatto che il bilanciatore del carico 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 Auto Scaling
- 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 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 bilanciatore del carico e un URL del nuovo server Web. Sono i seguenti:

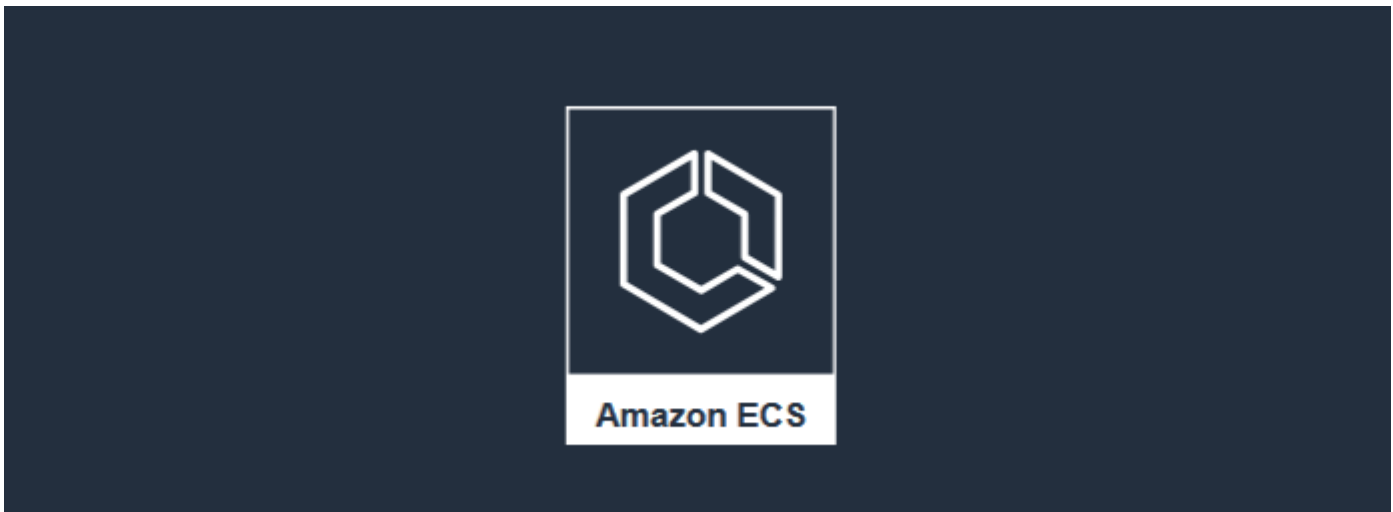
```

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

Fasi successive

Per ulteriori informazioni su come sviluppare l' AWS infrastruttura utilizzando la 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 CI/CD pipeline 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. Console di gestione AWS

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.

Per informazioni sullo sviluppo di Copilot, consulta [What's the future of Copilot?](#).

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.

Utilizzo di 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
```

Download del file binario

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

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/KH8VH1q2uRw32/B94XLEgRbGTmdWfdKuxoPCttBQaMj3LGn6Pe+6xVWRkChQu
BoQAhjbQ+bEm0kNy0LjNgjNlnL3UMAG56t8E3LANIgGgEnpNsB1UwfwluPoGZoTx
N+6pHBJrKIL/1v/ETU4FXpYw2zvhwNahxeNRnoYj3uycHkeliCrw4kj0+skizBg0
2K7oVX80c3j5+ZilhL/qDLXmUCb2az5cMM1m0oF8EKX5HaNuq1KfwJxqXE6NNIc0
lFTTrT7QwD5fMNld3FanLgv/ZnIrsSaqJOL6zRSq804LN10WBVBndExk2Kr+5kFxn
5lBPgFPgRj5hQ+KTHMa9Y8Z7yUc64BJiN6F9N17FJuSsfqbdkvRLsQRbcBG9qxX3
rJAEhieJzVMEUNl+EgeCkxj5xuSkNU7zw2c3hQZqEcraDLV+hvFJkt0z9Gm6xzbq
lTnWWCz4xrIWtuEBA2qE+MlDheVd78a3gIsEaSTfQq0osYXaQbvlnSW0oc1y/5Zb
zizHTJIhLtUyls9WisP2s0emeHZicVMfw61EgPrJAiupgc7kyZvFt4YwfwARAQAB
tCRBbWF6b24gRUNTIDx1Y3Mtc2VjdXJpdHlAYW1hem9uLmNvbT6JAhwEEAECAAYF
AlrjL0YACgkQHivRXs0TaQrg1g/+JppwPqHn1VPmv7lessB8I5UqZeD6p6uVpHd7
Bs3pcPp8BV7BdRbs3sPLt5bV1+rkq0lw+0gZ4Q/ue/YbWt0At4qY00cEo0HgcnaX
lsB827QIfZIVtGWMhuh94xzm/SJkvngml6KB3YJNnWP61A9qJ37/VbVVLzvcmazA
McWB4HUMNrhD0JgBCo0gIpqCbpJEvUc02Bjn23eEJSs9kC70UAHyQkVnx4d9UzXF
```

40oISF6hmQKIBoLnRrAlj5Qvs3GhvhQ0ThYq0Grk/KMJJX2CSqt7tWJ8gk1n3H3Y
SReRXJRnv7DsDDBwFgT6r5Q2HW1TBUvaoZy5hF6maD09nHcNnvBjqADzeT8Tr/Qu
bBCLzkNSYqqkpgtwv7seoD2P4n1giRvDA0EfMZpVkuR+C252IaH1HZFEz+TvBVQM
Y80WWxmIJW+J6evjo3N1e019UHv71jvoF8zljBI4bsL2c+QTJm0v7nRqzDQgCWyp
Id/v2dUVVtK1j9omuLBBwNjzQCB+72LcIzJhYmaP1HC4LcKQG+/f41exuItenatK
1EJQhYtyVXcBlh6Yn/wzNg2NW0wb3vqY/F7m6u9ixAwgtIMgPCDE4aJ86zrrXYFz
N2HqkTSQh77Z8KPKmyGopsmN/reMuilPdINb249nA0dzoN+nj+tTF0YCIaLaFyjs
Z0r1QA0JAjkeEEwECACMFAlq1SasCGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIEAQIX
gAAKCRc86dmkLVF4T9iFEACEnkm1dNXsWUx34R3c0vamHrPxvfkyI1F1EUen8D1h
uX9xy6jCER0HWEp0rjGK4QDPgM93sWJ+s1UAKg214QRVzft0y9/DdR+twApA0fzy
uavIthGd6+03jAAo6udYDE+cZC3P7XBbDiYEWk4XAF9I1JjB8hTZUgvXBL046JhG
eM17+crgUyQeetki0QemLbsbXQ40Bd9V7zf7XJraFd8VrwNUwNb+9KFtgAsc9rk+
YIT/PEf+Y0PysgcxI4sTWghtyCu1VnuGoskgDv4v73PALU0ieUrvvQVqWMrvhVx1
0X90J7cC1K0yh1EQQ1aFTgmQjmxexVTwIBm8LvysFK6YXM41Kj0rlz3+6xBIm/qe
bFyLUnf4Woiu0p1AaJhK9pRY+XENGNxdTn4D26Kd0F+PLkm3Tr3Hy3b10k34F1Gr
KVHUq1T2D7cvMnnKEELTUcKX+1mV3an16nmAg/my1JSUt6BNK2rJpY1s/kkSGSE
XQ4zuF2IGCpVBFhYAlT5Un5zwqkwQR3/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+UDgl
2iHi0KIppQqLbHEfQmHcDd2fix+AaJKMnPGNku9qCFEMbgSRJpXz6BfwnY1QuKE+I
R6jA0frUNT2jhiGG/F8RceXzohaaC/Cx7LUCUFwc0n7z32C9/Dtj7I1PM0acdZzz
bjJzRK0/ZDv+UN/c9dwAk11zAyPMwGBkUaY68EBstnIliw34aWm6IiHhxioVPKSp
VJfyiXP00EXqujtHLAeChfjcns3I12YshT1dv2PafG53fp33ZdzeUgsBo+EAEQEA
AYkCHwQYAQIACQCWrvJqWibDAKCRc86dmkLVF4T+ZdD/9x/8APzgNJF3o3STrF
jvnV1ycyhWYGaEBJiu7wjsNWwzMF0v15tLjB7AqeVxZn+WKDD/mIOQ450ZvnYZuy
X7DR0JszaH9wrYTxZLVruAu+t6UL0y/XQ4L1GZ9QR6+r+7t1Mvbfy7B1HbvX/gYt
Rwe/uwdibI0CagEzyX+2D3kT0LH05XThbXaNf8AN8zha91Jt2Q2UR2X5T6JcwtMz
FBvZn13LSmZyE0EQehS2iUurU4uW0pGppuqVnbi0jbCvCHKgDGrqZ0smKNAQng54
F365W3g8AFY48s8XQwzmcliowYX9bT8PZiEi0J4QmQh0aXkpqZyFefuWe0L2R94S
XKzr+gRh3BAULoqF+qK+IUMxTip9KTPNvYDpiC66yBiT6gFDji5Ca9pGpJXrC3xe
TXiKQ8DBWdhBPVPrRuLiaenTtZE0sPc4I85yt5U9RoPTStc0r34s3w5yEaJagt6S
Gc5r9ysjkfH6+6rbi1ujxMgR0Sqtqr+RyB+V9A5/OgtNZc811K6u4Uo0Cde8jUuW
vqWKvjJB/Kz3u4zaeNu2ZyyHa0q0uH+TETcW+jsY9IhbEzqN5yQYGi4pVmDkY5vu
lXbJnbqPKpRXgM9BecV9AMbPgbdQ/5LhNJJXg+G8YQ0gp4lR/hC1TEFdIp5wM8AK
CWsENyt2o1rjgMXiZ0MF8A5oBlkCDQRatUuSARAAr77kj7j2QR2SZe0S1FBvV7oS
mFeSNnz9xZssqism6bTwSHM6YLDwc7Sdf2esDdyz0NETwqrVCg+FxgL8hmo9hS4c
rR6tmrP0m0mptr+xlLsKcaP7ogIXsyZnrEAEsw8PnfayoiPCdc3cMCR/1TnHFGA

```
7EuR/XLBmi7Qg9tByVYQ5Yj5wB9V4B2yeCt3XtzPqeLKvaxl7PNe1aHGJQY/xo+m
V0bndxf9IY+4oFJ4b1D32WqvYxEso7vW6WBh7oqv3Zbm0yQrr8a6mDBpqLkvWwNI
3kpJR974tg5o5LfDu1BeeyHWPSGm4U/G4JB+JIG1ADy+RmoWEt4BqTCZ/knnoGvw
D5sTCxbKdmu0mhGyTssog+300cGYHV7pWYP hazKHMPm201xKCjH1RfzRULzGKjD+
yMLT1I3AXFmLmZJXika01vE3/wgMqCXscbycbLjLD/bXIuFwo3rzozeXjgi/DJx
jKBAyBTY05nMctH109oaFd9d0Hbs0UDkIMnsgGBE766Piro6MHo0T0rXl07Tp4pI
rwuS0sc6XzCzdImj0Wc6axS/HeUKRXWdXJwno5awTwXKRJMXGfhCvSvbcbc2Wx+L
IKvmb7EB4K3fmjFFE67yolmiw2qRcUBfygtH3eL5XZU28MiCpue8Y8GKJoBAUyvf
KeM1r08Jm3iRac5a/D0AEQEAAyKEPgQYAQIACQUCWrVLkgIbAgIpCRC86dmkLVF4
T8FdIAQZAQIABgUCWrVLkgAKCRDePL1hra+LjtHYD/9MucxdFe6bX01dQR4tKhhQ
P0LRqy6z1BY9ILCLowNdGZdqorogUiUymgn3VhEhVtxT0oHcN7q0uM01PNsRn0eS
EYjf8Xrb1clzkD6xULwm0clTb9bBxBc/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
FJVORaivhoWwzjpfQKhwcU91ABXi2UvVm14v0AfeI7oiJPSU1zM4fEny4oiIBX1R
zhFNih1UjIu82X16mTm3BwbIga/s1fnQRGzyhqUIMii+mWra23EwjChaxpvjjcUH
5illc5Zq781aCYRygYQw+hu5nFk0H1R+Z50Ubxjd/aqUfnGIAX7kPMD3Lof4K1dD
Q8ppQriUvxVo+4nPv6rpTy/PyqCLWDjkguHpJsEFsMkwajrAz0QNSAU5CJ0G2Zu4
yxvYlumHCE17nbFrm0vIiA75Sa8KnywTDsyZsu3Xc0cf3g+g1xWtpjJqy2bYX1qz
9uD0WtArWH0is6bq8l9RE6xr1RBVXS6uqqQIZFBGyq66b0dIq4D2JdsUvgEMaHbc
e7tBfeB1CMBdA64e9Rq7bFR7Tvt8gasCZY1Nr3lydh+dFHIEkH53HzQe6l88HEic
+0jVnLkCDQRa55wJARAaYLYa2Lx6gyoWoJN1a6740q3o8e9d4KggQ0fGMTCf1meq
ivuzgN+3DZHN+9ty2KxXMtn0mhHBerZdbNjyMNT1gAgrhPNB4HtXBxum2wS57WK
DNmade914L7FWTPAWBG2Wn4480EHTqsC1ICXXWy9IICgclAEyIq0Yq5mAdTEgRJS
Z8t4GpwtDL9gNQyFXaWQmDmkAsCygQMvha1mu9x0IzQG5CxSnZFk7zcuL60k14Z3
Cmt49k4T/7ZU8goWi8tt+rU78/IL3J/ff9+1civ10wuUidgfPCSv0UW1JojsdCQA
L+RZJcoXq71f0Fj/eNje0SstCTDPfTCL+kThE6E5neDtbQHBYkEX1BRiTedsV4+M
ucgiTrdQFWKf89G72xdv8ut9AAYQ2BbEYU+JAYhUH8rYYui2dHKJIgjNvJscuUWb
+QEjQIRleJRhr0+/CHgMs4fZAKWF1VFhKBkcKmEjLn1f7EJJUW84ZhKXj0/AUPX
1CHsNjziRceujCJYox1cwsoq6jTE50GiNzcIxTn9xUc0UMKFeggNAFys1K+TDTm3
Bzo8H5ucjCUemUm9lhkGwqTZg0LRX5eqPX+JBoSa0bqhggCa5IPinKR6MgoFPHK
6sYKqroYwBGgZm6Js5chpNchvJMs/3WXN0EVg0J3z3vP0DMhxqWm+r+n9z1W8qsA
EQEAAYKEPgQYAQgACQUCWuecCQIbAgIpCRC86dmkLVF4T8FdIAQZAQgABgUCWuec
CQAKCRBQ3szEcQ5hr+ykD/4t0LRHFHXuKUcxgGaubUcVtsFrwBKma1cYjqaPms8u
```

```

6Sk0wfGRI32G/Gh0rp0Ts/M0kb0bq6VLTh8N5Yc/53ME18zQFw9Y5AmRoW4PZXER
ujs5s7p4oR7xHMHmJCCBn1bvrR+34YPfgzTcgLi0EFHYT8UTxwnGmX0vNkMM7md
xD3CV5q6VAte8WKBo/220II3fcQ1c9r/oWX4kXXkb0v9hoGwKbDJ1tzqTPrp/xFt
yohqnvImpnlz+Q9zXmbrWYL9/g8VCmW/NN2gju2G3Lu/T1FUWIT4v/50PK6TdeNb
VKJ04+S8bTayqSG9CML1S57KSgCo5HUHQWeSNHI+fpe5oX6FALPT9JLDce80Zz1i
cZZ0MELP37m00Qun0AlmHm/hVzf0f311PtzbzcqWaE51tJvgUR/nZFo6Ta305Ezhs
3V1EJNQ1Ijf/6DH87SxvAoRIARCuZd0qxBCDK0avpFzUtbJd241RA3WJpkEiMqKv
RDVZkE4b6TW61f0o+LaVfK6E8oLpixonS4fiqC16mFr0dyRk+RJJfIUyz0WTDVmt
g0U1C01ezokMSqkJ7724pyjr2xf/r9/sC6a0JwB/lKgZkJfC6NqL7T1xVA31dUga
LE0vEJTTE4gl+tYtfsCDvALCtqL0jduSkUo+RXcBIxmXhA+tShW0pbS2Rtx/ixua
KohVD/0R4QxiSwQmICNtm9mw9ydI1lyjYXX5a9x4wMJracNY/LBybJPFnZnT4dYR
z4XjqysDwvvYZByaWoIe3QxjX84V6M1I2IdAT/xImu8gbaCI8tmyfpIrLnPKiR9D
VFYfGBXuAX7+HgPPSFtrHQONCALxxz1bNpS+zxt9r0MiLgcLyspWxSdmoYGZ6nQP
R05Nm/ZVS+u2imPCRzNUZEMa+d1E6kHx0rS0dPiuJ407NtPeYDKkoQtNagspsDvh
cK7CSqAiKMq06UBTxqlTSRkm62e0Ctcs3p30eHu5GRZF1uzTET0ZxYkaPgdrQknx
ozjP5mC7X+451cCfmcVt94TFNL5HwEUVJpm0gmzILCI8yoDTWzloo+i+fPFsXX4f
kynhE83mSEcr5VHFYrTY3mQXGmNJ3bCLuc/jq7ysGq69xiKmT1UeXFm+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 AWS firme 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](#).

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

1. Implementate un'applicazione Web di esempio clonata da un GitHub repository utilizzando il seguente comando. [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. Inoltre, per individuare lo stato dell'app puoi utilizzare i seguenti comandi.

- Elenca tutte le tue applicazioni Copilot. AWS

```
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. Una volta terminata questa dimostrazione, esegui il seguente comando per pulire le risorse associate ed evitare costi aggiuntivi 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. CloudFormation utilizza modelli che sono un file di testo JSON formattato YAML o un file di testo. I modelli sono come progetti per la AWS risorsa che desideri creare. Quando crei e invii un modello, CloudFormation crea uno stack. Puoi gestire le risorse che hai definito nel modello tramite lo stack. Quando vuoi creare, aggiornare o eliminare una risorsa, puoi creare, aggiornare o eliminare lo stack creato da tale risorsa. Per aggiornare gli stack, devi prima creare un set di modifiche. I set di modifiche mostrano su cosa influisce la modifica prima di apportarla. In questo modo, è possibile evitare di eliminare accidentalmente i database, ad esempio modificando il loro nome. Per ulteriori informazioni su modelli, pile e set di modifiche, consulta [How CloudFormation works](#) nella AWS CloudFormation User Guide.

Utilizzando 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, CloudFormation si occupa del provisioning e della configurazione di tali risorse per te.

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

CloudFormation i modelli possono essere utilizzati sia con che AWS Command Line Interface per creare risorse. Console di gestione AWS

Per ulteriori informazioni 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

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

Creazione di risorse Amazon ECS utilizzando la console CloudFormation

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

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- L'utente IAM ha le autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS FullAccess](#).

Fase 1: crea un modello di stack

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

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

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWS Docs] ECS: load-balanced-web-application'

Parameters:
  VpcCidr:
    Type: String
    Default: '10.0.0.0/16'
    Description: CIDR block for the VPC
  ContainerImage:
    Type: String
    Default: 'public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest'
    Description: Container image to use in task definition
  PublicSubnet1Cidr:
    Type: String
```

Default: '10.0.1.0/24'

Description: CIDR block for public subnet 1

PublicSubnet2Cidr:

Type: String

Default: '10.0.2.0/24'

Description: CIDR block for public subnet 2

PrivateSubnet1Cidr:

Type: String

Default: '10.0.3.0/24'

Description: CIDR block for private subnet 1

PrivateSubnet2Cidr:

Type: String

Default: '10.0.4.0/24'

Description: CIDR block for private subnet 2

ServiceName:

Type: String

Default: 'tutorial-app'

Description: Name of the ECS service

ContainerPort:

Type: Number

Default: 80

Description: Port on which the container listens

DesiredCount:

Type: Number

Default: 2

Description: Desired number of tasks

MinCapacity:

Type: Number

Default: 1

Description: Minimum number of tasks for auto scaling

MaxCapacity:

Type: Number

Default: 10

Description: Maximum number of tasks for auto scaling

Resources:

```
# VPC and Networking
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: !Ref VpcCidr
    EnableDnsHostnames: true
    EnableDnsSupport: true
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-vpc'

# Internet Gateway
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

# Public Subnets for ALB
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}-public-subnet-1'

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}-public-subnet-2'

# Private Subnets for ECS Tasks
PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet1Cidr
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-subnet-1'

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet2Cidr
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-subnet-2'

# NAT Gateways for private subnet internet access
NatGateway1EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-eip-1'

NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-eip-2'
```



```
NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-1'

NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-2'

# Route Tables
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-public-routes'

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

PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-routes-1'

DefaultPrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-routes-2'

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2
```

```
# Security Groups
ALBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: !Sub '${AWS::StackName}-alb-sg'
    GroupDescription: Security group for Application Load Balancer
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
        Description: Allow HTTP traffic from internet
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0
        Description: Allow all outbound traffic
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-alb-sg'

ECSSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: !Sub '${AWS::StackName}-ecs-sg'
    GroupDescription: Security group for ECS tasks
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: !Ref ContainerPort
        ToPort: !Ref ContainerPort
        SourceSecurityGroupId: !Ref ALBSecurityGroup
        Description: Allow traffic from ALB
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0
        Description: Allow all outbound traffic
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-ecs-sg'

# Application Load Balancer
ApplicationLoadBalancer:
```

```
Type: AWS::ElasticLoadBalancingV2::LoadBalancer
Properties:
  Name: !Sub '${AWS::StackName}-alb'
  Scheme: internet-facing
  Type: application
  Subnets:
    - !Ref PublicSubnet1
    - !Ref PublicSubnet2
  SecurityGroups:
    - !Ref ALBSecurityGroup
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-alb'

ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: !Sub '${AWS::StackName}-tg'
    Port: !Ref ContainerPort
    Protocol: HTTP
    VpcId: !Ref VPC
    TargetType: ip
    HealthCheckIntervalSeconds: 30
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 2
    UnhealthyThresholdCount: 5
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-tg'

ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP

# ECS Cluster
ECSCluster:
```

```

Type: AWS::ECS::Cluster
Properties:
  ClusterName: !Sub '${AWS::StackName}-cluster'
  CapacityProviders:
    - FARGATE
    - FARGATE_SPOT
  DefaultCapacityProviderStrategy:
    - CapacityProvider: FARGATE
      Weight: 1
    - CapacityProvider: FARGATE_SPOT
      Weight: 4
  ClusterSettings:
    - Name: containerInsights
      Value: enabled
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-cluster'

# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub '${AWS::StackName}-task-execution-role'
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    ManagedPolicyArns:
      - arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-task-execution-role'

ECSTaskRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub '${AWS::StackName}-task-role'
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow

```

```
Principal:
  Service: ecs-tasks.amazonaws.com
  Action: sts:AssumeRole
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-task-role'

# CloudWatch Log Group
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties:
    LogGroupName: !Sub '/ecs/${AWS::StackName}'
    RetentionInDays: 7

# ECS Task Definition
TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Family: !Sub '${AWS::StackName}-task'
    Cpu: '256'
    Memory: '512'
    NetworkMode: awsvpc
    RequiresCompatibilities:
      - FARGATE
    ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
    TaskRoleArn: !GetAtt ECSTaskRole.Arn
    ContainerDefinitions:
      - Name: !Ref ServiceName
        Image: !Ref ContainerImage
        PortMappings:
          - ContainerPort: !Ref ContainerPort
            Protocol: tcp
        Essential: true
        LogConfiguration:
          LogDriver: awslogs
          Options:
            awslogs-group: !Ref LogGroup
            awslogs-region: !Ref AWS::Region
            awslogs-stream-prefix: ecs
        HealthCheck:
          Command:
            - CMD-SHELL
            - curl -f http://localhost/ || exit 1
          Interval: 30
```

```
        Timeout: 5
        Retries: 3
        StartPeriod: 60
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-task'

# ECS Service
ECSService:
  Type: AWS::ECS::Service
  DependsOn: ALBListener
  Properties:
    ServiceName: !Sub '${AWS::StackName}-service'
    Cluster: !Ref ECSCluster
    TaskDefinition: !Ref TaskDefinition
    DesiredCount: !Ref DesiredCount
    LaunchType: FARGATE
    PlatformVersion: LATEST
    NetworkConfiguration:
      AwsVpcConfiguration:
        AssignPublicIp: DISABLED
        SecurityGroups:
          - !Ref ECSSecurityGroup
        Subnets:
          - !Ref PrivateSubnet1
          - !Ref PrivateSubnet2
    LoadBalancers:
      - ContainerName: !Ref ServiceName
        ContainerPort: !Ref ContainerPort
        TargetGroupArn: !Ref ALBTargetGroup
    DeploymentConfiguration:
      MaximumPercent: 200
      MinimumHealthyPercent: 50
      DeploymentCircuitBreaker:
        Enable: true
        Rollback: true
    EnableExecuteCommand: true # For debugging
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-service'

# Auto Scaling Target
ServiceScalingTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
```

```

Properties:
  MaxCapacity: !Ref MaxCapacity
  MinCapacity: !Ref MinCapacity
  ResourceId: !Sub 'service/${ECSCluster}/${ECSService.Name}'
  RoleARN: !Sub 'arn:aws:iam:${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService'
  ScalableDimension: ecs:service:DesiredCount
  ServiceNamespace: ecs

# Auto Scaling Policy - CPU Utilization
ServiceScalingPolicy:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub '${AWS::StackName}-cpu-scaling-policy'
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ServiceScalingTarget
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageCPUUtilization
      TargetValue: 70.0
      ScaleOutCooldown: 300
      ScaleInCooldown: 300

Outputs:
  VPCId:
    Description: VPC ID
    Value: !Ref VPC
    Export:
      Name: !Sub '${AWS::StackName}-VPC-ID'

  LoadBalancerURL:
    Description: URL of the Application Load Balancer
    Value: !Sub 'http://${ApplicationLoadBalancer.DNSName}'
    Export:
      Name: !Sub '${AWS::StackName}-ALB-URL'

  ECSClusterName:
    Description: Name of the ECS Cluster
    Value: !Ref ECSCluster
    Export:
      Name: !Sub '${AWS::StackName}-ECS-Cluster'

  ECSServiceName:

```



```
Description: Name of the ECS Service
Value: !GetAtt ECSService.Name
Export:
  Name: !Sub '${AWS::StackName}-ECS-Service'

PrivateSubnet1:
  Description: Private Subnet 1 ID
  Value: !Ref PrivateSubnet1
  Export:
    Name: !Sub '${AWS::StackName}-Private-Subnet-1'

PrivateSubnet2:
  Description: Private Subnet 2 ID
  Value: !Ref PrivateSubnet2
  Export:
    Name: !Sub '${AWS::StackName}-Private-Subnet-2'
```

Il modello utilizzato in questo tutorial crea un servizio Amazon ECS con due attività eseguite su Fargate. Ciascuna delle attività esegue un'applicazione Amazon ECS di esempio. Il modello crea anche un Application Load Balancer che distribuisce il traffico delle applicazioni e una policy Application Auto Scaling che scala l'applicazione in base all'utilizzo della CPU. Il modello crea anche le risorse di rete necessarie per distribuire l'applicazione, le risorse di registrazione per i log dei container e un ruolo IAM per l'esecuzione delle attività di Amazon ECS. Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#). Per ulteriori informazioni sul dimensionamento automatico, consulta [Scalabilità automatica del servizio Amazon ECS](#).

Fase 2: crea 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 CloudFormation

Per informazioni su come creare uno stack utilizzando la CloudFormation console, consulta [Creazione di uno stack sulla CloudFormation console](#) nella Guida per l'AWS CloudFormation utente e utilizza la tabella seguente per determinare quali opzioni specificare.

Opzione	Valore	
Prerequisito: preparare il modello	Scegli un modello esistente	
Specificare il modello	Upload a template file (Carica un file modello)	
Scegli un file	ecs-tutorial-template.yaml	
Stack name (Nome stack)	ecs-tutorial-stack	
Parameters	Lascia tutti i valori dei parametri come predefiniti.	
Funzionalità	Scegli Riconosco che questo modello può creare risorse IAM per confermare la CloudFormation creazione di risorse IAM.	

Fase 3: verifica

Utilizza la procedura seguente per verificare la creazione di risorse Amazon ECS tramite il modello fornito.

Per informazioni su come visualizzare le informazioni e le risorse dello stack, consulta [Visualizzazione delle informazioni sullo stack dalla CloudFormation console](#) nella Guida per l'AWS CloudFormation utente e utilizza la tabella seguente per determinare cosa verificare.

Campo dei dettagli dello stack	Cosa cercare	
Info stack	Uno stato di CREATE_COMPLETE .	
Resources	Un elenco delle risorse create con collegamenti alla console del servizio. Scegli i link a	

Campo dei dettagli dello stack	Cosa cercare	
	ECSCluster , ECSService , TaskDefinition per visualizzare maggiori dettagli sul servizio creato, sul cluster e sulla definizione delle attività nella console Amazon ECS.	
Output	LoadBalancerURL. Incolla l'URL in un browser web per visualizzare una pagina web che mostra un'applicazione Amazon ECS di esempio.	

Fase 4: Eliminazione delle risorse

Per ripulire le risorse ed evitare di incorrere in ulteriori costi, segui i passaggi descritti in [Eliminare uno stack dalla CloudFormation console nella guida per l'utente](#).CloudFormation

Creazione di risorse Amazon ECS utilizzando AWS CLI comandi per CloudFormation

Un altro modo per utilizzare Amazon ECS con CloudFormation è tramite AWS CLI. Puoi usare i comandi per creare i tuoi CloudFormation stack per i componenti di Amazon ECS come definizioni di attività, cluster e servizi e distribuirli. Il seguente tutorial mostra come utilizzare il AWS CLI per creare risorse Amazon ECS utilizzando un CloudFormation modello.

Prerequisiti

- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- L'utente IAM ha le autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS FullAccess](#).

Fase 1: creazione di uno stack

Per creare uno stack utilizzando il file AWS CLI salvato in un file chiamato `ecs-tutorial-template.yaml`, esegui il seguente comando.

```
cat << 'EOF' > ecs-tutorial-template.yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWS Docs] ECS: load-balanced-web-application'
Parameters:
  VpcCidr:
    Type: String
    Default: '10.0.0.0/16'
    Description: CIDR block for the VPC
  ContainerImage:
    Type: String
    Default: 'public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest'
    Description: Container image to use in task definition

  PublicSubnet1Cidr:
    Type: String
    Default: '10.0.1.0/24'
    Description: CIDR block for public subnet 1

  PublicSubnet2Cidr:
    Type: String
    Default: '10.0.2.0/24'
    Description: CIDR block for public subnet 2

  PrivateSubnet1Cidr:
    Type: String
    Default: '10.0.3.0/24'
    Description: CIDR block for private subnet 1

  PrivateSubnet2Cidr:
    Type: String
    Default: '10.0.4.0/24'
    Description: CIDR block for private subnet 2

  ServiceName:
    Type: String
    Default: 'tutorial-app'
    Description: Name of the ECS service
```

```
ContainerPort:
  Type: Number
  Default: 80
  Description: Port on which the container listens

DesiredCount:
  Type: Number
  Default: 2
  Description: Desired number of tasks

MinCapacity:
  Type: Number
  Default: 1
  Description: Minimum number of tasks for auto scaling

MaxCapacity:
  Type: Number
  Default: 10
  Description: Maximum number of tasks for auto scaling

Resources:
  # VPC and Networking
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCidr
      EnableDnsHostnames: true
      EnableDnsSupport: true
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-vpc'

  # Internet Gateway
  InternetGateway:
    Type: AWS::EC2::InternetGateway
    Properties:
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-igw'

  InternetGatewayAttachment:
    Type: AWS::EC2::VPCElasticNetworkAttachment
    Properties:
      InternetGatewayId: !Ref InternetGateway
```

```
VpcId: !Ref VPC

# Public Subnets for ALB
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}-public-subnet-1'

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}-public-subnet-2'

# Private Subnets for ECS Tasks
PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet1Cidr
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-subnet-1'

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet2Cidr
    Tags:
      - Key: Name
```

```
Value: !Sub '${AWS::StackName}-private-subnet-2'

# NAT Gateways for private subnet internet access
NatGateway1EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-eip-1'

NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-eip-2'

NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-1'

NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-2'

# Route Tables
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
```

Tags:

- Key: Name
Value: !Sub '\${AWS::StackName}-public-routes'

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

PrivateRouteTable1:

Type: AWS::EC2::RouteTable
Properties:
VpcId: !Ref VPC
Tags:
- Key: Name
Value: !Sub '\${AWS::StackName}-private-routes-1'

DefaultPrivateRoute1:

Type: AWS::EC2::Route
Properties:
RouteTableId: !Ref PrivateRouteTable1
DestinationCidrBlock: 0.0.0.0/0
NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation
Properties:
RouteTableId: !Ref PrivateRouteTable1
SubnetId: !Ref PrivateSubnet1


```

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-routes-2'

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

# Security Groups
ALBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: !Sub '${AWS::StackName}-alb-sg'
    GroupDescription: Security group for Application Load Balancer
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
        Description: Allow HTTP traffic from internet
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0
        Description: Allow all outbound traffic
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-alb-sg'

ECSSecurityGroup:

```

```
Type: AWS::EC2::SecurityGroup
Properties:
  GroupName: !Sub '${AWS::StackName}-ecs-sg'
  GroupDescription: Security group for ECS tasks
  VpcId: !Ref VPC
  SecurityGroupIngress:
    - IpProtocol: tcp
      FromPort: !Ref ContainerPort
      ToPort: !Ref ContainerPort
      SourceSecurityGroupId: !Ref ALBSecurityGroup
      Description: Allow traffic from ALB
  SecurityGroupEgress:
    - IpProtocol: -1
      CidrIp: 0.0.0.0/0
      Description: Allow all outbound traffic
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-ecs-sg'
```

Application Load Balancer

```
ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Name: !Sub '${AWS::StackName}-alb'
    Scheme: internet-facing
    Type: application
    Subnets:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2
    SecurityGroups:
      - !Ref ALBSecurityGroup
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-alb'
```

```
ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: !Sub '${AWS::StackName}-tg'
    Port: !Ref ContainerPort
    Protocol: HTTP
    VpcId: !Ref VPC
    TargetType: ip
    HealthCheckIntervalSeconds: 30
```

```
HealthCheckPath: /
HealthCheckProtocol: HTTP
HealthCheckTimeoutSeconds: 5
HealthyThresholdCount: 2
UnhealthyThresholdCount: 5
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-tg'
```

ALBListener:

```
Type: AWS::ElasticLoadBalancingV2::Listener
Properties:
  DefaultActions:
    - Type: forward
      TargetGroupArn: !Ref ALBTargetGroup
  LoadBalancerArn: !Ref ApplicationLoadBalancer
  Port: 80
  Protocol: HTTP
```

ECS Cluster

ECSCluster:

```
Type: AWS::ECS::Cluster
Properties:
  ClusterName: !Sub '${AWS::StackName}-cluster'
  CapacityProviders:
    - FARGATE
    - FARGATE_SPOT
  DefaultCapacityProviderStrategy:
    - CapacityProvider: FARGATE
      Weight: 1
    - CapacityProvider: FARGATE_SPOT
      Weight: 4
  ClusterSettings:
    - Name: containerInsights
      Value: enabled
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-cluster'
```

IAM Roles

ECSTaskExecutionRole:

```
Type: AWS::IAM::Role
Properties:
  RoleName: !Sub '${AWS::StackName}-task-execution-role'
```

```

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
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-task-execution-role'

```

```

ECSTaskRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub '${AWS::StackName}-task-role'
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-task-role'

```

```

# CloudWatch Log Group
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties:
    LogGroupName: !Sub '/ecs/${AWS::StackName}'
    RetentionInDays: 7

```

```

# ECS Task Definition
TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Family: !Sub '${AWS::StackName}-task'
    Cpu: '256'
    Memory: '512'
    NetworkMode: awsvpc
    RequiresCompatibilities:

```

```

- FARGATE
ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
TaskRoleArn: !GetAtt ECSTaskRole.Arn
ContainerDefinitions:
  - Name: !Ref ServiceName
    Image: !Ref ContainerImage
    PortMappings:
      - ContainerPort: !Ref ContainerPort
        Protocol: tcp
    Essential: true
    LogConfiguration:
      LogDriver: awslogs
      Options:
        awslogs-group: !Ref LogGroup
        awslogs-region: !Ref AWS::Region
        awslogs-stream-prefix: ecs
    HealthCheck:
      Command:
        - CMD-SHELL
        - curl -f http://localhost/ || exit 1
      Interval: 30
      Timeout: 5
      Retries: 3
      StartPeriod: 60
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-task'

```

```
# ECS Service
```

```

ECSService:
  Type: AWS::ECS::Service
  DependsOn: ALBListener
  Properties:
    ServiceName: !Sub '${AWS::StackName}-service'
    Cluster: !Ref ECSCluster
    TaskDefinition: !Ref TaskDefinition
    DesiredCount: !Ref DesiredCount
    LaunchType: FARGATE
    PlatformVersion: LATEST
    NetworkConfiguration:
      AwsvpcConfiguration:
        AssignPublicIp: DISABLED
      SecurityGroups:
        - !Ref ECSSecurityGroup

```

```

    Subnets:
      - !Ref PrivateSubnet1
      - !Ref PrivateSubnet2
  LoadBalancers:
    - ContainerName: !Ref ServiceName
      ContainerPort: !Ref ContainerPort
      TargetGroupArn: !Ref ALBTargetGroup
  DeploymentConfiguration:
    MaximumPercent: 200
    MinimumHealthyPercent: 50
    DeploymentCircuitBreaker:
      Enable: true
      Rollback: true
  EnableExecuteCommand: true # For debugging
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-service'

# Auto Scaling Target
ServiceScalingTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: !Ref MaxCapacity
    MinCapacity: !Ref MinCapacity
    ResourceId: !Sub 'service/${ECSCluster}/${ECSService.Name}'
    RoleARN: !Sub 'arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService'
    ScalableDimension: ecs:service:DesiredCount
    ServiceNamespace: ecs

# Auto Scaling Policy - CPU Utilization
ServiceScalingPolicy:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub '${AWS::StackName}-cpu-scaling-policy'
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ServiceScalingTarget
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageCPUUtilization
      TargetValue: 70.0
      ScaleOutCooldown: 300
      ScaleInCooldown: 300

```

```
Outputs:
  VPCId:
    Description: VPC ID
    Value: !Ref VPC
    Export:
      Name: !Sub '${AWS::StackName}-VPC-ID'

  LoadBalancerURL:
    Description: URL of the Application Load Balancer
    Value: !Sub 'http://${ApplicationLoadBalancer.DNSName}'
    Export:
      Name: !Sub '${AWS::StackName}-ALB-URL'

  ECSClusterName:
    Description: Name of the ECS Cluster
    Value: !Ref ECSCluster
    Export:
      Name: !Sub '${AWS::StackName}-ECS-Cluster'

  ECSServiceName:
    Description: Name of the ECS Service
    Value: !GetAtt ECSService.Name
    Export:
      Name: !Sub '${AWS::StackName}-ECS-Service'

  PrivateSubnet1:
    Description: Private Subnet 1 ID
    Value: !Ref PrivateSubnet1
    Export:
      Name: !Sub '${AWS::StackName}-Private-Subnet-1'

  PrivateSubnet2:
    Description: Private Subnet 2 ID
    Value: !Ref PrivateSubnet2
    Export:
      Name: !Sub '${AWS::StackName}-Private-Subnet-2'

EOF
```

Il modello utilizzato in questo tutorial crea un servizio Amazon ECS con due attività eseguite su Fargate. Ciascuna delle attività esegue un'applicazione Amazon ECS di esempio. Il modello crea anche un Application Load Balancer che distribuisce il traffico delle applicazioni e una policy Application Auto Scaling che scala l'applicazione in base all'utilizzo della CPU. Il modello crea anche

le risorse di rete necessarie per distribuire l'applicazione, le risorse di registrazione per i log dei container e un ruolo IAM per l'esecuzione delle attività di Amazon ECS. Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#). Per ulteriori informazioni sul dimensionamento automatico, consulta [Scalabilità automatica del servizio Amazon ECS](#).

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

```
aws cloudformation create-stack \  
  --stack-name ecs-tutorial-stack \  
  --template-body file://ecs-tutorial-template.yaml \  
  --region aws-region \  
  --capabilities CAPABILITY_NAMED_IAM
```

Dopo aver eseguito il comando `create-stack`, puoi utilizzare `describe-stacks` per controllare lo stato della creazione dello stack.

```
aws cloudformation describe-stacks \  
  --stack-name ecs-tutorial-stack \  
  --region aws-region
```

Fase 2: verifica della creazione di risorse Amazon ECS

Per garantire che le risorse Amazon ECS vengano create correttamente, segui questi passaggi.

1. Esegui il comando seguente per elencare tutte le definizioni di attività in una Regione AWS.

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

Il comando restituisce un elenco dei nomi della risorsa Amazon (ARN) delle definizioni delle attività. L'ARN della definizione dell'attività creata utilizzando il modello verrà visualizzato nel formato seguente.

```
{  
  "taskDefinitionArns": [  
    .....
```



```
    "arn:aws:ecs:aws-region:111122223333:task-definition/ecs-tutorial-stack-  
task:1",  
    .....  
  ]  
}
```

2. Esegui il comando seguente per elencare tutti i cluster in una 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 formato seguente.

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

3. Esegui il comando seguente per elencare tutti i servizi definiti nel cluster `ecs-tutorial-stack-cluster`.

```
aws ecs list-services \  
  --cluster ecs-tutorial-stack-cluster
```

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

```
{  
  "serviceArns": [  
    "arn:aws:ecs:aws-region:111122223333:service/ecs-tutorial-stack-cluster/  
ecs-tutorial-stack-service"  
  ]  
}
```

È inoltre possibile ottenere il nome DNS dell'Application Load Balancer che è stato creato e utilizzarlo per verificare la creazione di risorse. Per ottenere il nome DNS, esegui il comando seguente:

Esegui il comando seguente per recuperare gli output dello stack creato.

```
aws cloudformation describe-stacks \  
  --stack-name ecs-tutorial-stack \  
  --region aws-region \  
  --query 'Stacks[0].Outputs[?OutputKey==`LoadBalancerURL`].OutputValue' \  
  --output text
```

Output:

```
http://ecs-tutorial-stack-alb-0123456789.aws-region.elb.amazonaws.com
```

Incolla il nome DNS in un browser per visualizzare una pagina web che mostra un'applicazione Amazon ECS di esempio.

Fase 3: pulizia

Per ripulire le risorse create, esegui il comando seguente.

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

Il `delete-stack` comando avvia l'eliminazione dello CloudFormation stack creato in questo tutorial, eliminando tutte le risorse nello stack. Per verificare l'eliminazione, ripeti la procedura in [Fase 2: verifica della creazione di risorse Amazon ECS](#). L'elenco degli ARNs output non includerà più una definizione di attività chiamata o un cluster chiamato `ecs-tutorial-stack-task`. `ecs-tutorial-stack-cluster` La chiamata `list-services` avrà esito negativo.

CloudFormation modelli di esempio per Amazon ECS

Puoi creare cluster Amazon ECS, definizioni di attività e servizi utilizzando. CloudFormation I seguenti argomenti includono modelli che illustrano come creare risorse con configurazioni diverse. Puoi creare queste risorse con questi modelli utilizzando la CloudFormation console o il. AWS CLI

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

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

Argomenti

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

Definizioni di processi

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

Definizione dell'attività Linux Fargate

Per creare un'attività Linux Fargate di esempio, puoi utilizzare il modello seguente.

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 dell'attività di Amazon EFS

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

```

Fornitori di capacità

I provider di capacità sono associati a un cluster Amazon ECS e vengono utilizzati per gestire la capacità di elaborazione dei carichi di lavoro.

Crea un provider di capacità per le istanze gestite Amazon ECS

Per impostazione predefinita, Amazon ECS fornisce un provider di capacità che seleziona automaticamente i tipi di istanze generiche più ottimizzati in termini di costi. Tuttavia, puoi creare provider di capacità personalizzati per specificare requisiti delle istanze come tipi di istanze, produttori di CPU, tipi di acceleratori e altri requisiti. Puoi utilizzare il seguente modello per creare un provider di capacità per le istanze gestite Amazon ECS che soddisfi i requisiti di memoria e CPU specificati.

JSON

```

{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Resources": {
    "MyCapacityProvider": {
      "Type": "AWS::ECS::CapacityProvider",
      "Properties": {
        "ManagedInstancesProvider": {
          "InfrastructureRoleArn": "arn:aws:iam::123456789012:role/
ECSInfrastructureRole",
          "InstanceLaunchTemplate": {
            "Ec2InstanceProfileArn":
"arn:aws:iam::123456789012:instance-profile/ecsInstanceProfile",

```



```
SecurityGroups:
  - "sg-87654321"
StorageConfiguration:
  StorageSizeGiB: 30
InstanceRequirements:
  VCpuCount:
    Min: 1
    Max: 4
  MemoryMiB:
    Min: 2048
    Max: 8192
```

Cluster

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

Crea un cluster vuoto con impostazioni predefinite

Per creare un cluster vuoto con le impostazioni predefinite, puoi utilizzare il modello seguente.

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 dell'archiviazione gestita e Container Insights avanzato

Puoi utilizzare il modello seguente per creare un cluster con archiviazione gestita 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 di Amazon EBS, consulta [Crittografia dei dati archiviati nei volumi Amazon EBS allegati alle attività Amazon ECS](#). Per ulteriori informazioni sull'utilizzo di Container Insights con osservabilità avanzata, consulta [Monitora i container di 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.xlarge",
```

```

        "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 i servizi. Per ulteriori informazioni sui servizi di Amazon ECS, consulta [Servizi Amazon ECS](#).

Distribuisci un'applicazione web con bilanciatore del carico

Il modello seguente crea un servizio Amazon ECS con due attività eseguite su Fargate. Ciascuna delle attività ha un container NGINX. Il modello crea anche un Application Load Balancer che distribuisce il traffico delle applicazioni e una policy Application Auto Scaling che scala l'applicazione in base all'utilizzo della CPU. Il modello crea anche le risorse di rete necessarie per distribuire l'applicazione, le risorse di registrazione per i log dei container e un ruolo IAM per l'esecuzione delle attività di Amazon ECS. Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#). Per ulteriori informazioni sul dimensionamento automatico, consulta [Scalabilità automatica del servizio Amazon ECS](#).

JSON

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Description": "[AWS Docs] ECS: load-balanced-web-application",
  "Parameters": {
    "VpcCidr": {
      "Type": "String",
      "Default": "10.0.0.0/16",
      "Description": "CIDR block for the VPC"
    },
    "ContainerImage": {
      "Type": "String",
      "Default": "public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest",
      "Description": "Container image to use in task definition"
    }
  }
}
```

```
},
"PublicSubnet1Cidr": {
  "Type": "String",
  "Default": "10.0.1.0/24",
  "Description": "CIDR block for public subnet 1"
},
"PublicSubnet2Cidr": {
  "Type": "String",
  "Default": "10.0.2.0/24",
  "Description": "CIDR block for public subnet 2"
},
"PrivateSubnet1Cidr": {
  "Type": "String",
  "Default": "10.0.3.0/24",
  "Description": "CIDR block for private subnet 1"
},
"PrivateSubnet2Cidr": {
  "Type": "String",
  "Default": "10.0.4.0/24",
  "Description": "CIDR block for private subnet 2"
},
"ServiceName": {
  "Type": "String",
  "Default": "tutorial-app",
  "Description": "Name of the ECS service"
},
"ContainerPort": {
  "Type": "Number",
  "Default": 80,
  "Description": "Port on which the container listens"
},
"DesiredCount": {
  "Type": "Number",
  "Default": 2,
  "Description": "Desired number of tasks"
},
"MinCapacity": {
  "Type": "Number",
  "Default": 1,
  "Description": "Minimum number of tasks for auto scaling"
},
"MaxCapacity": {
  "Type": "Number",
  "Default": 10,
```

```

        "Description": "Maximum number of tasks for auto scaling"
    },
    "Resources": {
        "VPC": {
            "Type": "AWS::EC2::VPC",
            "Properties": {
                "CidrBlock": {
                    "Ref": "VpcCidr"
                },
                "EnableDnsHostnames": true,
                "EnableDnsSupport": 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}-public-subnet-1"
          }
        }
      ]
    }
  },
  "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": "${AWS::StackName}-public-subnet-2"
                }
            }
        ]
    },
    "PrivateSubnet1": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {
            "VpcId": {
                "Ref": "VPC"
            },
            "AvailabilityZone": {
                "Fn::Select": [
                    0,
                    {
                        "Fn::GetAZs": ""
                    }
                ]
            },
            "CidrBlock": {
                "Ref": "PrivateSubnet1Cidr"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": {
                        "Fn::Sub": "${AWS::StackName}-private-subnet-1"
                    }
                }
            ]
        }
    },
    "PrivateSubnet2": {
        "Type": "AWS::EC2::Subnet",
        "Properties": {

```

```

        "VpcId": {
            "Ref": "VPC"
        },
        "AvailabilityZone": {
            "Fn::Select": [
                1,
                {
                    "Fn::GetAZs": ""
                }
            ]
        },
        "CidrBlock": {
            "Ref": "PrivateSubnet2Cidr"
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-private-subnet-2"
                }
            }
        ]
    },
    "NatGateway1EIP": {
        "Type": "AWS::EC2::EIP",
        "DependsOn": "InternetGatewayAttachment",
        "Properties": {
            "Domain": "vpc",
            "Tags": [
                {
                    "Key": "Name",
                    "Value": {
                        "Fn::Sub": "${AWS::StackName}-nat-eip-1"
                    }
                }
            ]
        }
    },
    "NatGateway2EIP": {
        "Type": "AWS::EC2::EIP",
        "DependsOn": "InternetGatewayAttachment",
        "Properties": {
            "Domain": "vpc",

```

```

        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-nat-eip-2"
                }
            }
        ]
    },
    "NatGateway1": {
        "Type": "AWS::EC2::NatGateway",
        "Properties": {
            "AllocationId": {
                "Fn::GetAtt": [
                    "NatGateway1EIP",
                    "AllocationId"
                ]
            },
            "SubnetId": {
                "Ref": "PublicSubnet1"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": {
                        "Fn::Sub": "${AWS::StackName}-nat-1"
                    }
                }
            ]
        }
    },
    "NatGateway2": {
        "Type": "AWS::EC2::NatGateway",
        "Properties": {
            "AllocationId": {
                "Fn::GetAtt": [
                    "NatGateway2EIP",
                    "AllocationId"
                ]
            },
            "SubnetId": {
                "Ref": "PublicSubnet2"
            }
        },
    },

```

```

        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-nat-2"
                }
            }
        ]
    },
    "PublicRouteTable": {
        "Type": "AWS::EC2::RouteTable",
        "Properties": {
            "VpcId": {
                "Ref": "VPC"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": {
                        "Fn::Sub": "${AWS::StackName}-public-routes"
                    }
                }
            ]
        }
    },
    "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"
            }
        }
    },
    },
    "PrivateRouteTable1": {
        "Type": "AWS::EC2::RouteTable",
        "Properties": {
            "VpcId": {
                "Ref": "VPC"
            },
            "Tags": [
                {
                    "Key": "Name",
                    "Value": {
                        "Fn::Sub": "${AWS::StackName}-private-routes-1"
                    }
                }
            ]
        }
    },
    },
    "DefaultPrivateRoute1": {
        "Type": "AWS::EC2::Route",
        "Properties": {
            "RouteTableId": {
                "Ref": "PrivateRouteTable1"
            },
            "DestinationCidrBlock": "0.0.0.0/0",
            "NatGatewayId": {
                "Ref": "NatGateway1"
            }
        }
    }
}

```

```

    },
    "PrivateSubnet1RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {
          "Ref": "PrivateRouteTable1"
        },
        "SubnetId": {
          "Ref": "PrivateSubnet1"
        }
      }
    },
    "PrivateRouteTable2": {
      "Type": "AWS::EC2::RouteTable",
      "Properties": {
        "VpcId": {
          "Ref": "VPC"
        },
        "Tags": [
          {
            "Key": "Name",
            "Value": {
              "Fn::Sub": "${AWS::StackName}-private-routes-2"
            }
          }
        ]
      }
    },
    "DefaultPrivateRoute2": {
      "Type": "AWS::EC2::Route",
      "Properties": {
        "RouteTableId": {
          "Ref": "PrivateRouteTable2"
        },
        "DestinationCidrBlock": "0.0.0.0/0",
        "NatGatewayId": {
          "Ref": "NatGateway2"
        }
      }
    },
    "PrivateSubnet2RouteTableAssociation": {
      "Type": "AWS::EC2::SubnetRouteTableAssociation",
      "Properties": {
        "RouteTableId": {

```

```

        "Ref": "PrivateRouteTable2"
      },
      "SubnetId": {
        "Ref": "PrivateSubnet2"
      }
    }
  },
  "ALBSecurityGroup": {
    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupName": {
        "Fn::Sub": "${AWS::StackName}-alb-sg"
      },
      "GroupDescription": "Security group for Application Load Balancer",
      "VpcId": {
        "Ref": "VPC"
      },
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0",
          "Description": "Allow HTTP traffic from internet"
        }
      ],
      "SecurityGroupEgress": [
        {
          "IpProtocol": "-1",
          "CidrIp": "0.0.0.0/0",
          "Description": "Allow all outbound traffic"
        }
      ],
      "Tags": [
        {
          "Key": "Name",
          "Value": {
            "Fn::Sub": "${AWS::StackName}-alb-sg"
          }
        }
      ]
    }
  },
  "ECSSecurityGroup": {

```



```

    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupName": {
        "Fn::Sub": "${AWS::StackName}-ecs-sg"
      },
      "GroupDescription": "Security group for ECS tasks",
      "VpcId": {
        "Ref": "VPC"
      },
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": {
            "Ref": "ContainerPort"
          },
          "ToPort": {
            "Ref": "ContainerPort"
          },
          "SourceSecurityGroupId": {
            "Ref": "ALBSecurityGroup"
          },
          "Description": "Allow traffic from ALB"
        }
      ],
      "SecurityGroupEgress": [
        {
          "IpProtocol": "-1",
          "CidrIp": "0.0.0.0/0",
          "Description": "Allow all outbound traffic"
        }
      ],
      "Tags": [
        {
          "Key": "Name",
          "Value": {
            "Fn::Sub": "${AWS::StackName}-ecs-sg"
          }
        }
      ]
    }
  },
  "ApplicationLoadBalancer": {
    "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties": {

```

```

        "Name": {
            "Fn::Sub": "${AWS::StackName}-alb"
        },
        "Scheme": "internet-facing",
        "Type": "application",
        "Subnets": [
            {
                "Ref": "PublicSubnet1"
            },
            {
                "Ref": "PublicSubnet2"
            }
        ],
        "SecurityGroups": [
            {
                "Ref": "ALBSecurityGroup"
            }
        ],
        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-alb"
                }
            }
        ]
    },
    "ALBTargetGroup": {
        "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
        "Properties": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-tg"
            },
            "Port": {
                "Ref": "ContainerPort"
            },
            "Protocol": "HTTP",
            "VpcId": {
                "Ref": "VPC"
            },
            "TargetType": "ip",
            "HealthCheckIntervalSeconds": 30,
            "HealthCheckPath": "/",

```

```

        "HealthCheckProtocol": "HTTP",
        "HealthCheckTimeoutSeconds": 5,
        "HealthyThresholdCount": 2,
        "UnhealthyThresholdCount": 5,
        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-tg"
                }
            }
        ]
    },
    "ALBListener": {
        "Type": "AWS::ElasticLoadBalancingV2::Listener",
        "Properties": {
            "DefaultActions": [
                {
                    "Type": "forward",
                    "TargetGroupArn": {
                        "Ref": "ALBTargetGroup"
                    }
                }
            ],
            "LoadBalancerArn": {
                "Ref": "ApplicationLoadBalancer"
            },
            "Port": 80,
            "Protocol": "HTTP"
        }
    },
    "ECSCluster": {
        "Type": "AWS::ECS::Cluster",
        "Properties": {
            "ClusterName": {
                "Fn::Sub": "${AWS::StackName}-cluster"
            },
            "CapacityProviders": [
                "FARGATE",
                "FARGATE_SPOT"
            ],
            "DefaultCapacityProviderStrategy": [
                {

```

```

        "CapacityProvider": "FARGATE",
        "Weight": 1
    },
    {
        "CapacityProvider": "FARGATE_SPOT",
        "Weight": 4
    }
],
"ClusterSettings": [
    {
        "Name": "containerInsights",
        "Value": "enabled"
    }
],
"Tags": [
    {
        "Key": "Name",
        "Value": {
            "Fn::Sub": "${AWS::StackName}-cluster"
        }
    }
]
},
"ECSTaskExecutionRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "RoleName": {
            "Fn::Sub": "${AWS::StackName}-task-execution-role"
        },
        "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"
    ],
    "Tags": [
        {
            "Key": "Name",
            "Value": {
                "Fn::Sub": "${AWS::StackName}-task-execution-role"
            }
        }
    ]
},
"ECSTaskRole": {
    "Type": "AWS::IAM::Role",
    "Properties": {
        "RoleName": {
            "Fn::Sub": "${AWS::StackName}-task-role"
        },
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ecs-tasks.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        },
        "Tags": [
            {
                "Key": "Name",
                "Value": {
                    "Fn::Sub": "${AWS::StackName}-task-role"
                }
            }
        ]
    }
},
"LogGroup": {
    "Type": "AWS::Logs::LogGroup",
    "Properties": {

```

```

        "LogGroupName": {
            "Fn::Sub": "/ecs/${AWS::StackName}"
        },
        "RetentionInDays": 7
    }
},
"TaskDefinition": {
    "Type": "AWS::ECS::TaskDefinition",
    "Properties": {
        "Family": {
            "Fn::Sub": "${AWS::StackName}-task"
        },
        "Cpu": "256",
        "Memory": "512",
        "NetworkMode": "awsvpc",
        "RequiresCompatibilities": [
            "FARGATE"
        ],
        "ExecutionRoleArn": {
            "Fn::GetAtt": [
                "ECSTaskExecutionRole",
                "Arn"
            ]
        },
        "TaskRoleArn": {
            "Fn::GetAtt": [
                "ECSTaskRole",
                "Arn"
            ]
        },
        "ContainerDefinitions": [
            {
                "Name": {
                    "Ref": "ServiceName"
                },
                "Image": {
                    "Ref": "ContainerImage"
                },
                "PortMappings": [
                    {
                        "ContainerPort": {
                            "Ref": "ContainerPort"
                        },
                        "Protocol": "tcp"
                    }
                ]
            }
        ]
    }
}

```

```

    }
  ],
  "Essential": true,
  "LogConfiguration": {
    "LogDriver": "awslogs",
    "Options": {
      "awslogs-group": {
        "Ref": "LogGroup"
      },
      "awslogs-region": {
        "Ref": "AWS::Region"
      },
      "awslogs-stream-prefix": "ecs"
    }
  },
  "HealthCheck": {
    "Command": [
      "CMD-SHELL",
      "curl -f http://localhost/ || exit 1"
    ],
    "Interval": 30,
    "Timeout": 5,
    "Retries": 3,
    "StartPeriod": 60
  }
}
],
"Tags": [
  {
    "Key": "Name",
    "Value": {
      "Fn::Sub": "${AWS::StackName}-task"
    }
  }
]
},
"ECSService": {
  "Type": "AWS::ECS::Service",
  "DependsOn": "ALBListener",
  "Properties": {
    "ServiceName": {
      "Fn::Sub": "${AWS::StackName}-service"
    }
  },

```

```
"Cluster": {
    "Ref": "ECSCluster"
},
"TaskDefinition": {
    "Ref": "TaskDefinition"
},
"DesiredCount": {
    "Ref": "DesiredCount"
},
"LaunchType": "FARGATE",
"PlatformVersion": "LATEST",
"NetworkConfiguration": {
    "AwsvpcConfiguration": {
        "AssignPublicIp": "DISABLED",
        "SecurityGroups": [
            {
                "Ref": "ECSSecurityGroup"
            }
        ],
        "Subnets": [
            {
                "Ref": "PrivateSubnet1"
            },
            {
                "Ref": "PrivateSubnet2"
            }
        ]
    }
},
"LoadBalancers": [
    {
        "ContainerName": {
            "Ref": "ServiceName"
        },
        "ContainerPort": {
            "Ref": "ContainerPort"
        },
        "TargetGroupArn": {
            "Ref": "ALBTargetGroup"
        }
    }
],
"DeploymentConfiguration": {
    "MaximumPercent": 200,
```



```

        "MinimumHealthyPercent": 50,
        "DeploymentCircuitBreaker": {
            "Enable": true,
            "Rollback": true
        }
    },
    "EnableExecuteCommand": true,
    "Tags": [
        {
            "Key": "Name",
            "Value": {
                "Fn::Sub": "${AWS::StackName}-service"
            }
        }
    ]
}
},
"ServiceScalingTarget": {
    "Type": "AWS::ApplicationAutoScaling::ScalableTarget",
    "Properties": {
        "MaxCapacity": {
            "Ref": "MaxCapacity"
        },
        "MinCapacity": {
            "Ref": "MinCapacity"
        },
        "ResourceId": {
            "Fn::Sub": "service/${ECSCluster}/${ECSService.Name}"
        },
        "RoleARN": {
            "Fn::Sub": "arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService"
        },
        "ScalableDimension": "ecs:service:DesiredCount",
        "ServiceNamespace": "ecs"
    }
},
"ServiceScalingPolicy": {
    "Type": "AWS::ApplicationAutoScaling::ScalingPolicy",
    "Properties": {
        "PolicyName": {
            "Fn::Sub": "${AWS::StackName}-cpu-scaling-policy"
        }
    },

```

```

        "PolicyType": "TargetTrackingScaling",
        "ScalingTargetId": {
            "Ref": "ServiceScalingTarget"
        },
        "TargetTrackingScalingPolicyConfiguration": {
            "PredefinedMetricSpecification": {
                "PredefinedMetricType": "ECSServiceAverageCPUUtilization"
            },
            "TargetValue": 70,
            "ScaleOutCooldown": 300,
            "ScaleInCooldown": 300
        }
    }
},
"Outputs": {
    "VPCId": {
        "Description": "VPC ID",
        "Value": {
            "Ref": "VPC"
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-VPC-ID"
            }
        }
    },
    "LoadBalancerURL": {
        "Description": "URL of the Application Load Balancer",
        "Value": {
            "Fn::Sub": "http://${ApplicationLoadBalancer.DNSName}"
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-ALB-URL"
            }
        }
    },
    "ECSClusterName": {
        "Description": "Name of the ECS Cluster",
        "Value": {
            "Ref": "ECSCluster"
        },
        "Export": {

```

```

        "Name": {
            "Fn::Sub": "${AWS::StackName}-ECS-Cluster"
        }
    },
    "ECSServiceName": {
        "Description": "Name of the ECS Service",
        "Value": {
            "Fn::GetAtt": [
                "ECSService",
                "Name"
            ]
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-ECS-Service"
            }
        }
    },
    "PrivateSubnet1": {
        "Description": "Private Subnet 1 ID",
        "Value": {
            "Ref": "PrivateSubnet1"
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-Private-Subnet-1"
            }
        }
    },
    "PrivateSubnet2": {
        "Description": "Private Subnet 2 ID",
        "Value": {
            "Ref": "PrivateSubnet2"
        },
        "Export": {
            "Name": {
                "Fn::Sub": "${AWS::StackName}-Private-Subnet-2"
            }
        }
    }
}

```

YAML

```
AWSTemplateFormatVersion: '2010-09-09'
Description: '[AWSDocs] ECS: load-balanced-web-application'

Parameters:
  VpcCidr:
    Type: String
    Default: '10.0.0.0/16'
    Description: CIDR block for the VPC
  ContainerImage:
    Type: String
    Default: 'public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest'
    Description: Container image to use in task definition

  PublicSubnet1Cidr:
    Type: String
    Default: '10.0.1.0/24'
    Description: CIDR block for public subnet 1

  PublicSubnet2Cidr:
    Type: String
    Default: '10.0.2.0/24'
    Description: CIDR block for public subnet 2

  PrivateSubnet1Cidr:
    Type: String
    Default: '10.0.3.0/24'
    Description: CIDR block for private subnet 1

  PrivateSubnet2Cidr:
    Type: String
    Default: '10.0.4.0/24'
    Description: CIDR block for private subnet 2

  ServiceName:
    Type: String
    Default: 'tutorial-app'
    Description: Name of the ECS service

  ContainerPort:
    Type: Number
    Default: 80
    Description: Port on which the container listens
```

```
DesiredCount:
  Type: Number
  Default: 2
  Description: Desired number of tasks

MinCapacity:
  Type: Number
  Default: 1
  Description: Minimum number of tasks for auto scaling

MaxCapacity:
  Type: Number
  Default: 10
  Description: Maximum number of tasks for auto scaling
```

```
Resources:
  # VPC and Networking
  VPC:
    Type: AWS::EC2::VPC
    Properties:
      CidrBlock: !Ref VpcCidr
      EnableDnsHostnames: true
      EnableDnsSupport: true
      Tags:
        - Key: Name
          Value: !Sub '${AWS::StackName}-vpc'

  # Internet Gateway
  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

  # Public Subnets for ALB
  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}-public-subnet-1'

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}-public-subnet-2'

# Private Subnets for ECS Tasks
PrivateSubnet1:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [0, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet1Cidr
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-subnet-1'

PrivateSubnet2:
  Type: AWS::EC2::Subnet
  Properties:
    VpcId: !Ref VPC
    AvailabilityZone: !Select [1, !GetAZs '']
    CidrBlock: !Ref PrivateSubnet2Cidr
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-subnet-2'

# NAT Gateways for private subnet internet access
NatGateway1EIP:
```

```
Type: AWS::EC2::EIP
DependsOn: InternetGatewayAttachment
Properties:
  Domain: vpc
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-nat-eip-1'
```

```
NatGateway2EIP:
  Type: AWS::EC2::EIP
  DependsOn: InternetGatewayAttachment
  Properties:
    Domain: vpc
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-eip-2'
```

```
NatGateway1:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway1EIP.AllocationId
    SubnetId: !Ref PublicSubnet1
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-1'
```

```
NatGateway2:
  Type: AWS::EC2::NatGateway
  Properties:
    AllocationId: !GetAtt NatGateway2EIP.AllocationId
    SubnetId: !Ref PublicSubnet2
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-nat-2'
```

```
# Route Tables
PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-public-routes'
```

```
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

PrivateRouteTable1:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-private-routes-1'

DefaultPrivateRoute1:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable1
    SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:
  Type: AWS::EC2::RouteTable
  Properties:
```



```

VpcId: !Ref VPC
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-private-routes-2'

DefaultPrivateRoute2:
  Type: AWS::EC2::Route
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    DestinationCidrBlock: 0.0.0.0/0
    NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:
  Type: AWS::EC2::SubnetRouteTableAssociation
  Properties:
    RouteTableId: !Ref PrivateRouteTable2
    SubnetId: !Ref PrivateSubnet2

# Security Groups
ALBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: !Sub '${AWS::StackName}-alb-sg'
    GroupDescription: Security group for Application Load Balancer
    VpcId: !Ref VPC
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 80
        ToPort: 80
        CidrIp: 0.0.0.0/0
        Description: Allow HTTP traffic from internet
    SecurityGroupEgress:
      - IpProtocol: -1
        CidrIp: 0.0.0.0/0
        Description: Allow all outbound traffic
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-alb-sg'

ECSSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupName: !Sub '${AWS::StackName}-ecs-sg'
    GroupDescription: Security group for ECS tasks

```

```

VpcId: !Ref VPC
SecurityGroupIngress:
  - IpProtocol: tcp
    FromPort: !Ref ContainerPort
    ToPort: !Ref ContainerPort
    SourceSecurityGroupId: !Ref ALBSecurityGroup
    Description: Allow traffic from ALB
SecurityGroupEgress:
  - IpProtocol: -1
    CidrIp: 0.0.0.0/0
    Description: Allow all outbound traffic
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-ecs-sg'

```

Application Load Balancer

```

ApplicationLoadBalancer:
  Type: AWS::ElasticLoadBalancingV2::LoadBalancer
  Properties:
    Name: !Sub '${AWS::StackName}-alb'
    Scheme: internet-facing
    Type: application
    Subnets:
      - !Ref PublicSubnet1
      - !Ref PublicSubnet2
    SecurityGroups:
      - !Ref ALBSecurityGroup
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-alb'

```

```

ALBTargetGroup:
  Type: AWS::ElasticLoadBalancingV2::TargetGroup
  Properties:
    Name: !Sub '${AWS::StackName}-tg'
    Port: !Ref ContainerPort
    Protocol: HTTP
    VpcId: !Ref VPC
    TargetType: ip
    HealthCheckIntervalSeconds: 30
    HealthCheckPath: /
    HealthCheckProtocol: HTTP
    HealthCheckTimeoutSeconds: 5
    HealthyThresholdCount: 2

```

```
UnhealthyThresholdCount: 5
Tags:
  - Key: Name
    Value: !Sub '${AWS::StackName}-tg'

ALBListener:
  Type: AWS::ElasticLoadBalancingV2::Listener
  Properties:
    DefaultActions:
      - Type: forward
        TargetGroupArn: !Ref ALBTargetGroup
    LoadBalancerArn: !Ref ApplicationLoadBalancer
    Port: 80
    Protocol: HTTP

# ECS Cluster
ECSCluster:
  Type: AWS::ECS::Cluster
  Properties:
    ClusterName: !Sub '${AWS::StackName}-cluster'
    CapacityProviders:
      - FARGATE
      - FARGATE_SPOT
    DefaultCapacityProviderStrategy:
      - CapacityProvider: FARGATE
        Weight: 1
      - CapacityProvider: FARGATE_SPOT
        Weight: 4
    ClusterSettings:
      - Name: containerInsights
        Value: enabled
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-cluster'

# IAM Roles
ECSTaskExecutionRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub '${AWS::StackName}-task-execution-role'
    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
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-task-execution-role'

ECSTaskRole:
  Type: AWS::IAM::Role
  Properties:
    RoleName: !Sub '${AWS::StackName}-task-role'
    AssumeRolePolicyDocument:
      Version: '2012-10-17'
      Statement:
        - Effect: Allow
          Principal:
            Service: ecs-tasks.amazonaws.com
          Action: sts:AssumeRole
    Tags:
      - Key: Name
        Value: !Sub '${AWS::StackName}-task-role'

# CloudWatch Log Group
LogGroup:
  Type: AWS::Logs::LogGroup
  Properties:
    LogGroupName: !Sub '/ecs/${AWS::StackName}'
    RetentionInDays: 7

# ECS Task Definition
TaskDefinition:
  Type: AWS::ECS::TaskDefinition
  Properties:
    Family: !Sub '${AWS::StackName}-task'
    Cpu: '256'
    Memory: '512'
    NetworkMode: awsvpc
    RequiresCompatibilities:
      - FARGATE
    ExecutionRoleArn: !GetAtt ECSTaskExecutionRole.Arn
    TaskRoleArn: !GetAtt ECSTaskRole.Arn
    ContainerDefinitions:

```

```

- Name: !Ref ServiceName
  Image: !Ref ContainerImage
  PortMappings:
    - ContainerPort: !Ref ContainerPort
      Protocol: tcp
  Essential: true
  LogConfiguration:
    LogDriver: awslogs
    Options:
      awslogs-group: !Ref LogGroup
      awslogs-region: !Ref AWS::Region
      awslogs-stream-prefix: ecs
  HealthCheck:
    Command:
      - CMD-SHELL
      - curl -f http://localhost/ || exit 1
    Interval: 30
    Timeout: 5
    Retries: 3
    StartPeriod: 60
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-task'

```

ECS Service

ECSService:

Type: AWS::ECS::Service

DependsOn: ALBListener

Properties:

ServiceName: !Sub '\${AWS::StackName}-service'

Cluster: !Ref ECSCluster

TaskDefinition: !Ref TaskDefinition

DesiredCount: !Ref DesiredCount

LaunchType: FARGATE

PlatformVersion: LATEST

NetworkConfiguration:

AwsvpcConfiguration:

AssignPublicIp: DISABLED

SecurityGroups:

- !Ref ECSSecurityGroup

Subnets:

- !Ref PrivateSubnet1

- !Ref PrivateSubnet2

LoadBalancers:

```

    - ContainerName: !Ref ServiceName
      ContainerPort: !Ref ContainerPort
      TargetGroupArn: !Ref ALBTargetGroup
  DeploymentConfiguration:
    MaximumPercent: 200
    MinimumHealthyPercent: 50
    DeploymentCircuitBreaker:
      Enable: true
      Rollback: true
  EnableExecuteCommand: true # For debugging
  Tags:
    - Key: Name
      Value: !Sub '${AWS::StackName}-service'

# Auto Scaling Target
ServiceScalingTarget:
  Type: AWS::ApplicationAutoScaling::ScalableTarget
  Properties:
    MaxCapacity: !Ref MaxCapacity
    MinCapacity: !Ref MinCapacity
    ResourceId: !Sub 'service/${ECSCluster}/${ECSService.Name}'
    RoleARN: !Sub 'arn:aws:iam::${AWS::AccountId}:role/
aws-service-role/ecs.application-autoscaling.amazonaws.com/
AWSServiceRoleForApplicationAutoScaling_ECSService'
    ScalableDimension: ecs:service:DesiredCount
    ServiceNamespace: ecs

# Auto Scaling Policy - CPU Utilization
ServiceScalingPolicy:
  Type: AWS::ApplicationAutoScaling::ScalingPolicy
  Properties:
    PolicyName: !Sub '${AWS::StackName}-cpu-scaling-policy'
    PolicyType: TargetTrackingScaling
    ScalingTargetId: !Ref ServiceScalingTarget
    TargetTrackingScalingPolicyConfiguration:
      PredefinedMetricSpecification:
        PredefinedMetricType: ECSServiceAverageCPUUtilization
      TargetValue: 70.0
      ScaleOutCooldown: 300
      ScaleInCooldown: 300

Outputs:
  VPCId:
    Description: VPC ID

```

```
Value: !Ref VPC
Export:
  Name: !Sub '${AWS::StackName}-VPC-ID'

LoadBalancerURL:
  Description: URL of the Application Load Balancer
  Value: !Sub 'http://${ApplicationLoadBalancer.DNSName}'
  Export:
    Name: !Sub '${AWS::StackName}-ALB-URL '

ECSClusterName:
  Description: Name of the ECS Cluster
  Value: !Ref ECSCluster
  Export:
    Name: !Sub '${AWS::StackName}-ECS-Cluster'

ECSServiceName:
  Description: Name of the ECS Service
  Value: !GetAtt ECSService.Name
  Export:
    Name: !Sub '${AWS::StackName}-ECS-Service'

PrivateSubnet1:
  Description: Private Subnet 1 ID
  Value: !Ref PrivateSubnet1
  Export:
    Name: !Sub '${AWS::StackName}-Private-Subnet-1'

PrivateSubnet2:
  Description: Private Subnet 2 ID
  Value: !Ref PrivateSubnet2
  Export:
    Name: !Sub '${AWS::StackName}-Private-Subnet-2'
```

Distribuisci un servizio con l'esecutivo ECS abilitato

Per distribuire un servizio con l'esecutivo ECS abilitato, puoi utilizzare il modello seguente. Il servizio viene eseguito in un cluster con una chiave KMS per la crittografia delle sessioni dell'esecutivo ECS e una configurazione di log per il reindirizzamento dei log delle sessioni del comando di esecuzione verso un bucket Amazon S3. Per ulteriori informazioni, consulta [Monitora i container 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

Distribuisci il servizio che utilizza Amazon VPC Lattice

Puoi utilizzare il seguente modello per iniziare a creare un servizio Amazon ECS con VPC Lattice. Per configurare VPC Lattice, potresti dover completare le fasi aggiuntive seguenti:

- Aggiorna le regole in entrata del gruppo di sicurezza per VPC Lattice per consentire il prefisso `vpc-lattice` delle 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 dei listener in un servizio VPC Lattice.
- Verifica le configurazioni per il controllo dell'integrità 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": [],
  "EnableECSManagedTags": 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

```

Distribuisci il servizio con una configurazione dei volumi

Il modello seguente include una configurazione dei volumi nella definizione del servizio. Amazon ECS supporta la configurazione dei seguenti volumi di dati utilizzando una configurazione dei volumi al momento dell'avvio: volumi Amazon EBS. Per ulteriori informazioni sui volumi Amazon EBS, consulta [Usare 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": [],
  "EnableECSManagedTags": 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: []
    EnableECSManagedTags: 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

```

Distribuisci il servizio con i provider di capacità

Il modello seguente definisce un servizio che utilizza il provider di capacità per richiedere la capacità 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 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 seguente modello può essere utilizzato per creare un semplice ruolo di esecuzione delle attività che utilizza la policy 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"

```

Best practice di Amazon ECS

Puoi utilizzare una delle seguenti pagine per scoprire le best practice operative più importanti per il networking di Amazon ECS.

Panoramica delle best practice	Ulteriori informazioni	
Connetti le applicazioni a Internet	Connetti le applicazioni Amazon ECS a Internet	
Ricevi connessioni in entrata per 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 su e Account AWS VPCs	Best practice per il collegamento in rete dei servizi Amazon ECS tra e Account AWS VPCs	
Risoluzione dei problemi di rete	AWS servizi per la risoluzione dei problemi di rete Amazon ECS	

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

Panoramica delle best practice	Ulteriori informazioni	
Sicurezza di Fargate	Best practice sulla sicurezza di Fargate in Amazon ECS	

Panoramica delle best practice	Ulteriori informazioni	
Considerazioni relative alla sicurezza di Fargate	Considerazioni relative sicurezza di Fargate per Amazon ECS	
Comportamento dell'estrazione delle immagini dei container Linux sui container Fargate	Comportamento dell'estrazione delle immagini dei container Linux sui container Fargate per Amazon ECS	
Comportamento dell'estrazione delle immagini dei container Windows sui container Fargate	Comportamento dell'estrazione delle immagini dei container Windows sui container Fargate per Amazon ECS	
Ritiro delle attività di Fargate	Ritiro e manutenzione delle attività per AWS Fargate su Amazon ECS	

Puoi utilizzare una delle seguenti pagine per scoprire le best practice operative più importanti per le definizioni delle attività.

Panoramica delle best practice	Ulteriori informazioni	
Immagini di container	Best practice per le immagini dei container Amazon ECS	
Dimensioni processo	Best practice per le dimensioni delle attività di Amazon ECS	
Best practice per il volume	Opzioni di archiviazione per le attività di Amazon ECS	

Puoi utilizzare una delle seguenti pagine per scoprire le best practice operative più importanti per i cluster e la capacità.

Panoramica delle best practice	Ulteriori informazioni	
Sicurezza di Fargate	Best practice sulla sicurezza di Fargate in Amazon ECS	
EC2 considerazioni sulla sicurezza delle istanze di container	Considerazioni sulla sicurezza delle istanze di EC2 container Amazon per Amazon ECS	
Scalabilità automatica del cluster	Ottimizzazione del dimensionamento automatico del cluster Amazon ECS	
Utilizzo su larga scala	Utilizzo di Amazon ECS su larga scala	
Dimensionamento automatico e gestione della capacità	Best practice di dimensionamento automatico e gestione della capacità di Amazon ECS	
Best practice per la selezione delle istanze per Amazon ECS Managed Instances	Best practice per la selezione delle istanze per le Istanze gestite da Amazon ECS	

Puoi utilizzare una delle seguenti pagine per scoprire le best practice operative più importanti per le attività e i servizi.

Panoramica delle best practice	Ulteriori informazioni	
Ottimizzazione dei tempi di avvio delle attività	Ottimizza i tempi di avvio delle attività di Amazon ECS	
Parametri dei servizi	Best practice per i parametri del servizio di Amazon ECS	
Ottimizzazione dei parametri di controllo dell'integrità del bilanciamento del carico	Ottimizzare i parametri di controllo dell'integrità del	

Panoramica delle best practice	Ulteriori informazioni	
	<u>bilanciatore del carico per Amazon ECS</u>	
Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore del carico	<u>Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore del carico per Amazon ECS</u>	
Ottimizzazione del dimensionamento automatico dei servizi	<u>Ottimizzazione del dimensionamento automatico del servizio Amazon ECS</u>	

Puoi utilizzare una delle seguenti pagine per scoprire le best practice operative più importanti per la sicurezza.

Panoramica delle best practice	Ulteriori informazioni	
Sicurezza di rete	<u>Best practice sulla sicurezza di rete per Amazon ECS</u>	
Sicurezza delle attività e dei container	<u>Best practice per la sicurezza di attività e container di Amazon ECS</u>	

Progetta la tua soluzione per Amazon ECS

È necessario progettare le applicazioni in modo che possano essere eseguite su container. 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. Un'immagine del container è un artefatto statico contenente l'applicazione e le sue dipendenze. 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 container. 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 un blueprint 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

Capacity

Capacità: l'infrastruttura su cui vengono eseguiti i container Amazon ECS offre tre tipi di infrastruttura per i tuoi cluster:

- Amazon ECS Managed Instances: gestisce AWS completamente le istanze Amazon EC2 sottostanti in esecuzione nel tuo account, inclusi il provisioning, l'applicazione di patch e la scalabilità. Questa opzione offre un equilibrio ottimale tra prestazioni, convenienza economica e semplicità operativa.
- Serverless (Fargate): paghi solo per le risorse utilizzate dalle tue attività senza gestire alcuna infrastruttura. Ideale per carichi di lavoro variabili e per iniziare rapidamente.
- EC2 Istanze Amazon: gestisci direttamente le EC2 istanze Amazon sottostanti, inclusa la selezione, la configurazione e la manutenzione delle istanze.

Usa le istanze gestite da Amazon ECS quando:

- Desideri la semplicità di Fargate con un maggiore controllo sull'infrastruttura sottostante.
- Hai bisogno di prestazioni prevedibili e ottimizzazione dei costi.
- Vuoi AWS gestire la gestione dell'infrastruttura mantenendo la flessibilità.

Usa Fargate quando:

- Vuoi concentrarti sulla tua applicazione senza gestire l'infrastruttura.
- Hai carichi di lavoro variabili o imprevedibili.
- Stai iniziando con i container e desideri l'opzione di implementazione più semplice.

Usa le EC2 istanze Amazon quando:

- Necessiti di requisiti hardware specializzati (accelerazione GPU, elaborazione ad alte prestazioni).
- Necessiti di prenotazioni della capacità o tipi di istanze specifici.
- Hai bisogno di funzionalità privilegiate o personalizzate. AMIs

Specifica l'infrastruttura quando crei un cluster. Quando registri una definizione di attività, devi specificare anche il tipo di infrastruttura. La definizione dell'attività si riferisce all'infrastruttura come al "tipo di avvio". Puoi anche utilizzare fornitori di capacità.

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 ulteriori informazioni sugli 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 oppure on-premises, 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 la sottorete può trovarsi in una delle seguenti risorse: AWS

- Zone di disponibilità
- Local Zones
- Zone Wavelength
- Regioni AWS
- AWS Outposts

Per ulteriori informazioni, consulta [Applicazioni Amazon ECS in sottoreti condivise, zone locali e zone Wavelength](#) o [Amazon Elastic Container Service su AWS Outposts](#).

È possibile collegare 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 le sottoreti private quando desideri proteggere i container dall'accesso esterno diretto. Gli scenari applicabili includono sistemi di elaborazione dei pagamenti o container che archiviano dati e password degli utenti.

- AWS PrivateLink

AWS PrivateLink Utilizzala per avere una connettività privata tra VPCs i AWS servizi e le reti locali senza esporre il traffico alla rete Internet pubblica.

Accesso alle funzionalità

Puoi utilizzare le impostazioni dell'account Amazon ECS per accedere alle seguenti funzionalità:

- Container Insights

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. I parametri includono l'utilizzo di risorse come CPU, memoria, dischi e rete.

- Trunking awsvpc

Per determinati tipi di EC2 istanze, puoi avere interfacce di rete aggiuntive () disponibili sulle istanze di container appena lanciate. ENIs

- Autorizzazione all'assegnazione di tag

Gli utenti devono disporre dell'autorizzazione per le operazioni che creano la 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 `ecs:TagResource` ruoli dispongono delle 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 ritiro delle attività Fargate

Puoi configurare il periodo di attesa prima che le attività Fargate vengano ritirate per l'applicazione di patch.

- VPC dual stack

Consenti alle attività di comunicare più IPv4 o meno entrambe. IPv6

- Formato Amazon Resource Name (ARN)

Alcune funzionalità, ad esempio l'autorizzazione all'assegnazione dei tag, richiedono un nuovo formato del Nome della risorsa Amazon (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 container o servizi.

Alcune funzionalità di Amazon ECS richiedono dei ruoli. Per ulteriori informazioni, consulta [Ruoli IAM per Amazon ECS](#).

Registrazione dei log

I log e il monitoraggio sono importanti per garantire l'affidabilità, la disponibilità e le prestazioni dei carichi di lavoro di Amazon ECS. Sono disponibili le seguenti opzioni:

- Amazon CloudWatch logs: 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 per i clienti.

Best Practice per l'immagine dei container

Di seguito sono riportati i principi chiave per le immagini dei container Amazon ECS:

- Includere tutte le dipendenze nell'immagine
- Eseguire un processo per ciascun container
- Gestire SIGTERM per spegnimenti senza intoppi
- Scrivere i log su `stdout` e `stderr`
- Utilizzare tag univoci, evitare `latest` nella produzione

Tipi di lancio e fornitori di capacità di Amazon Amazon ECS

Amazon ECS offre due metodi per configurare la capacità per i carichi di lavoro. Puoi utilizzare tipi di lancio o fornitori di capacità. I tipi di lancio includono EC2 Fargate ed External. I provider di capacità offrono una maggiore flessibilità e funzionalità avanzate per la gestione della capacità. Puoi eseguire carichi di lavoro su elaborazione serverless con i provider di capacità Fargate e Fargate Spot, su istanze autogestite tramite i provider di capacità di gruppo EC2 Auto Scaling o su elaborazione completamente gestita utilizzando i provider di capacità di Amazon ECS Managed Instances che combinano la semplicità di Fargate con la flessibilità di elaborazione. EC2 I fornitori di capacità offrono un migliore controllo sull'allocazione delle risorse e possono aiutare a ottimizzare sia le prestazioni che i costi. I provider di capacità sono il modo consigliato per configurare la capacità per i carichi di lavoro rispetto ai tipi di lancio tradizionali. Utilizza quanto segue per comprendere le differenze tra i provider di capacità e i tipi di lancio.

Best practice

Di seguito sono riportate le best practice:

Utilizza i tipi di lancio per definire la compatibilità dell'infrastruttura

I tipi di avvio definiscono l'infrastruttura su cui vengono eseguite le attività e i servizi. Quando definisci le attività, specifica `RequiresCompatibilities` di includere uno o più tipi di avvio compatibili con le attività. Puoi utilizzare i seguenti tipi di avvio: Fargate EC2, External e Amazon ECS Managed Instances. Sebbene sia possibile utilizzare il tipo di avvio anche per eseguire attività o servizi, consigliamo di utilizzare il tipo di avvio solo per definire le compatibilità nelle definizioni delle attività e di utilizzare fornitori di capacità per avviare attività o servizi. Tieni presente che puoi scegliere uno o più tipi di avvio per definire le compatibilità per le attività.

Utilizza i provider di capacità per configurare la capacità di calcolo

Quando avvii attività o servizi, configura una strategia per i provider di capacità. Amazon ECS supporta i seguenti provider di capacità: Fargate e FARGATE_SPOT, gruppi Auto Scaling EC2 per istanze autogestite e Amazon ECS Managed Instances. Tieni presente che Spot Fleet è disponibile solo come fornitore di capacità e non come tipo di lancio. Puoi creare uno o più fornitori di capacità di Amazon ECS Managed Instances o Auto Scaling group in un cluster. I provider di capacità Fargate e Fargate Spot vengono creati e gestiti da Amazon ECS su ogni cluster e non è necessario crearli. Un cluster può avere una combinazione di tutti i tipi di provider di capacità, tuttavia una strategia di provider di capacità non può includere una combinazione di diversi tipi di provider di capacità.

Aggiornare la capacità dei servizi

È sufficiente aggiornare la strategia di un provider di capacità per un servizio per spostarlo da un tipo di elaborazione all'altro.

Mutabilità del servizio

Amazon ECS supporta i servizi di aggiornamento tra diversi fornitori di capacità. Ciò consente di:

- Aggiornamento senza interruzioni dai tipi di avvio ai provider di capacità
- Transizioni tra diversi tipi di provider di capacità
- Test di diverse opzioni di elaborazione senza ricreare un servizio

Di seguito è riportata una panoramica di alto livello del processo:

1. Aggiorna la definizione dell'attività: assicurati che `requiresCompatibilities` includa il provider di capacità di destinazione, ad esempio `MANAGED_INSTANCES`.

Note

Le definizioni delle attività devono superare la convalida della compatibilità per il provider di capacità di destinazione. Se il `requiresCompatibilities` controllo della versione della definizione dell'attività ha esito negativo, la `UpdateService` chiamata ha esito negativo.

2. Crea un provider di capacità: se utilizzi gruppi Amazon EC2 Auto Scaling personalizzati, crea il provider di capacità.
3. Aggiorna il servizio: modifica il servizio per utilizzare una strategia di capacity provider anziché il tipo di avvio.
4. Convalida la distribuzione: conferma che le attività siano state distribuite correttamente.
5. Monitoraggio e ottimizzazione: modifica le impostazioni del provider di capacità in base alle esigenze.

Da fornitore di capacità a fornitore di capacità

Sono supportati tutti gli aggiornamenti da fornitore di capacità a fornitore di capacità:

- Fornitore di capacità di gruppo Amazon EC2 Auto Scaling per istanze gestite Amazon ECS
- Fornitore di capacità Fargate per Amazon ECS Managed Instances
- Da fornitore di capacità di gruppo Amazon EC2 Auto Scaling a fornitore di capacità Fargate
- Da istanze gestite da Amazon ECS al provider di capacità Fargate
- Da fornitore di capacità Fargate a fornitore di capacità del gruppo Amazon Auto EC2 Scaling
- Da Amazon ECS Managed Instances al provider di capacità di gruppo Amazon Auto EC2 Scaling

Da tipo di avvio a fornitore di capacità

Sono supportati tutti gli aggiornamenti dal tipo di avvio al provider di capacità:

- EC2 tipo di avvio per Amazon ECS Managed Instances
- Tipo di lancio Fargate per Amazon ECS Managed Instances
- EC2 tipo di avvio al provider di capacità Fargate
- EC2 tipo di avvio al provider di EC2 capacità del gruppo Auto Scaling
- Tipo di lancio di Fargate al fornitore di capacità del gruppo Amazon Auto EC2 Scaling
- Tipo di lancio da Fargate a Fargate Capacity Provider
- Tipo di avvio esterno per Amazon ECS Managed Instances
- Tipo di lancio esterno al provider di capacità Fargate
- Tipo di avvio esterno al provider di capacità del gruppo Amazon EC2 Auto Scaling

Da tipo di avvio a tipo di avvio

Gli aggiornamenti da tipo di avvio a tipo di avvio non sono supportati:

- EC2 dal tipo di avvio al tipo di avvio Fargate (utilizzare invece il provider di capacità Fargate)
- Da tipo di avvio Fargate a tipo di EC2 avvio (utilizza invece il provider di capacità EC2 di gruppo Amazon Auto Scaling)

Invece di migrare tra tipi di avvio, esegui la migrazione al provider di capacità equivalente per funzionalità avanzate e compatibilità future.

Note

Le definizioni delle attività devono superare la convalida della compatibilità per il provider di capacità di destinazione. Se il `requiresCompatibilities` controllo della versione della definizione dell'attività ha esito negativo, la `UpdateService` chiamata avrà esito negativo.

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 degli endpoint Amazon ECS, consulta [Endpoint e quote Amazon ECS](#) in Riferimenti generali di AWS.

Quando utilizzi la REST API puoi accedere direttamente a un endpoint Amazon ECS utilizzando il nome dell'endpoint (URI). Amazon ECS supporta solo i nomi di endpoint dual-stack regionali, il che significa che è necessario specificare la regione come parte del nome.

I nomi degli endpoint dual-stack usano la seguente convenzione di denominazione:

`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 sostituzione dell'endpoint Amazon ECS nel file di configurazione.

Le sezioni seguenti descrivono come utilizzare gli endpoint dual-stack provenienti da, the AWS CLI e dall'API REST. AWS SDKs

Argomenti

- [Utilizzo degli endpoint dual-stack dalla AWS CLI](#)
- [Utilizzo di endpoint dual-stack provenienti da AWS SDKs](#)
- [Utilizzo degli endpoint dual-stack dall'API REST](#)

Utilizzo degli endpoint dual-stack dalla 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 alla versione più recente 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. La regione va specificata nel file `config` o in un comando tramite l'opzione `--region`. Per ulteriori informazioni sui file di configurazione per AWS CLI, consulta [Impostazioni dei file di configurazione e credenziali nella Guida per l'utente per la versione 2](#). AWS CLI AWS Command Line Interface

Se desideri utilizzare un endpoint dual-stack per AWS CLI comandi specifici, puoi utilizzare uno dei seguenti metodi:

- Puoi utilizzare l'endpoint dual-stack per comando, impostando il parametro `--endpoint-url` su `https://ecs.aws-region.api.aws` o `http://ecs.aws-region.api.aws` per qualsiasi comando `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
```

- Puoi impostare profili separati nel tuo 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

Gli esempi seguenti mostrano come specificare un endpoint per la Regione us-east-1 utilizzando 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

Gli esempi seguenti mostrano come specificare un endpoint per la Regione us-east-1 utilizzando 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 la REST API 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, zone locali e zone Wavelength

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 for EC2 tenendo conto delle seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarlo per risorse Amazon ECS.
- Non è possibile utilizzare il gruppo di sicurezza predefinito per il VPC per le istanze di container, in quanto questo appartiene al proprietario. Inoltre, i partecipanti non possono avviare le istanze utilizzando i gruppi di sicurezza appartenenti ad altri partecipanti o al 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 for Fargate con le seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarlo per risorse Amazon ECS.
- Non è possibile creare un servizio o lanciare un'attività utilizzando il gruppo di sicurezza predefinito per il VPC, in quanto questo appartiene al proprietario. Inoltre, i partecipanti non possono creare un servizio o lanciare un'attività utilizzando i gruppi di sicurezza appartenenti ad altri partecipanti o al 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.

Local Zones

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 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 in 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 le considerazioni sull'utilizzo di Amazon ECS in 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 indicati i prerequisiti per l'utilizzo di Amazon ECS in 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 all' AWS Outposts.

Questa è la sottorete in cui vengono eseguite le istanze di container.

5. Crea un gruppo di sicurezza per le istanze di container nel tuo cluster.
6. Crea un cluster Amazon ECS.
7. Definisci le variabili di ambiente dell'agente del container di Amazon ECS per avviare l'istanza nel cluster.
8. Esegui un container.

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://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-xxxxxxx \
  --outpost-arn arn:aws:outposts:us-west-2:123456789012:outpost/op-
  xxxxxxxxxxxxxxxxx \
  --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.

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 apporti modifiche nell'immagine del container, crea una nuova immagine del container con le modifiche apportate.

- 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 gestisca SIGTERM.

Quando Amazon ECS arresta un'attività invia prima un segnale SIGTERM all'attività per notificare all'applicazione che deve essere terminata e chiusa. Amazon ECS invia quindi un messaggio SIGKILL. Quando le applicazioni ignorano il segnale SIGTERM, il servizio Amazon ECS deve attendere l'invio del segnale SIGKILL per terminare il processo.

Devi identificare il tempo necessario all'applicazione per completare il lavoro e assicurarti che le applicazioni gestiscano il segnale SIGTERM. La gestione dei segnali dell'applicazione deve impedire all'applicazione di eseguire nuovi valori e completare quelli in corso, oppure salvare il lavoro incompiuto in uno spazio di archiviazione esterno all'attività nel caso in cui il completamento richieda troppo tempo.

- Configura le applicazioni containerizzate per la scrittura di log su `stdout` e `stderr`.

Separare la gestione dei log dal codice dell'applicazione ti consente una maggiore flessibilità per adattare la gestione dei log a livello di infrastruttura. Un esempio è la modifica del sistema di log. Invece di modificare i servizi, creare e distribuire una nuova immagine del container, 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 al contrario il caricamento di una nuova immagine 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 container 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 container separatamente.

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 best practice 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 su Account AWS e VPCs, consulta [Best practice per il collegamento in rete dei servizi Amazon ECS tra e Account AWS VPCs](#).

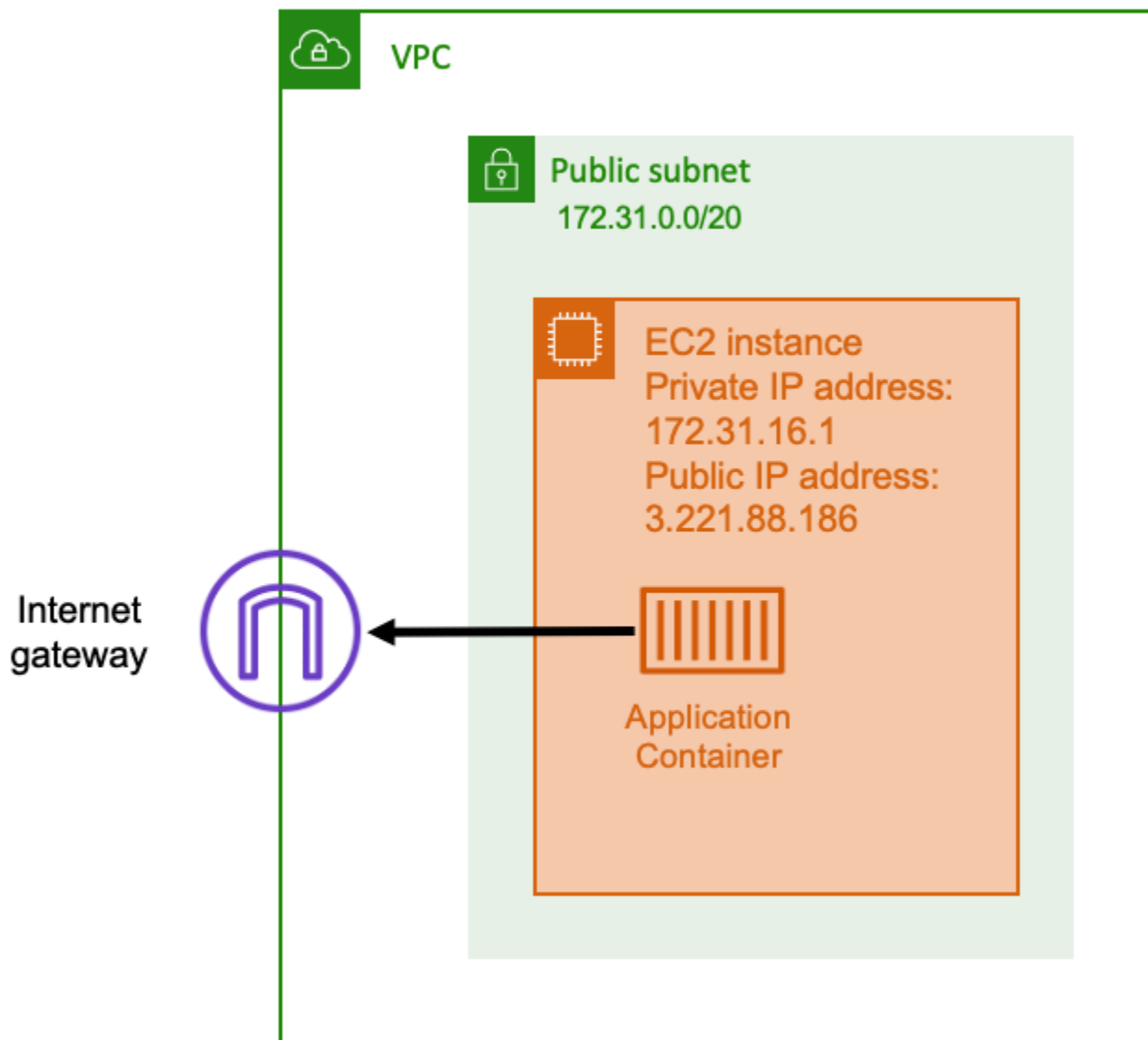
Per informazioni sulle best practice per i servizi per la risoluzione dei problemi di rete, consulta [AWS servizi per la risoluzione dei problemi di rete Amazon ECS](#).

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 per un'app mobile richiede l'accesso in uscita alle notifiche push.

Il cloud privato di Amazon Virtual offre due metodi principali per facilitare la comunicazione tra il tuo VPC e Internet.

Sottorete pubblica e gateway Internet



Quando utilizzi una sottorete pubblica con un routing a un gateway Internet, l'applicazione containerizzata può essere eseguita su un host all'interno di un VPC su una sottorete pubblica. All'host che esegue il container viene assegnato un indirizzo IP pubblico. Questo indirizzo IP pubblico è instradabile da Internet. Per ulteriori informazioni, consulta [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 che vengano aperte.

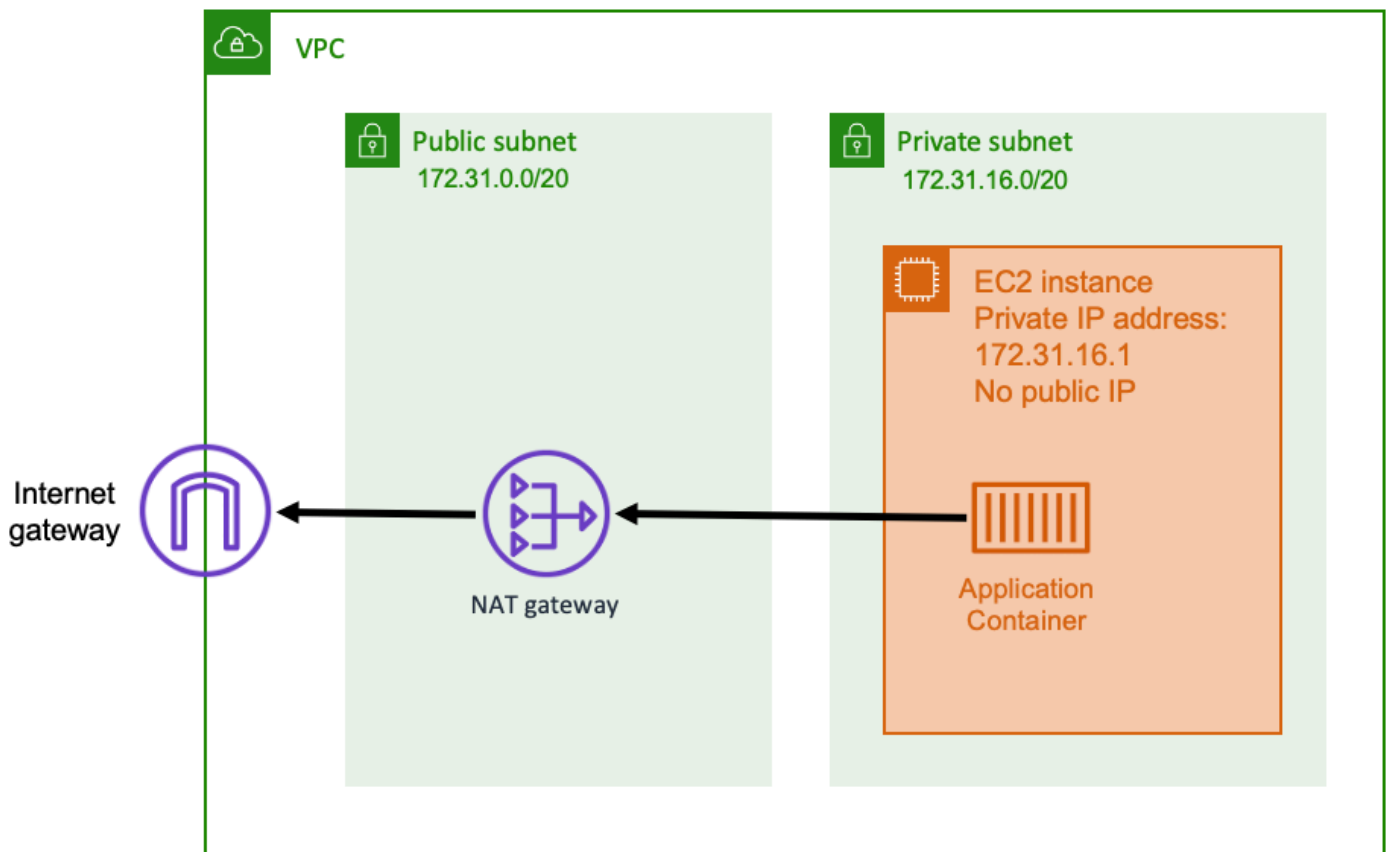
Ad esempio, se la tua applicazione viene eseguita su Amazon EC2, assicurati che la porta 22 per l'accesso SSH non sia aperta. In caso contrario, l'istanza potrebbe ricevere continui tentativi di connessione SSH da parte di bot malintenzionati su Internet. Questi bot esplorano gli indirizzi IP pubblici. 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 host che bridge. Tuttavia, la modalità awsvpc di rete non prevede 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 utilizzi una sottorete privata e un gateway NAT, puoi 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 puoi utilizzare un gateway Network Address Translation (NAT) 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 container dall'accesso esterno diretto. Gli scenari applicabili includono sistemi di elaborazione dei pagamenti o

container 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 le tariffe orarie di utilizzo ed elaborazione dati del gateway NAT. Per motivi di ridondanza devi 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 specifica le sottoreti private per la configurazione di rete del servizio e non utilizzare l'opzione Assegna indirizzo IP pubblico. Ciascun processo Fargate è ospitato in una sottorete privata. Il suo traffico in uscita viene instradato attraverso qualsiasi gateway NAT associato a quella sottorete privata.

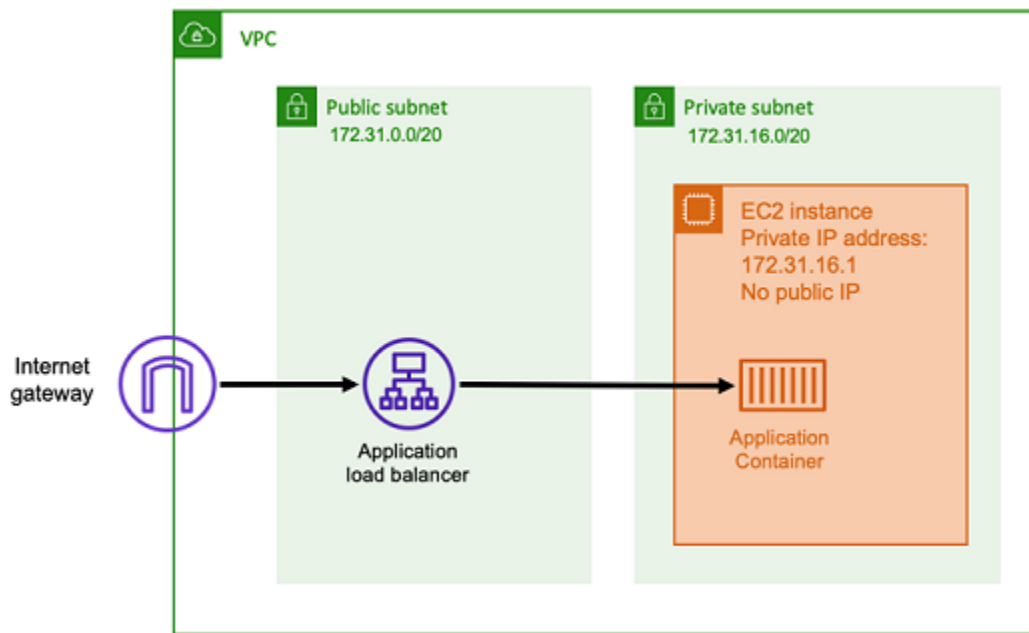
Best practice per ricevere connessioni in entrata ad Amazon ECS da Internet

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 container su host che si trovano in una sottorete pubblica con un indirizzo IP pubblico. Tuttavia, lo sconsigliamo 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 servizi AWS 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 una REST API HTTP, un Application Load Balancer è un bilanciatore 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 Balancer.



Con questa architettura crei 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.

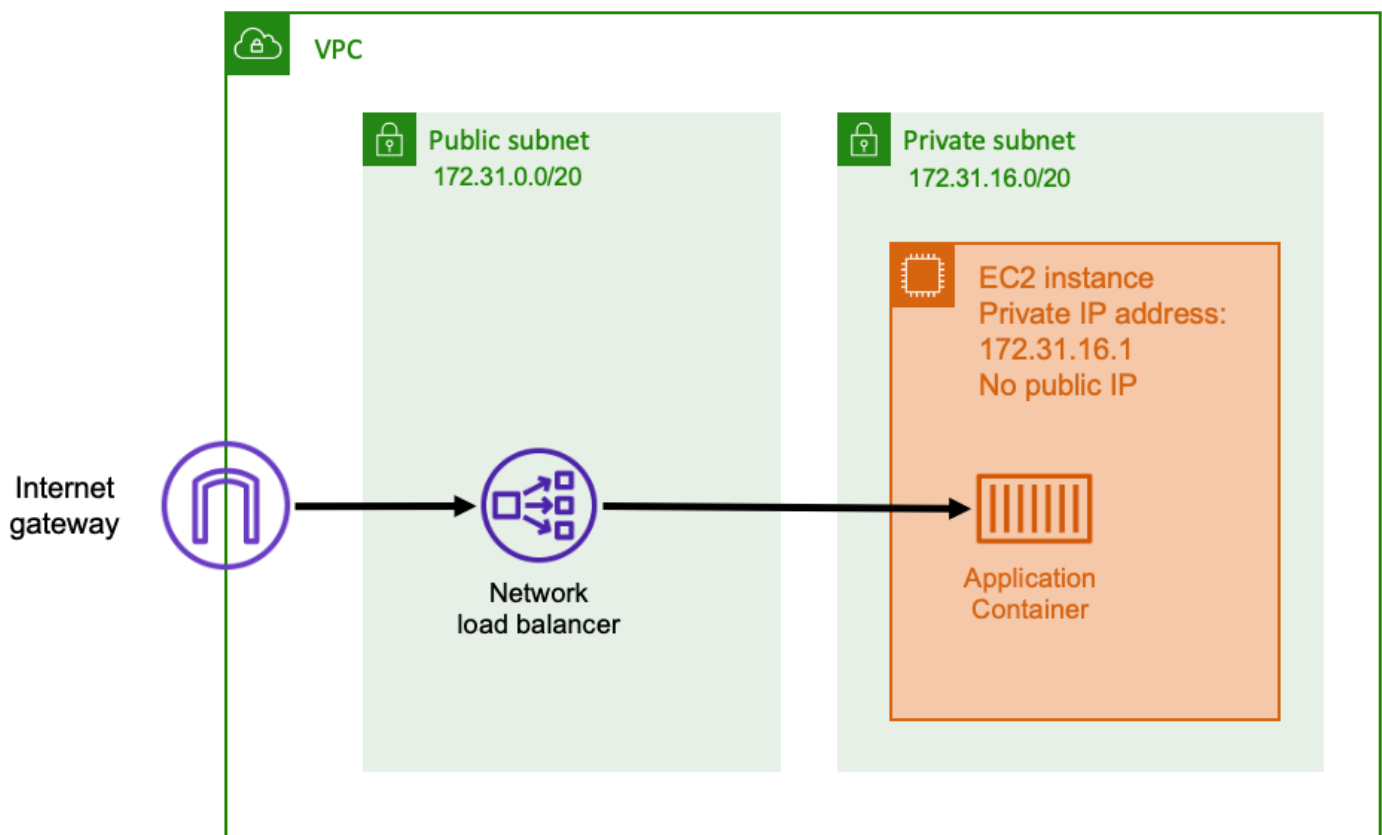
Un Application Load Balancer ha i seguenti vantaggi.

- **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 bilanciatore del carico 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 puoi utilizzare un singolo Application Load Balancer come input per molti servizi interni diversi o anche microservizi su percorsi diversi di una REST API.
- **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 HTTP de sync 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 per i Network Load Balancer.



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 tuo container e invia i pacchetti dal lato pubblico a quello privato.

Funzionalità del Network Load Balancer

Poiché il Network Load Balancer opera a un livello inferiore dello stack di rete, non ha lo stesso set di funzionalità dell'Application Load Balancer. Tuttavia, presenta le seguenti funzionalità 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

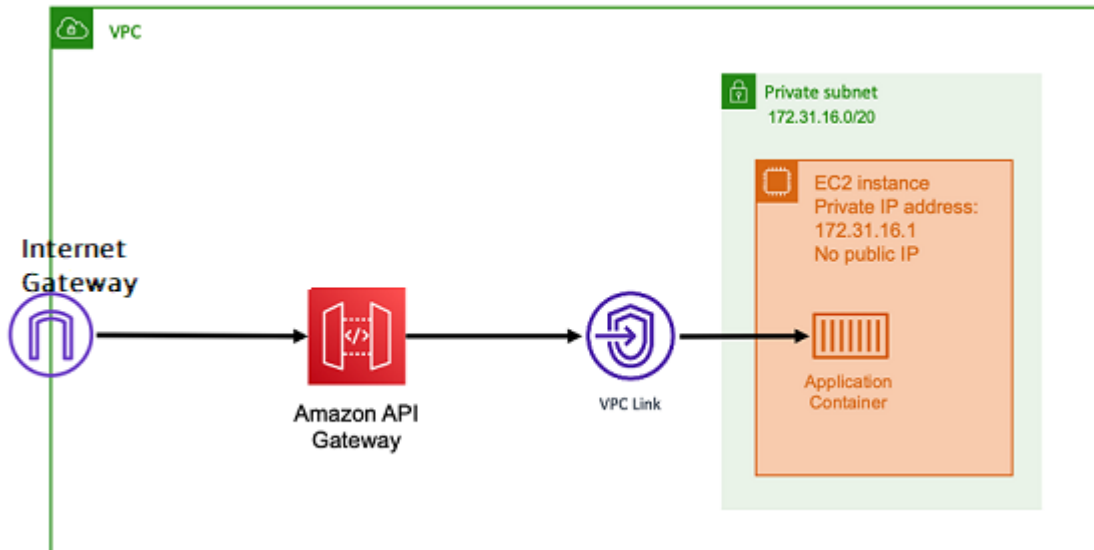
Poiché il Network Load Balancer non rispetta il protocollo applicativo ai livelli superiori del modello OSI, non può inviare messaggi di chiusura ai client in tali protocolli. A differenza dell'Application Load Balancer, tali connessioni devono essere chiuse dall'applicazione oppure è possibile configurare il Network Load Balancer per chiudere le connessioni di quarto livello quando un'attività viene interrotta o sostituita. Verifica l'impostazione di terminazione della connessione per i gruppi di destinazione 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 uno di questi è garantire che il ritardo di annullamento della registrazione della destinazione del Network Load Balancer sia più lungo del timeout della connessione client. Il client prima andava in timeout e si riconnetteva senza problemi tramite il Network Load Balancer all'attività successiva, mentre la vecchia operazione esauriva lentamente tutti i client. Per ulteriori informazioni sul ritardo di annullamento della registrazione della destinazione previsto per Network Load Balancer, consulta la [documentazione di Network Load Balancer](#).

HTTP API del Gateway Amazon API

Gateway Amazon API è adatto per applicazioni HTTP con picchi improvvisi di volumi di richieste o volumi di richieste bassi. Per ulteriori informazioni, consulta [Cos'è il Gateway Amazon API?](#) nella Guida per gli 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 addebiti. 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 bilanciatore del carico poco utilizzato. 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 bilanciatore 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 request/response modifica. Per ulteriori informazioni, consulta [Funzionalità di Gateway Amazon API](#).
- L'API Gateway può supportare endpoint API gateway 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

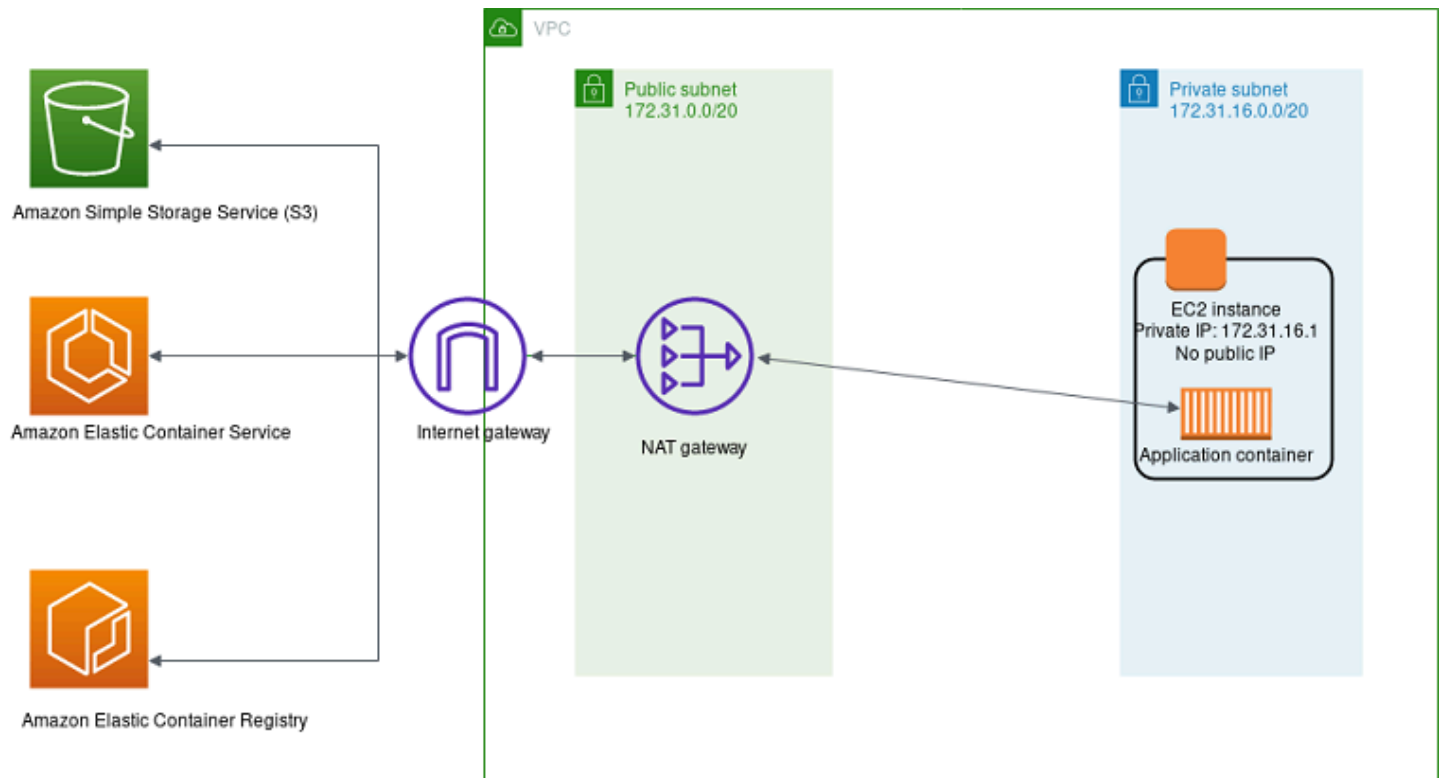
Oltre alle funzionalità precedenti, API Gateway supporta anche l'utilizzo di autorizzatori Lambda personalizzati che puoi utilizzare per proteggere l'API dall'utilizzo non autorizzato. Per ulteriori informazioni, consulta [Field Notes: Serverless Container based with APIs Amazon ECS e Amazon API Gateway](#).

Le migliori pratiche per connettere Amazon ECS ai AWS servizi dall'interno del tuo VPC

Affinché Amazon ECS funzioni correttamente, l'agente del container Amazon ECS in esecuzione su ogni host deve comunicare con il piano di controllo (control-plane) 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 indicati 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:
 - Download di file di grandi dimensioni da Amazon S3
 - Esecuzione di un volume elevato di query di database su DynamoDB
 - Estrazione di immagini da Amazon ECR

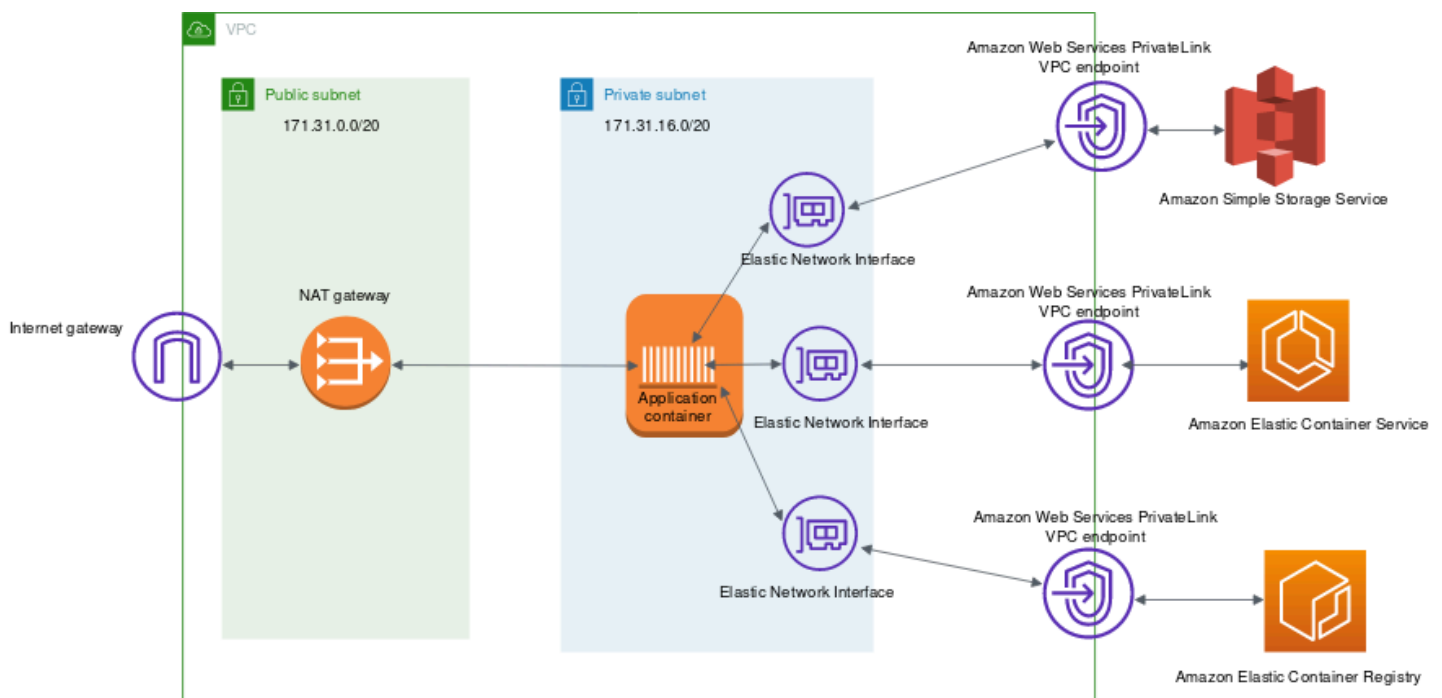
Un gateway NAT supporta 5 Gbps di larghezza di banda e aumenta verticalmente in modo automatico fino a 45 Gbps. Se si esegue il routing attraverso un singolo gateway NAT, le applicazioni che richiedono connessioni a larghezza di banda molto elevata potrebbero incontrare vincoli di rete. Come soluzione 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 VPCs e AWS servizi e le reti locali senza esporre il traffico alla rete Internet pubblica.

Un endpoint VPC consente connessioni private tra il VPC e i servizi AWS supportati 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. 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 servizio AWS , dovresti consultare la documentazione specifica per quel servizio e per la creazione di un endpoint VPC per quel traffico.

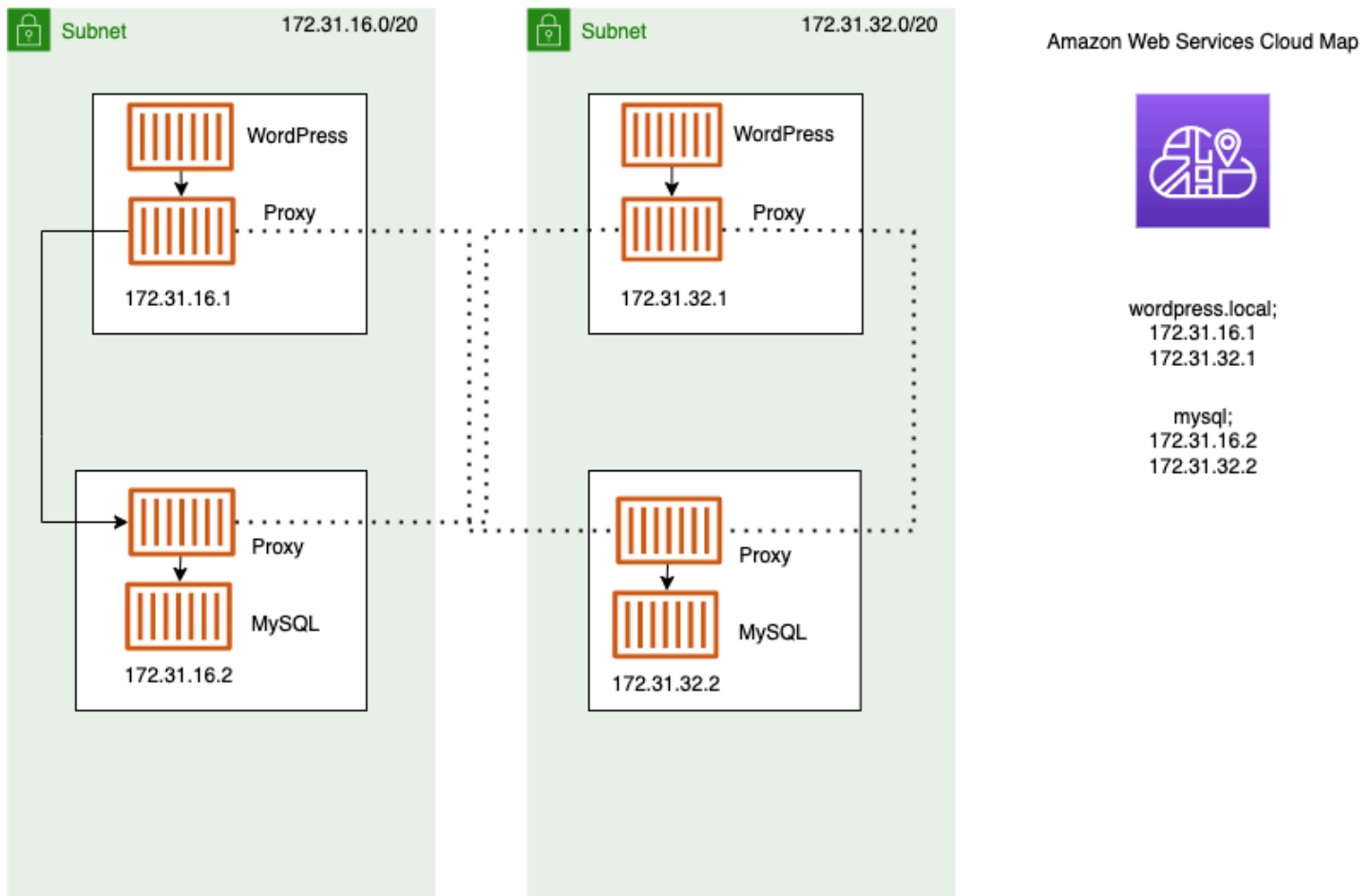
Best practice per connettere i servizi Amazon ECS in un VPC

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 il rilevamento 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 [Service Connect di Amazon ECS](#).

Quando usi Service Connect, Amazon ECS gestisce tutte le parti del rilevamento servizi: crea i nomi che possono essere rilevati, gestisce dinamicamente le voci per ogni attività all'inizio e all'interruzione, esegue un agente in ogni attività configurato per rilevare 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 modificare l'applicazione per utilizzare Service Connect.



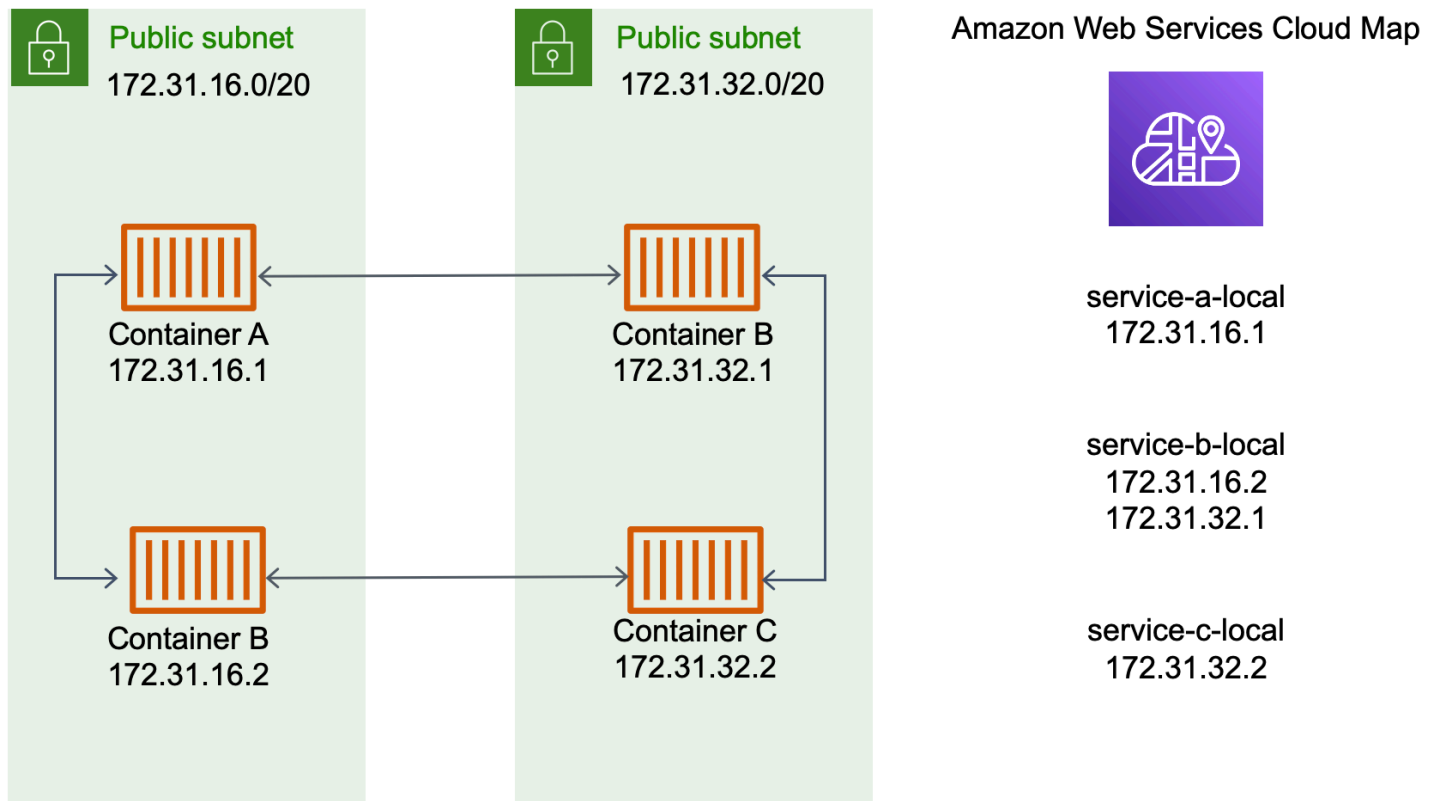
Le modifiche avvengono solo durante le implementazioni

Fornisci la configurazione completa all'interno di ogni definizione di servizio e attività. Amazon ECS gestisce le modifiche a questa configurazione in ogni implementazione del servizio, per garantire che tutte le attività di un'implementazione si comportino nello stesso modo. Ad esempio, un problema comune relativo al DNS come rilevamento servizi 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, l'implementazione del client aggiorna la configurazione sostituendo le attività del client. È possibile configurare l'interruttore automatico di implementazione e altre configurazioni di implementazione per influire sulle modifiche di Service Connect allo stesso modo di qualsiasi altra implementazione.

Utilizzo del rilevamento servizi

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 nel VPC Amazon possono utilizzare questo nome host DNS per inviare il traffico direttamente a un altro container utilizzando il suo indirizzo IP interno. Per ulteriori informazioni, consulta [Rilevamento servizi](#).



Nel diagramma precedente sono presenti tre servizi. `service-a-local` ha un container e comunica con `service-b-local`, che ha due container. `service-b-local` deve comunicare anche con `service-c-local`, che ha un container. 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 dal servizio downstream 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 container. Il traffico viaggia direttamente da un container all'altro.

Questo approccio è adatto quando si utilizza la modalità di rete `awsvpc`, in cui ogni attività ha il proprio indirizzo IP univoco. La maggior parte dei software supporta solo l'uso di record A del DNS, che si risolvono direttamente negli indirizzi IP. Quando si utilizza la modalità di rete `awsvpc`, l'indirizzo IP per ogni attività è un record A. Tuttavia, se si utilizza la modalità di rete `bridge`, è possibile che più container condividano lo stesso indirizzo IP. Inoltre, le mappature dinamiche delle

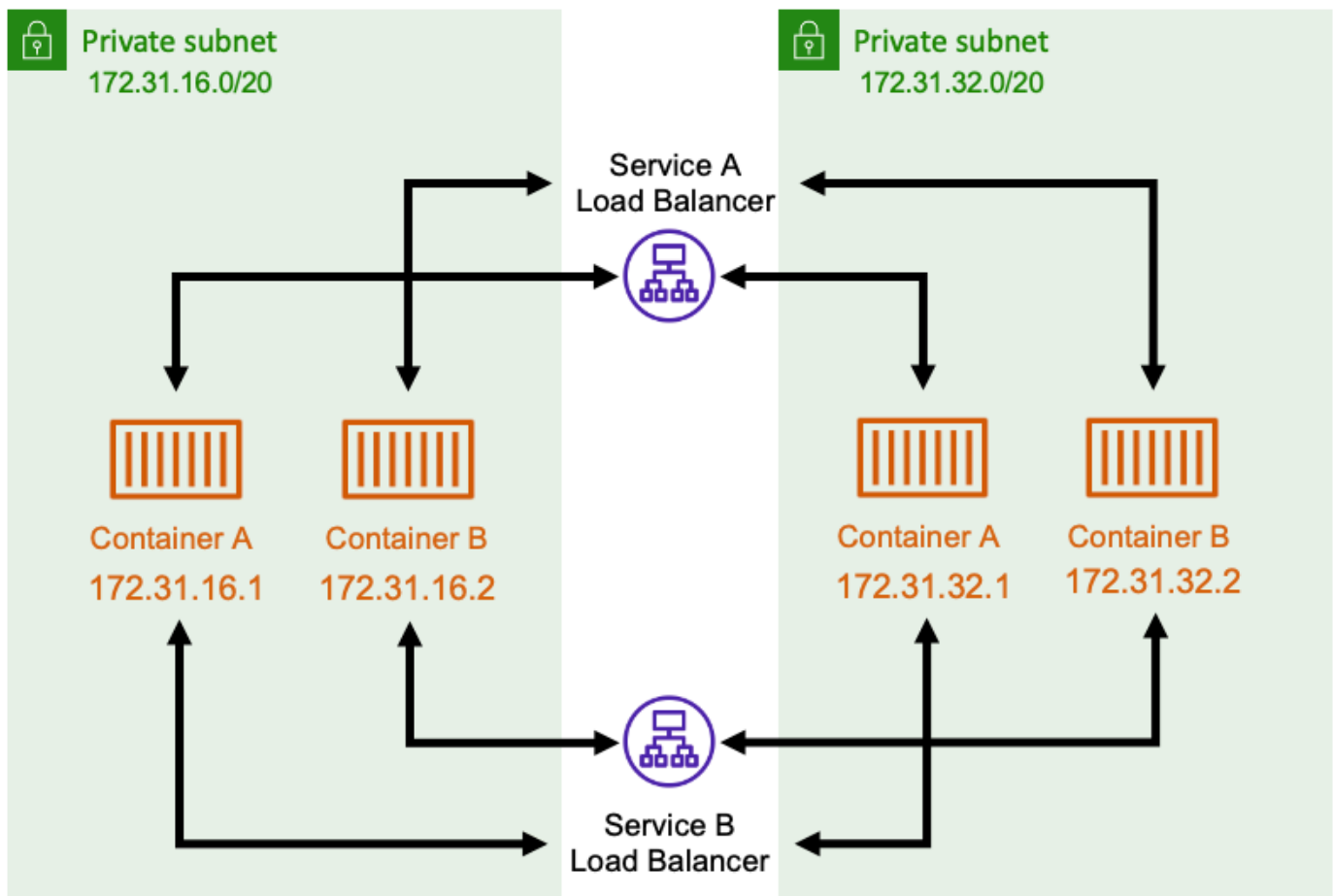
porte fanno sì che ai container vengano assegnati in modo casuale i numeri di porta su quel singolo indirizzo IP. A questo punto, un record A non è più sufficiente per il rilevamento servizi. Devi utilizzare un record SRV. Questo tipo di record può tenere traccia sia degli indirizzi IP che dei numeri di porta, ma richiede la configurazione appropriata delle applicazioni. Alcune applicazioni predefinite utilizzate potrebbero non supportare i record SRV.

Un altro vantaggio della modalità di rete `awsvpc` 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.

Il principale svantaggio della service-to-service comunicazione diretta che utilizza il 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 tale che punti a un altro container non più presente. L'applicazione deve gestire i nuovi tentativi e disporre di una logica per ignorare i backend non validi.

Utilizzo di un bilanciatore del carico interno

Un altro approccio alla service-to-service comunicazione consiste nell'utilizzare un sistema di bilanciamento del carico interno. Un bilanciatore del carico interno esiste interamente all'interno del tuo VPC ed è accessibile solo ai servizi all'interno del tuo VPC.



Il bilanciatore del carico mantiene un'elevata disponibilità distribuendo risorse ridondanti in ogni sottorete. Quando un container di serviceA deve comunicare con un container di serviceB, apre una connessione al bilanciatore del carico. Il bilanciatore del carico apre quindi una connessione a un container di service B. Il bilanciatore del carico funge da luogo centralizzato per la gestione di tutte le connessioni tra ciascun servizio.

Se un container di serviceB si ferma, il bilanciatore del carico può rimuoverlo dal pool. Il bilanciatore del carico esegue inoltre controlli di integrità su ogni destinazione a valle del relativo pool e può rimuovere automaticamente le destinazioni danneggiate dal pool fino a quando non tornano a funzionare di nuovo. Non è più necessario che le applicazioni siano consapevoli del numero di container a valle presenti. Si limitano ad aprire le connessioni al bilanciatore del carico.

Questo approccio è vantaggioso per tutte le modalità di rete. Il bilanciatore del carico può tenere traccia degli indirizzi IP delle attività quando si utilizza la modalità di rete `awsvpc`, nonché delle combinazioni più avanzate di indirizzi IP e porte quando si utilizza la modalità di rete `bridge`.

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 bilanciatore del carico deve disporre di risorse in ogni zona di disponibilità. Ciò comporta costi aggiuntivi a causa del sovraccarico dovuto al pagamento del bilanciatore del carico e alla quantità di traffico che attraversa il bilanciatore del carico.

Tuttavia, è possibile ridurre i costi generali facendo in modo che più servizi condividano un bilanciatore del carico. Ciò è particolarmente adatto per i servizi REST che utilizzano un sistema Application Load Balancer. È possibile creare regole di instradamento basate sul percorso che instradano il traffico verso servizi diversi. Ad esempio, `/api/user/*` potrebbe indirizzare verso un container che fa parte del servizio `user`, mentre `/api/order/*` potrebbe indirizzare verso il servizio `order` associato. Con questo approccio, paghi solo per un Application Load Balancer e disponi di un URL adatto 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'organizzazione condivisa 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 facilitare l'instradamento del 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?](#) in Guida ai gateway di transito di Amazon VPC.
- **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 servizi e le funzionalità riportati di seguito possono aiutarti a ottenere informazioni dettagliate sulle configurazioni di rete e dei servizi. Puoi utilizzare queste informazioni per risolvere i problemi relativi alla rete e ottimizzare al meglio i servizi.

CloudWatch Container Insights

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. I parametri includono l'utilizzo di risorse come CPU, memoria, dischi e rete. Sono CloudWatch disponibili in dashboard 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 log 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 eseguire il fine-tuning di alcune impostazioni per migliorare la tua rete.

ulimit nofile

Se prevedi che l'applicazione abbia un traffico elevato e gestisca molte connessioni simultanee, devi 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 tua quota di descrittori di file è troppo bassa, limiterà i socket di rete. Ciò si traduce in connessioni non riuscite o errori. Puoi aggiornare la quota specifica del container per il numero di file nella definizione dell'attività 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`. Fai 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.

Ottimizza i tempi 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 container e le istanze binpack

Se lo utilizzi EC2, puoi configurare il comportamento pull dell'agente container Amazon ECS su `ECS_IMAGE_PULL_BEHAVIOR:prefer-cached`. L'immagine viene estratta da remoto se non esiste un'immagine memorizzata nella cache. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. La pulizia automatizzata dell'immagine è disattivata per il container, per avere la certezza che l'immagine memorizzata nella cache non venga rimossa. Ciò riduce il tempo di recupero delle immagini per gli avvii successivi. L'effetto della memorizzazione nella cache è ancora maggiore quando nelle istanze di container 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 posizionamento `binpack` puoi anche prendere in considerazione l'utilizzo del trunking di Interfaccia di rete elastica (ENI) per collocare più attività con la modalità di rete `awsvpc` su ogni istanza di container. Il trunking ENI aumenta il numero di attività che puoi eseguire in modalità `awsvpc`. 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 `awsvpc` rete è ideale, questa modalità di rete può aumentare intrinsecamente la latenza di avvio delle attività, perché per ogni attività in modalità `awsvpc`, 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 rete `awsvpc` 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 tua priorità è la velocità di implementazione, puoi prendere in considerazione l'utilizzo della modalità `bridge` 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 container, l'esecuzione di script di avvio e altre configurazioni durante l'avvio dell'applicazione possono richiedere una quantità di tempo incredibile. Puoi utilizzare l'endpoint Metadati delle attività per pubblicare parametri per tenere traccia del tempo di avvio dell'applicazione da `StartedAt` nella risposta dei metadati del container a `StartedAt` dell'attività o del servizio. Grazie a questi dati puoi comprendere in che modo l'applicazione contribuisce al tempo totale di avvio e individuare aree in cui puoi ridurre il sovraccarico non necessario specifico dell'applicazione e ottimizzare le immagini del container. Per ulteriori informazioni, consulta [Best practice di dimensionamento automatico e gestione della capacità di Amazon ECS](#).

- Scegli un tipo di istanza ottimale (per) EC2

La scelta del tipo di istanza corretto si basa sulla prenotazione di risorse (ad esempio, CPU, memoria) che configuri per la tua attività. Pertanto, quando ridimensioni l'istanza, puoi 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.

Utilizzo di Amazon ECS su larga scala

Quando inizi a utilizzare Amazon ECS su larga scala, considera in che modo le quote di servizio e le limitazioni delle API per Amazon ECS e l'integrazione con Amazon ECS Servizi AWS potrebbero influire su di te.

Argomenti

- [Quote di servizio di Amazon ECS e limiti di limitazione delle API](#)
- [Gestisci i problemi di limitazione di Amazon ECS](#)

Quote di servizio di Amazon ECS e limiti di limitazione delle API

Amazon ECS si integra 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, il rilevamento servizi, le reti di processi e il dimensionamento automatico 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 i costi.

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 [Quote di servizio Amazon ECS](#) e [Richiesta di limitazione per l'API Amazon ECS](#).

Quando ridimensioni i carichi di lavoro su Amazon ECS, tieni presente la quota di servizio seguente. Per istruzioni su come richiedere un aumento della quota di servizio, consulta [Gestire le quote di Amazon ECS e AWS Fargate di servizio nel](#) Console di gestione AWS.

- AWS Fargate dispone di quote che limitano il numero di attività in esecuzione simultanee in ciascuna di esse. Regione AWS Esistono quote per le attività on demand e Fargate Spot su Amazon ECS. Ogni quota di servizio comprende anche tutti i pod Amazon EKS che esegui su Fargate. Per ulteriori informazioni, consulta [Quote di servizio AWS Fargate](#) nella Guida per l'utente di Amazon Elastic Container Service per AWS 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 usi il dimensionamento automatico del cluster Amazon ECS con un provider di capacità del gruppo Auto Scaling, quando scali i tuoi servizi considera la quota `Tasks in the PROVISIONING state per cluster`. Questa quota è il numero massimo di attività nello stato `PROVISIONING` per ogni cluster per le quali i provider di capacità possono aumentare

la capacità. Quando avvii un gran numero di attività contemporaneamente, puoi facilmente raggiungere la quota. Un esempio è l'implementazione simultanea di decine di servizi, ognuno con centinaia di attività. Quando ciò accade, il provider di capacità deve avviare nuove istanze di container per collocare le attività quando il cluster non dispone di capacità sufficiente. 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 di una capacità insufficiente del cluster. Il pianificatore del servizio Amazon ECS implementa una strategia di back-off e limitazione esponenziale per riprovare il posizionamento delle attività man mano che vengono avviate nuove istanze di container. Di conseguenza, potrebbero verificarsi tempi di implementazione o di aumento orizzontale più lenti. Per evitare questa situazione puoi pianificare le implementazioni dei servizi in uno dei seguenti modi. Implementa un gran numero di attività che non richiedono un aumento della capacità del cluster oppure mantieni la capacità del cluster di riserva per l'avvio 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 che sono integrati con Amazon ECS. La sezione di seguito illustra nel dettaglio i limiti di velocità chiave per ciascun servizio e fornisce consigli per affrontare potenziali problemi di limitazione.

bilanciatore del carico elastico

Puoi configurare i servizi Amazon ECS in modo da usare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività. Per ulteriori informazioni su come scegliere un bilanciatore del carico, consulta [Utilizzo del bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

Quote di servizio Elastic Load Balancing

Quando scali i carichi di lavoro, tieni conto delle seguenti quote del servizio Elastic Load Balancing. La maggior parte delle quote del servizio Elastic Load Balancing sono regolabili ed è possibile richiedere un aumento nella console Service Quotas.

Application Load Balancer

Quando usi un Application Load Balancer, a seconda del caso d'uso, potresti dover richiedere un aumento della quota per:

- La quota **Targets** per **Application Load Balancer**, ossia il numero di destinazioni alla base dell'Application Load Balancer.

- La quota **Targets** per **Target Group** per **Region**, ossia il numero di destinazioni alla base dei gruppi di destinazioni.

Per ulteriori informazioni, consulta la [Quote per Application Load Balancer](#).

Network Load Balancer

Ci sono limitazioni più rigorose sul numero di destinazioni che è possibile registrare con un Network Load Balancer. Quando si usa un Network Load Balancer, spesso si vuole abilitare il supporto tra zone, che comporta limitazioni di dimensionamento aggiuntive sui **Targets** per **Availability Zone** Per **Network Load Balancer**, ovvero sul numero massimo di destinazioni per zona di disponibilità per ogni Network Load Balancer. Per ulteriori informazioni, consulta [Quote per i Network Load Balancer](#).

Limitazione dell'API Elastic Load Balancing

Quando configuri un servizio Amazon ECS per usare un bilanciatore del carico, i controlli dell'integrità del gruppo di destinazione devono essere superati prima che il servizio venga considerato integro. Per eseguire questi controlli dell'integrità, Amazon ECS invoca le operazioni dell'API Elastic Load Balancing per tuo conto. Se hai un gran numero di servizi configurati con bilanciatori del carico nel tuo account, potresti riscontrare implementazioni dei servizi più lente a causa di una potenziale limitazione specifica per le operazioni **RegisterTarget**, **DeregisterTarget** e **DescribeTargetHealth** dell'API Elastic Load Balancing. Quando si verifica una limitazione, compaiono 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 [Handling throttling issues](#).

Interfacce di rete elastiche

Quando le tue attività utilizzano la modalità di rete **awsvpc**, Amazon ECS fornisce un'interfaccia di rete elastica (ENI) unica per ogni attività. Quando i tuoi servizi Amazon ECS usano un bilanciatore del carico Elastic Load Balancing, queste interfacce di rete vengono registrate anche come destinazioni per il gruppo di destinazione appropriato definito nel servizio.

Quote di servizio dell'interfaccia di rete elastica

Quando esegui attività che usano la modalità di rete `awsvpc`, a ciascuna attività viene collegata un'interfaccia elastica di rete unica. Se tali attività devono essere eseguite tramite Internet, assegna un indirizzo IP pubblico all'interfaccia di rete elastica per tali attività. Quando scali 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 quota `Elastic IP addresses per Region`, ossia il numero massimo di indirizzi IP elastici in una Regione AWS.

Entrambe le quote di servizio sono regolabili e puoi richiedere un aumento dalla tua console Service Quotas. Per ulteriori informazioni, consulta [Quote di servizio Amazon VPC](#).

Per i carichi di lavoro Amazon ECS ospitati su EC2 istanze Amazon, quando esegui attività che utilizzano la modalità di rete `awsvpc`, 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 possono essere eseguite su un'istanza. La quota non può essere modificata 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'istanza `c5.large` può avere fino a tre interfacce di rete. L'interfaccia di rete principale per l'istanza viene calcolata come unica. Quindi, puoi collegare due interfacce di rete aggiuntive all'istanza. Poiché ogni attività che usa la modalità di rete `awsvpc` richiede un'ENI, di solito si possono eseguire solo due attività su questo tipo di istanza. Questo può portare a un sottoutilizzo della capacità del cluster. Se abiliti il trunking dell'interfaccia di rete elastica, puoi aumentare la densità dell'interfaccia di rete per collocare un numero maggiore di attività su ciascuna istanza. Con il trunking attivo, un'istanza `c5.large` può avere fino a 12 interfacce di rete. L'istanza ha un'interfaccia di rete primaria e Amazon ECS crea e collega un'interfaccia di rete “trunk” all'istanza. Di conseguenza, con questa configurazione puoi eseguire 10 attività sull'istanza invece delle due attività predefinite. Per ulteriori informazioni, consulta [Trunking delle interfacce di rete elastiche](#).

Limitazione della larghezza di banda della rete dell'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 [Request throttling for the Amazon EC2 API](#).

- `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. Talvolta, questi nuovi tentativi automatici possono causare un ritardo nell'avvio delle attività, con conseguente rallentamento della velocità di implementazione. Quando si verifica la limitazione delle API, compare 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 sulla risoluzione di tali errori di limitazione della larghezza di banda della rete, consulta [Handling throttling issues](#).

AWS Cloud Map

Amazon ECS Service Discovery utilizza AWS Cloud Map APIs per gestire i namespace per i tuoi servizi Amazon ECS. Se i tuoi servizi prevedono un gran numero di attività, valuta i seguenti suggerimenti. Per maggiori informazioni, consulta le [Considerazioni sul rilevamento servizi di Amazon ECS](#).

AWS Cloud Map quote di servizio

Quando i servizi Amazon ECS sono configurati per utilizzare il service discovery, 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 la quantità di attività che è possibile eseguire fino a un massimo di 1.000 attività per servizio. Non puoi cambiare le quote AWS Cloud Map . Per ulteriori informazioni, consulta [AWS Cloud Map service quotas](#).

AWS Cloud Map Limitazione delle API

Amazon ECS chiama `listInstances`, `getInstanceHealthStatus` e `registerInstance`, e per tuo `deregisterInstance` AWS Cloud Map APIs conto. Aiutano nel rilevamento servizi ed eseguono controlli dell'integrità quando si avvia un'attività. Quando più servizi che utilizzano il service discovery con un gran numero di attività vengono distribuiti contemporaneamente, 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 [Handling throttling issues](#).

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 in questione e sui relativi limiti di limitazione, consulta la sezione [Richiesta di limitazione per l'API Amazon ECS](#).

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 di backoff esponenziale e jitter con logica di ripetizione per le chiamate API. Per maggiori informazioni, consulta [Timeout, nuovi tentativi e backoff con jitter](#).

Se utilizzi un AWS SDK, la logica di ripetizione automatica dei tentativi è integrata e configurabile.

Limitazione asincrona

La limitazione asincrona si verifica a causa di flussi di lavoro asincroni in cui Amazon ECS o 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'API `CreateNetworkInterface` viene invocata per attività che utilizzano la modalità di rete `awsvpc` e l'API `DescribeTargetHealth` viene invocata quando si eseguono controlli dell'integrità per attività registrate su un bilanciatore del carico.

Quando i carichi di lavoro raggiungono una dimensione 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 `awsvpc` 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.

Monitoraggio della 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. Ciò identifica i dettagli degli 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 puoi utilizzare query SQL come la seguente per analizzare gli errori ThrottlingException.

Sostituiscili *user-input* con i tuoi valori.

```
select eventname, errorcode,eventsources,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, eventsources, 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. Questi includono eventi di limitazione delle API come ECS_OPERATION_THROTTLED e SERVICE_DISCOVERY_OPERATION_THROTTLED. Per ulteriori informazioni, consulta [Eventi di operazioni di servizi 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 come RunTask sono asincrone e le operazioni di nuovo tentativo non vengono eseguite automaticamente. In questi casi, è possibile utilizzare servizi come AWS Step Functions with EventBridge integration per riprovare operazioni limitate o

non riuscite. Per ulteriori informazioni, consulta [Gestione di un'attività del container \(Amazon ECS, Amazon SNS\)](#).

Da utilizzare per monitorare la limitazione CloudWatch

CloudWatch offre il monitoraggio dell'utilizzo delle API **Usage** nello spazio dei nomi in By Resource. AWS Queste metriche vengono registrate con il tipo di API e il nome della metrica. CallCount Puoi creare avvisi che si attivano ogni volta che queste metriche raggiungono una determinata soglia. Per ulteriori informazioni, consulta [Visualizzazione delle quote di servizio e impostazione degli avvisi](#).

CloudWatch offre anche il rilevamento delle anomalie. Questa funzionalità utilizza il machine learning per analizzare e stabilire linee di base a seconda del comportamento particolare della metrica su cui l'hai abilitato. In caso di attività insolite dell'API, puoi utilizzare questa funzione insieme agli CloudWatch allarmi. 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.

Best practice di dimensionamento automatico e gestione della capacità di Amazon ECS

Su Amazon ECS puoi eseguire carichi di lavoro di applicazioni containerizzate di tutte le dimensioni. Ciò include ambienti di test minimi e ambienti di produzione di grandi dimensioni che operano su scala globale.

Con Amazon ECS, come tutti i AWS servizi, paghi solo per ciò che usi. Quando progetti l'applicazione in modo adeguato puoi risparmiare sui costi consumando solo le risorse di cui hai bisogno, quando ne hai bisogno.

I consigli riportati di seguito mostrano come eseguire i carichi di lavoro Amazon ECS per soddisfare i tuoi obiettivi di livello di servizio operando a costi contenuti.

Argomenti

- [Determinazione delle dimensioni delle attività per Amazon ECS](#)
- [Ottimizzazione del dimensionamento automatico del servizio Amazon ECS](#)

- [Capacità e disponibilità di Amazon ECS](#)
- [Capacità dei cluster Amazon ECS](#)
- [Scelta delle dimensioni delle attività Fargate per Amazon ECS](#)
- [Accelerazione del provisioning della capacità dei cluster Amazon ECS con i fornitori di capacità su Amazon EC2](#)

Determinazione delle dimensioni delle attività per Amazon ECS

Una delle scelte più importanti quando implementi container su Amazon ECS è la dimensione dei container e delle attività. Le dimensioni dei container e delle attività sono essenziali per la scalabilità e la pianificazione della capacità.

Amazon ECS utilizza due parametri di risorse per la capacità: CPU e memoria. Amazon ECS misura la CPU in unità di 1/1024 di una vCPU completa (dove 1024 unità equivalgono a 1 vCPU intera). Amazon ECS misura la memoria in megabyte.

Nella definizione dell'attività, puoi dichiarare prenotazioni e limiti di risorse.

Quando dichiari una prenotazione, dichiari la quantità minima di risorse richiesta da un'attività. L'attività riceve almeno la quantità di risorse da te richiesta. L'applicazione potrebbe essere in grado di utilizzare più CPU o memoria rispetto alla prenotazione dichiarata. Tuttavia, ciò è soggetto ai limiti che hai dichiarato.

L'utilizzo di una quantità superiore alla quantità della prenotazione è noto come bursting. Il bursting significa che l'applicazione utilizza più risorse di quelle prenotate, ma rimane entro i limiti dichiarati. Amazon ECS garantisce le prenotazioni. Ad esempio, se utilizzi EC2 istanze Amazon per fornire capacità, Amazon ECS non inserisce un'attività su un'istanza in cui non è in grado di soddisfare la prenotazione.

Un limite è la quantità massima di unità CPU o memoria che il container o l'attività può utilizzare. Se il container tenta di utilizzare più CPU rispetto a questo limite, Amazon ECS lo limita. Se il container tenta di utilizzare più memoria oltre questo limite, Amazon ECS lo arresta.

La scelta di questi valori può essere complessa. I valori che funzionano meglio per l'applicazione dipendono molto dai requisiti di risorse dell'applicazione.

Il test di carico dell'applicazione è la chiave per una corretta pianificazione del fabbisogno di risorse. Il test di carico consente di comprendere meglio i requisiti dell'applicazione.

Applicazioni stateless

Per le applicazioni stateless con scalabilità orizzontale, ad esempio un'applicazione con bilanciatore del carico, consigliamo innanzitutto di determinare la quantità di memoria utilizzata dall'applicazione per soddisfare le richieste.

A tale scopo, puoi utilizzare strumenti tradizionali come `ps` o `top`. Puoi anche utilizzare soluzioni di monitoraggio come CloudWatch Container Insights.

Quando stabilisci una prenotazione della CPU, considera come vuoi scalare l'applicazione per soddisfare i requisiti aziendali.

Puoi utilizzare prenotazioni di CPU minori, ad esempio 256 unità CPU (o 1/4 vCPU), per aumentare orizzontalmente in modo da ridurre al minimo i costi. Tuttavia, potrebbero non scalare abbastanza velocemente per soddisfare i picchi significativi della domanda.

Puoi utilizzare prenotazioni di CPU maggiori per ridurre e aumentare orizzontalmente più in fretta. Ciò ti aiuta a soddisfare più rapidamente i picchi di domanda. Tuttavia, le prenotazioni di CPU maggiori costano di più.

Altre applicazioni

Per le applicazioni che non sono scalabili orizzontalmente, come worker singleton o server di database, le considerazioni più importanti sono la capacità e i costi disponibili.

Scegli la quantità di memoria e di CPU in base ai risultati dei test di carico necessari per gestire il traffico e raggiungere il tuo obiettivo di livello di servizio. Amazon ECS garantisce che l'applicazione sia collocata su un host con una capacità adeguata.

Ottimizzazione del dimensionamento automatico 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.

Il dimensionamento automatico del servizio Amazon ECS funziona 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.

Sei tu a fornire le soglie per la scalabilità. Puoi impostare un parametro di destinazione, denominato dimensionamento con monitoraggio della destinazione. Puoi anche specificare delle soglie, operazione denominata scalabilità a fasi.

Dopo aver configurato Application Auto Scaling, questo 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 verticalmente.

Per utilizzare il service di dimensionamento automatico in modo efficace, devi scegliere un parametro di dimensionamento adeguato. Nelle sezioni seguenti viene descritto come scegliere un parametro.

Caratterizzazione dell'applicazione

Per scalare correttamente un'applicazione devi conoscere le condizioni in cui è necessario scalare l'applicazione e quando è necessario scalarla orizzontalmente.

In sostanza, devi aumentare orizzontalmente l'applicazione se prevedi che la domanda superi la capacità. Al contrario, puoi ridurre orizzontalmente l'applicazione per ridurre i costi quando le risorse superano la domanda.

Identificazione di un parametro di utilizzo

Per una scalabilità efficace devi identificare un parametro che indichi l'utilizzo o la saturazione. Tale parametro deve presentare le seguenti proprietà per essere utile per la scalabilità.

- Il parametro deve essere correlato alla domanda. Quando si mantengono stabili le risorse ma la domanda cambia, deve cambiare anche il valore del parametro. Il parametro deve aumentare o diminuire quando la domanda aumenta o diminuisce.
- Il valore del parametro deve essere ridotto orizzontalmente in proporzione alla capacità. Quando la domanda rimane costante, l'aggiunta di più risorse deve comportare una modifica proporzionale del valore del parametro. Pertanto, il raddoppio del numero di attività dovrebbe far diminuire il parametro del 50%.

Il modo migliore per identificare un parametro di utilizzo è eseguire test di carico in un ambiente di preproduzione come un ambiente di staging. Sono ampiamente disponibili soluzioni commerciali e open source per i test di carico. Queste soluzioni di solito possono generare carichi sintetici o simulare il traffico utente reale.

Per avviare il processo di test di carico devi prima creare delle dashboard per i parametri di utilizzo dell'applicazione. Queste metriche includono l'utilizzo della CPU, l'utilizzo della memoria, I/O le

operazioni, la profondità della I/O coda e il throughput di rete. Puoi raccogliere queste metriche con un servizio come Container Insights. CloudWatch Puoi inoltre rilevarli utilizzando Amazon Managed Service per Prometheus insieme a Grafana gestito da Amazon. Durante questo processo, assicurati di rilevare e monitorare i parametri relative ai tempi di risposta della tua applicazione o ai tassi di completamento del lavoro.

Quando esegui il test di caricamento, inizia con un tasso ridotto di inserimento di richieste o processi. Mantieni questa frequenza costante per diversi minuti per consentire all'applicazione di prepararsi. Quindi, aumenta lentamente la velocità e mantienila costante per alcuni minuti. Ripeti questo ciclo, aumentando la frequenza ogni volta fino a quando i tempi di risposta o completamento dell'applicazione non sono troppo lenti per soddisfare gli obiettivi a livello di servizio (). SLOs

Durante il test di caricamento esamina ciascun parametro di utilizzo. I parametri che aumentano insieme al carico sono quelli più adatti come migliori parametri di utilizzo.

Successivamente, identifica la risorsa che raggiunge la saturazione. Allo stesso tempo, esamina anche i parametri di utilizzo per vedere quale si appiattisce per prima a un livello elevato. Oppure, esamina quale raggiunge per primo il picco massimo e poi blocca l'applicazione. Ad esempio, se l'utilizzo della CPU aumenta dallo 0% al 70-80% man mano che aggiungi carico e poi rimane a quel livello dopo aver aggiunto ancora più carico, allora si può tranquillamente affermare 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 utilizzare più risorse. Pertanto, scegli il parametro che rappresenta la risorsa che si esaurisce per prima.

Infine, riprova a testare il carico dopo aver raddoppiato il numero di attività o di EC2 istanze Amazon. Supponiamo che il parametro chiave aumenti o diminuisca della metà rispetto a prima. In tal caso, il parametro è proporzionale alla capacità. Si tratta di un buon parametro di utilizzo per il dimensionamento automatico.

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 aggiungi altro carico, l'utilizzo della CPU non aumenta più. Tuttavia, fa sì che l'applicazione risponda più lentamente. Poi esegui nuovamente il test di carico, raddoppiando il numero di attività ma mantenendo la frequenza al valore di picco precedente. Se noti che l'utilizzo medio della CPU scende a circa il 40%, l'utilizzo medio della CPU è un buon candidato come parametro 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 un buon

parametro di scalabilità. In tal caso, sono necessarie ulteriori ricerche per trovare un parametro adeguato.

Modelli di applicazione e proprietà di dimensionamento comuni

Puoi eseguire software di ogni tipo su AWS. Molti carichi di lavoro sono creati internamente, mentre altri si basano su software open source comuni. Indipendentemente dalla loro origine, abbiamo osservato alcuni modelli di progettazione comuni per i servizi. Il modo in cui esegui una scalabilità efficace dipende in gran parte dal modello.

Il server efficiente collegato alla CPU

Il server efficiente 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 del throughput, è una frazione delle risorse disponibili.

Per questo tipo di applicazione puoi utilizzare il dimensionamento automatico basato sulla CPU. L'applicazione gode della massima flessibilità in termini di dimensionamento. Puoi scalarlo verticalmente fornendo EC2 istanze Amazon più grandi o Fargate v. CPUs. Puoi anche scalarla orizzontalmente aggiungendo più repliche. L'aggiunta di altre 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

Il server efficiente legato alla memoria

Il server efficiente legato alla memoria alloca una quantità significativa di memoria per richiesta. In caso di massima concorrenza, ma non necessariamente throughput, la memoria si esaurisce prima che si esauriscano le risorse della CPU. La memoria associata a una richiesta viene liberata al termine della richiesta. Possono essere accettate richieste aggiuntive purché sia disponibile memoria.

Per questo tipo di applicazione puoi utilizzare il dimensionamento automatico basato sulla memoria. L'applicazione gode della massima flessibilità in termini di dimensionamento. Puoi scalarlo sia

verticalmente fornendogli risorse di memoria Amazon EC2 o Fargate più grandi. Puoi anche scalarla orizzontalmente aggiungendo più repliche. L'aggiunta di altre 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 consigliamo di non utilizzare il dimensionamento basato sulla memoria.

Il server basato sul lavoro

Il server basato sul lavoro elabora una richiesta per ogni singolo thread di lavoro una dopo l'altra. I thread di lavoro possono essere thread leggeri, come i 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, l'applicazione inserisce richieste aggiuntive in una coda di backlog. Se la coda di backlog si esaurisce, l'applicazione rifiuta immediatamente le richieste aggiuntive in entrata. Le applicazioni comuni che si adattano a questo modello includono il server web Apache e Unicorn.

La concorrenza delle richieste è in genere il parametro migliore per scalare questa applicazione. Poiché esiste un limite di concorrenza per ogni replica, è importante aumentare orizzontalmente prima che venga raggiunto il limite medio.

Il modo migliore per ottenere i parametri relativi alla concorrenza delle richieste è chiedere all'applicazione di segnalarli. CloudWatch Ogni replica dell'applicazione può pubblicare il numero di richieste simultanee come parametro personalizzato ad alta frequenza. Consigliamo di impostare la frequenza su almeno una volta al minuto. Dopo aver raccolto diversi report, puoi utilizzare la concorrenza media come parametro di dimensionamento. Tale parametro si calcola 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 il parametro `ActiveConnectionCount` per il bilanciatore del carico come fattore nel parametro di dimensionamento. Per ottenere un valore medio devi dividere il parametro `ActiveConnectionCount` per il numero di repliche. Devi utilizzare il valore medio per il dimensionamento, anziché il valore di conteggio non elaborato.

Affinché questa progettazione funzioni al meglio, la deviazione standard della latenza di risposta deve essere minima a basse frequenze di richiesta. È bene che, durante i periodi di scarsa domanda, la maggior parte delle richieste riceva risposta in breve tempo e che non vi siano molte richieste che impiegano 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. Consigliamo 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 molto da uno o più servizi a valle. Le applicazioni container fanno spesso un uso intensivo di servizi a valle 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. Ciò accade perché tali 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 sul lavoro, a seconda di come è progettata l'applicazione. Se la concorrenza dell'applicazione è limitata solo dalla memoria, bisognerebbe utilizzare l'utilizzo medio della memoria come parametro di dimensionamento. Se la concorrenza dell'applicazione si basa su un limite di lavoratori, bisognerebbe utilizzare la concorrenza media come parametro di dimensionamento.

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 adatto come parametro di dimensionamento. Tuttavia, molte applicazioni Java non sono legate alla CPU, il che le rende difficili da scalare.

Per prestazioni ottimali, consigliamo di allocare quanta più memoria possibile all'heap Java Virtual Machine (JVM). Le versioni recenti di JVM, tra cui l'aggiornamento 191 di Java 8 o le versioni successive, impostano automaticamente la dimensione dell'heap al massimo possibile per adattarla al container. 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, sconsigliamo di 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 vanno incontro all'esaurimento dell'heap prima di esaurire la CPU. Se l'applicazione è soggetta all'esaurimento dell'heap in caso di elevata simultaneità,

il parametro di dimensionamento migliore è rappresentato dalle connessioni medie. Se la tua applicazione è soggetta all'esaurimento dell'heap a un throughput elevato, il parametro di dimensionamento migliore è la frequenza media delle richieste.

Server che utilizzano altri runtime raccolti tramite rimozione di oggetti inutili

Molte applicazioni server si basano su runtime che effettuano la rimozione di oggetti inutili, ad esempio .NET e Ruby. Queste applicazioni server potrebbero rientrare in uno dei modelli descritti in precedenza. Tuttavia, come nel caso di Java, sconsigliamo di scalare tali applicazioni in base alla memoria, poiché il loro utilizzo medio della memoria osservato spesso non è correlato al throughput o alla concorrenza.

Per tali applicazioni consigliamo di scalare 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.

Elaboratori di processi

Molti carichi di lavoro prevedono l'elaborazione asincrona dei processi. Questi includono applicazioni che non ricevono richieste in tempo reale, ma si iscrivono a una coda di lavoro per ricevere processi. Per questi tipi di applicazioni, il parametro di dimensionamento adatto 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 il parametro migliore. Per Kinesis Data Streams, considera l'utilizzo del parametro `MillisBehindLatest`, pubblicato dalla Kinesis Client Library (KCL). È necessario calcolare la media di questo parametro tra tutti i consumatori prima di utilizzarlo per il dimensionamento.

Capacità e 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 accessibili e 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 Il dimensionamento automatico 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 correlati costi eccessivi, né troppo bassa, con la conseguenza di latenza e tasso di errore elevati.

Il dimensionamento automatico è un processo latente. Innanzitutto, CloudWatch deve ricevere metriche in tempo reale. Quindi, CloudWatch deve aggregarle 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 devi configurare gli avvisi in modo che la soglia impostata venga superata per alcuni minuti prima che l'avviso cessi. Inoltre, occorre tempo per fornire nuove attività e terminare quelle che non sono più necessarie.

A causa di questi potenziali ritardi nel sistema, dovresti mantenere un certo margine di manovra effettuando un provisioning eccessivo. Il provisioning eccessivo può contribuire a far fronte ai picchi di domanda a breve termine. Ciò consente inoltre all'applicazione di soddisfare le richieste aggiuntive senza raggiungere la saturazione. Come buona norma, dovresti impostare la destinazione di scalabilità tra il 60-80% dell'utilizzo. Ciò 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 un approvvigionamento eccessivo è quello di poter rispondere rapidamente ai guasti delle zone di disponibilità. AWS consiglia 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. Ossia, esegui tre attività ogni due necessarie per il servizio ordinario.

Ottimizzazione 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 aumentare orizzontalmente.

Ridurre al minimo le dimensioni dell'immagine. Le immagini più grandi richiedono più tempo per essere scaricate da un archivio di immagini e decomprese. Pertanto, mantenere le dimensioni delle

immagini più piccole riduce il tempo necessario per l'avvio di un container. 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 zero 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 includere artefatti di costruzione nell'immagine finale. L'utilizzo di build in più fasi può aiutarti in questo senso.
- Compatta le fasi RUN laddove possibile. Ogni fase RUN crea un nuovo livello di immagine, che comporta un ulteriore percorso di andata e ritorno per scaricare il livello. Una singola fase RUN con più comandi uniti da && ha meno livelli di una con più fasi RUN.
- 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 da un altro spazio di archiviazione senza influire sul servizio, allora archivia i dati lì.

Tieni vicine le tue immagini. Maggiore è la latenza di rete, maggiore è il tempo necessario per scaricare l'immagine. Ospita le tue immagini in un repository nella stessa AWS 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 container. 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 agevolarti.

Riduci le soglie di controllo dell'integrità per il bilanciatore del carico. I bilanciatori del carico eseguono controlli dell'integrità prima di inviare traffico all'applicazione. La configurazione predefinita del controllo dell'integrità per un gruppo di destinazioni può richiedere 90 secondi o più. In questo lasso di tempo, il bilanciatore del carico controlla l'integrità e riceve le richieste. La riduzione dell'intervallo dei controlli dell'integrità e del numero di soglie può far sì che l'applicazione accetti il traffico più rapidamente e riduca il carico su altre attività.

Considera 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 mostrare 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 mirate al monitoraggio delle destinazioni. 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. Per ulteriori informazioni, consulta [Scalabilità automatica del servizio](#) nella Guida per gli sviluppatori di Amazon Elastic Container Service.

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 EC2 istanze Amazon, la velocità effettiva massima 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 gp2, utilizza volumi più grandi che offrono un throughput maggiore. Se utilizzi volumi gp3, specifica throughput e IOPS al momento della creazione del volume.

Usa la modalità di rete bridge per le attività in esecuzione su EC2 istanze Amazon. Le attività che utilizzano la modalità di rete bridge su Amazon vengono EC2 avviate più rapidamente delle attività che utilizzano la modalità aws-vpc di rete. Quando viene utilizzata la modalità di rete aws-vpc, Amazon ECS collega un'interfaccia di rete elastica (ENI) all'istanza prima di avviare l'attività. Ciò introduce una latenza aggiuntiva. Tuttavia, ci sono diversi compromessi per l'utilizzo delle reti bridge. Queste attività non dispongono di un proprio gruppo di sicurezza e vi sono alcune implicazioni per il bilanciatore del carico. Per ulteriori informazioni, consulta [Eliminazione del bilanciatore del carico](#) nella Guida per l'utente dell'Elastic Load Balancing.

Gestione degli shock della domanda

Alcune applicazioni subiscono forti shock improvvisi 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 il dimensionamento automatico può richiedere del tempo, consigliamo di aumentare orizzontalmente 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 dell'applicazione. Questo dà al team abbastanza tempo per avere un progetto di pianificazione chiaro. Può pianificare di aumentare orizzontalmente la capacità prima dell'evento e ridurla orizzontalmente dopo l'evento. Per ulteriori informazioni, consulta [Dimensionamento pianificato](#) nella Guida per l'utente di Dimensionamento automatico delle applicazioni.

Se hai un piano Enterprise Support, assicurati di collaborare anche con il Technical Account Manager (TAM). Il TAM può verificare le tue quote di servizio e garantire che le quote necessarie vengano aumentate prima dell'inizio dell'evento. In questo modo non colpirai accidentalmente alcuna quota di servizio. Inoltre, può aiutarti con servizi di preriscaldamento, come i bilanciatori del carico, per assicurarsi che l'evento si svolga senza intoppi.

La gestione degli shock di domanda non programmati è un problema più difficile. Gli shock non programmati, se di portata 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, individua 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, inizia ad aumentare orizzontalmente all'istante 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 il throughput senza scalare affatto l'applicazione.

Infine, puoi prendere in considerazione la possibilità di smontare i servizi monolitici per fronteggiare meglio gli 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. Avere la flessibilità necessaria per aumentare orizzontalmente 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.

Capacità dei cluster Amazon ECS

Puoi fornire capacità a un cluster Amazon ECS in diversi modi. Ad esempio, puoi avviare EC2 istanze Amazon e registrarle nel cluster all'avvio utilizzando l'agente container Amazon ECS. Tuttavia, questo metodo può essere difficile perché devi gestire la scalabilità per conto tuo. Pertanto, consigliamo di utilizzare i provider di capacità di Amazon ECS. I provider di capacità gestiscono la scalabilità delle risorse per te. Esistono tre tipi di fornitori di capacità: Amazon EC2, Fargate e Fargate Spot. [Per ulteriori informazioni sui provider di capacità Fargate, consulta Amazon ECS Clusters per carichi di lavoro Fargate e per carichi di lavoro, consulta Amazon ECS EC2 Clusters for workload. EC2](#)

I provider di capacità Fargate e Fargate Spot gestiscono per conto tuo il ciclo di vita delle attività Fargate. Fargate fornisce capacità su richiesta e Fargate Spot fornisce capacità Spot. Quando avvii un'attività, Amazon ECS fornisce una risorsa Fargate per te. Questa risorsa Fargate include la memoria e le unità CPU che corrispondono direttamente ai limiti a livello di attività dichiarati nella definizione dell'attività. Ogni attività riceve la propria risorsa Fargate, che crea una relazione 1:1 tra l'attività e le risorse di elaborazione.

Le attività eseguite su Fargate Spot sono soggette a interruzioni. Le interruzioni arrivano dopo un avviso di due minuti. Si verificano durante i periodi di forte domanda. Fargate Spot funziona al meglio per carichi di lavoro tolleranti alle interruzioni come lavori in batch, ambienti di sviluppo o staging. È adatto anche per qualsiasi altro scenario in cui l'alta disponibilità e la bassa latenza non siano un requisito.

Puoi eseguire le attività Fargate Spot insieme alle attività Fargate on demand. Usandole insieme, ottieni una capacità “di espansione” di fornitura a un costo inferiore.

Amazon ECS può anche gestire la capacità delle EC2 istanze Amazon per le tue attività. Ogni provider EC2 di capacità Amazon è associato a un gruppo Amazon EC2 Auto Scaling da te specificato. Quando utilizzi il provider di EC2 capacità Amazon, Cluster Auto Scaling mantiene le dimensioni del gruppo Amazon EC2 Auto Scaling per garantire che tutte le attività pianificate possano essere eseguite.

Scelta delle dimensioni delle attività Fargate per Amazon ECS

Per le definizioni delle AWS Fargate attività, è necessario specificare CPU e memoria a livello di attività e non è necessario tenere conto di alcun sovraccarico. Per le attività Fargate, puoi anche specificare CPU e memoria a livello di container. Tuttavia non è obbligatorio farlo. I limiti delle risorse devono essere maggiori o uguali a qualsiasi prenotazione da te dichiarata. Nella maggior parte dei casi, puoi impostarli sulla somma delle prenotazioni di ogni container dichiarato nella definizione dell'attività. Dopodiché, arrotonda il numero alla dimensione Fargate più vicina. Per ulteriori informazioni sulle dimensioni disponibili, consulta [CPU e memoria delle attività](#).

Accelerazione del provisioning della capacità dei cluster Amazon ECS con i fornitori di capacità su Amazon EC2

I clienti che utilizzano Amazon ECS su Amazon EC2 possono sfruttare [Amazon ECS Cluster Auto Scaling \(CAS\) per gestire la scalabilità](#) dei gruppi Amazon Auto EC2 Scaling (ASG). Con il CAS puoi configurare Amazon ECS per scalare automaticamente il tuo ASG e concentrarti solo sull'esecuzione delle tue attività. Amazon ECS garantirà la scalabilità interna e orizzontale dell'ASG in base alle necessità senza ulteriori interventi. I provider di capacità Amazon ECS vengono utilizzati per gestire l'infrastruttura nel cluster, garantendo che vi siano istanze di container sufficienti a soddisfare le esigenze dell'applicazione. Per maggiori informazioni su come funziona il CAS di Amazon ECS, consulta [Approfondimento sul dimensionamento automatico dei cluster Amazon ECS](#).

Poiché CAS si basa su un'integrazione CloudWatch basata su ASG per regolare la capacità del cluster, ha una latenza intrinseca associata alla pubblicazione delle CloudWatch metriche, al tempo impiegato dalla metrica per CapacityProviderReservation violare gli CloudWatch allarmi (sia alto che basso) e al tempo impiegato da un'istanza Amazon appena lanciata per il riscaldamento. EC2 Puoi adottare le seguenti misure per rendere il CAS più reattivo per implementazioni più rapide:

Scalabilità graduale delle dimensioni del provider di capacità

I fornitori di capacità di Amazon ECS alla fine troveranno grow/shrink le istanze di container più adatte a soddisfare le esigenze della tua applicazione. Il numero minimo di istanze che Amazon ECS avvierà è impostato su 1 per impostazione predefinita. Ciò potrebbe comportare un aumento dei tempi di implementazione, qualora fossero necessarie diverse istanze per l'esecuzione delle attività in sospeso. Puoi aumentare la [minimumScalingStepSize](#) utilizzando l'API Amazon ECS per aumentare il numero minimo di istanze scalabili da Amazon ECS per volta. Un valore [maximumScalingStepSize](#) troppo basso può limitare il numero di istanze di container scalate in entrata o in uscita alla volta, rallentando le implementazioni.

 Note

Questa configurazione è attualmente disponibile solo utilizzando o.
[CreateCapacityProvider](#) [UpdateCapacityProvider](#) APIs

Periodo di preparazione 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. Alla fine del periodo di riscaldamento specificato, l'istanza viene conteggiata ai fini delle metriche aggregate dell'ASG e il CAS procede con la successiva iterazione di calcoli per stimare il numero di istanze richieste.

Il valore predefinito per [instanceWarmupPeriod](#) è 300 secondi, che puoi configurare su un valore inferiore utilizzando [CreateCapacityProvider](#) o [UpdateCapacityProvider](#) APIs per un ridimensionamento più reattivo.

Capacità di riserva

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. Questo può ridurre significativamente la velocità di avvio delle attività. Sono quindi disponibili due opzioni.

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 usare la configurazione `Target Capacity` per indicare che vuoi mantenere una capacità di riserva nei cluster. Ad esempio, impostando `Target Capacity` all'80%, indichi che il cluster necessita di una capacità di riserva del 20% in ogni momento. Questa capacità di riserva consente di avviare immediatamente qualsiasi attività autonoma, garantendo che l'avvio delle attività non sia limitato. Il compromesso, nel caso di questo approccio, è il potenziale aumento dei costi legati al mantenimento della capacità di riserva del cluster.

Un approccio alternativo che si può prendere in considerazione è quello di aggiungere margine di manovra al tuo servizio, e non al provider di capacità. Ciò significa che invece di ridurre la configurazione della `Target Capacity` per avviare la capacità di riserva, puoi aumentare il numero di repliche nel servizio modificando la metrica di dimensionamento di monitoraggio delle destinazioni

o le soglie di dimensionamento graduale del dimensionamento automatico. Tieni presente che questo approccio sarà utile solo per i carichi di lavoro soggetti a picchi, ma non avrà alcun effetto quando si implementano nuovi servizi e si passa da 0 a N attività per la prima volta. Per ulteriori informazioni sui criteri di dimensionamento correlati, consulta le [Policy di dimensionamento di monitoraggio delle destinazioni](#) o le [Policy di dimensionamento per fasi](#)

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 ha la possibilità di attivare o disattivare qualsiasi ruolo o utente specifico nell'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 ogni impostazione dell'account.

Nome risorsa	Ulteriori informazioni
containerInsights	Container Insights

Nome risorsa	Ulteriori informazioni
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
fargateEventWindows	AWS Fargate ritiri di attività utilizzando finestre di EC2 eventi
guardDutyActivate	Monitoraggio del runtime (GuardDuty integrazione con Amazon)
dualStackIPv6	VPC a doppio stack IPv6
awsvpcTrunking	Incrementa le interfacce di rete per istanze di container Linux
defaultLogDriverMode	Modalità driver log predefinita

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 servizi, attività e istanze di container 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 che avrai fornito il consenso esplicito, le eventuali nuove risorse create utilizzano 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 influisce sulla risorsa esistente. IDs

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	<code>arn:aws:ecs: <i>region</i>:aws_account_id :container-instance/ <i>container-instance-id</i></code> correnti Nuovo: <code>arn:aws:ecs: <i>region</i>:aws_account_id :container-instance/ <i>cluster-name</i> /<i>container-instance-id</i></code>

Tipo di risorsa	ARN
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 Fargate Amazon ECS Managed Instances e EC2. Dopo aver configurato Container Insights con una migliore osservabilità su Amazon ECS, Container Insights raccoglie automaticamente dati di telemetria dettagliata dell'infrastruttura dal livello di cluster fino al livello di container nel tuo ambiente e visualizza i tuoi dati in dashboard che mostrano una varietà di metriche e dimensioni. Puoi quindi utilizzare queste 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 osservabilità avanzata invece di Container Insights, in quanto fornisce una visibilità dettagliata nell'ambiente del container, riducendo il tempo medio di risoluzione. Per ulteriori informazioni, consulta [Amazon ECS Container Insights con metriche di osservabilità avanzate](#) nella Guida per l'utente Amazon CloudWatch .

L'impostazione predefinita per l'account `containerInsights` è `disabled`.

Container Insights con osservabilità migliorata

Utilizza il comando seguente per attivare Container Insights con osservabilità migliorata.

Configura l'impostazione dell'account `containerInsights` 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 osservabilità migliorata. Usa il comando `update-cluster-settings` 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 osservabilità migliorata. Per ulteriori informazioni, consulta [Modifica delle impostazioni dell'account Amazon ECS](#).

Container Insights

Quando configuri l'impostazione dell'account `containerInsights` su `enabled`, tutti i nuovi cluster hanno Container Insights abilitato per impostazione predefinita. Puoi modificare i cluster esistenti utilizzando `update-cluster-settings`.

Per utilizzare Container Insights, configura l'impostazione dell'account `containerInsights` su `enabled`. Utilizza il comando seguente 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 configuri l'impostazione dell'account `containerInsights` su `enabled`, tutti i nuovi cluster hanno Container Insights abilitato per impostazione predefinita. Usa il comando `update-cluster-settings` 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.

- Per attivare la conformità FIPS-140 puoi utilizzare il comando seguente.

```
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 dell'autorizzazione per le operazioni 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 ci 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 `on`. 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.

- Per abilitare l'autorizzazione all'assegnazione di tag puoi utilizzare il comando seguente.

```
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
  }
}
```

Una volta attivata l'opzione, devi configurare le autorizzazioni appropriate per consentire l'assegnazione di tag alle risorse durante la 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 aderire all'utilizzo dell'autorizzazione all'assegnazione di tag. 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'assegnazione di tag per impostazione predefinita. Puoi abilitare o disabilitare l'impostazione dell'account `tagResourceAuthorization` per verificare la policy IAM.

AWS ha notificato gli account interessati.

Per disabilitare la funzionalità, esegui `put-account-setting-default` con l'opzione `tagResourceAuthorization` impostata su `off`.

- 7 marzo 2024: se hai abilitato l'autorizzazione all'assegnazione di tag, non puoi più disabilitare l'impostazione dell'account.

Ti consigliamo di completare i test delle policy IAM entro questa data.

- 29 marzo 2024: Tutti gli account utilizzano l'autorizzazione all'assegnazione di tag. 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. Puoi scegliere se ritirare immediatamente l'attività o attendere 7 o 14 giorni.

Questa impostazione si applica a livello di account.

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Il periodo di attesa predefinito è di 7 giorni. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (0). Se hai bisogno di più tempo, configura l'opzione 7, o 14 day.

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. Questa è l'impostazione predefinita.
- 14: AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

Per ulteriori informazioni, consulta [put-account-setting-default](#) e consulta il riferimento [put-account-setting](#) all'API di Amazon Elastic Container Service.

Per impostare il periodo di attesa su 14 giorni, puoi eseguire il comando seguente.

```
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
```

AWS Fargate ritiri di attività utilizzando finestre di EC2 eventi

AWS è responsabile dell'applicazione di patch e della manutenzione dell'infrastruttura sottostante per. AWS Fargate Quando AWS determina la necessità di un aggiornamento di sicurezza dell'infrastruttura o per un processo Amazon ECS ospitato su Fargate, è necessario arrestare i processi e avviare nuovi processi sostitutivi. Abilitando questa impostazione dell'account AWS Fargate utilizzerai le finestre EC2 degli eventi per determinare l'ora e il giorno in cui le attività verranno ritirate. Tieni presente che si tratta di un'attivazione unica per un account.

Per abilitare l'utilizzo delle finestre degli EC2 eventi per i ritiri delle attività di Fargate, è possibile utilizzare il seguente comando CLI:

```
aws ecs put-account-setting-default --name fargateEventWindows --value enabled
```

Questa è un'impostazione a livello di account. È possibile associare le finestre EC2 degli eventi alle attività Fargate a livello di account, cluster e servizio utilizzando i seguenti tag di istanza:

- `aws:ecs:serviceArn`: per le attività di servizio
- `aws:ecs:clusterArn`: per le attività appartenenti a un cluster
- `aws:ecs:fargateTask`: impostato su `true` per indirizzare tutte le attività Fargate nell'account

Per ulteriori informazioni su come funzionano le [finestre EC2 degli eventi di Amazon](#) per le attività Amazon ECS eseguite su Fargate, consulta [Passaggio 1: imposta il tempo di attesa dell'attività o utilizza le finestre EC2 degli eventi di Amazon](#)

Incrementa le interfacce di rete per istanze di container Linux

Ogni attività Amazon ECS che utilizza la modalità di rete `awsvpc` riceve la propria interfaccia di rete elastica (ENI), che è collegata all'istanza di container 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. 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.

Amazon ECS supporta il lancio di istanze di container con maggiore ENI densità utilizzando i tipi di istanze Amazon EC2 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'istanza `c5.large` con `awsvpcTrunking` un limite ENI maggiore 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 monitoraggio del runtime 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 [Protezione dei 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 la EC2 capacità, [Utilizzo di un VPC in modalità dual-stack](#) per la capacità delle istanze gestite di Amazon ECS e per la capacità di [Utilizzo di un VPC in modalità dual-stack](#) 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'`dualStackIPv6`account 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 driver log predefinita

Amazon ECS supporta l'impostazione di una modalità di consegna predefinita dei messaggi di log da un container al driver di log scelto. La modalità di distribuzione influisce sulla stabilità dell'applicazione quando il flusso di log dal container al driver di log viene interrotto.

L'impostazione `defaultLogDriverMode` 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 nella `logConfiguration` delle definizioni del container, la modalità specificata utilizzando questa impostazione dell'account verrà utilizzata come predefinita.

La modalità di consegna predefinita è `non-blocking`.

📘 Note

Il 25 giugno 2025, Amazon ECS ha cambiato la modalità predefinita del driver di log da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto al logging. Per continuare a utilizzare la modalità `blocking` dopo questa modifica, procedere in uno dei seguenti modi:

- Impostare l'opzione `mode` nella definizione del container `logConfiguration` come `blocking`.

- Configura l'impostazione dell'account `defaultLogDriverMode` su `blocking`.

Per impostare una modalità di driver di log predefinita su `blocking`, esegui il comando seguente.

```
aws ecs put-account-setting-default --name defaultLogDriverMode --value "blocking"
```

Visualizzazione delle impostazioni dell'account Amazon ECS con la console

Visualizza le impostazioni del tuo account nella console per vedere a quali funzionalità puoi accedere.

Important

Le impostazioni `dualStackIPv6fargateFIPSMODE`, `fargateTaskRetirementWaitPeriod` e `fargateEventWindowsAccount` possono essere visualizzate o modificate solo utilizzando 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 monitoraggio del runtime 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 [Protezione dei carichi di lavoro Amazon ECS con Runtime Monitoring](#).

Important

Le impostazioni `dualStackIPv6fargateFIPSMODE`, `fargateTaskRetirementWaitPeriod` e le impostazioni `fargateEventWindows` dell'account possono essere visualizzate o modificate solo utilizzando 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).
4. Scegliere Aggiorna.
5. (Facoltativo) Per aumentare o diminuire il numero di attività che è possibile eseguire nella modalità di rete awsvpc per ciascuna EC2 istanza, in AWSVPCTrunking, seleziona Trunking. AWSVPC
6. (Facoltativo) Per utilizzare o interrompere l'utilizzo di CloudWatch Container Insights per impostazione predefinita 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. (Facoltativo) Per attivare o disattivare l'autorizzazione all'assegnazione di tag, in Autorizzazione all'assegnazione di tag alle risorse, seleziona o deselecta Autorizzazione all'assegnazione di tag alle risorse.
8. (Facoltativo) Per configurare una modalità di driver di log predefinita per quando una modalità di consegna dei log non è definita nella logConfiguration di un container, in Modalità predefinita dei driver di log scegli una delle seguenti opzioni:
 - Per impostare la modalità predefinita del driver di log come blocking, scegli Blocca.
 - Per impostare la modalità predefinita del driver di log come non-blocking, scegli Non bloccare.
9. Scegli Save changes (Salva modifiche).
10. Nella schermata di conferma, scegliere Conferma per salvare la selezione.

Fasi successive

Se hai attivato Container Insights con osservabilità migliorata o Container Insights, hai la facoltà di 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 Console di gestione AWS 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. Scegliere 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 `dualStackIPv6fargateFIPSPMode`, `fargateTaskRetirementWaitPeriod` e le impostazioni dell'`fargateEventWindowsaccount` 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 container o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità usate. Usa la seguente tabella per determinare i ruoli IAM 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 Fargate o un'istanza esterna e: • estrae un'immagine del contenitore da un repository privato Amazon ECR. • estrae un'immagine di container da un repository privato Amazon ECR in un account diverso da	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		quello 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: • usa 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	Definizione	Quando richiesto	Ulteriori informazioni
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 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 dell'infrastruttura Amazon ECS per i bilanciatori del carico	Questo ruolo consente ad Amazon ECS di gestire le risorse di bilanciamento del carico nei tuoi cluster per tuo conto per le distribuzioni. blue/green	Vuoi usare le blue/green distribuzioni di Amazon ECS.	Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico
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 permette ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• È preferibile usare le istanze gestite da Amazon ECS per la tua capacità. • Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo dell'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 servizi Amazon ECS Service Connect. • Desideri creare gruppi di destinazione di VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS
Profilo dell'istanza	Questo ruolo consente alle istanze gestite da Amazon ECS di assumere il ruolo dell'infrastruttura in modo sicuro.	Utilizzi Amazon ECS Managed Instances nei tuoi cluster.	Profilo delle istanze gestite da Amazon ECS

Architettura per le istanze gestite da Amazon ECS

Amazon ECS Managed Instances è un'opzione di elaborazione completamente gestita per Amazon ECS che ti consente di eseguire carichi di lavoro containerizzati sull'intera gamma di tipi di EC2 istanze Amazon, affidando la gestione dell'infrastruttura a AWS. Con Amazon ECS Managed Instances, puoi accedere a funzionalità di elaborazione specifiche come l'accelerazione GPU, particolari architetture di CPU, elevate prestazioni di rete e tipi di istanze specializzati, AWS gestendo al contempo il provisioning, la scalabilità, l'applicazione di patch e la manutenzione dell'infrastruttura sottostante.

Quando utilizzi Amazon ECS Managed Instances, impacchetti l'applicazione in contenitori e specifichi i requisiti di elaborazione. AWS seleziona automaticamente i tipi di EC2 istanza Amazon generici più ottimizzati in termini di costi che soddisfano le tue esigenze di carico di lavoro oppure puoi specificare gli attributi di istanza desiderati tra cui tipi di istanza, produttori di CPU e acceleratori. Le istanze gestite di Amazon ECS gestiscono completamente tutti gli aspetti dell'infrastruttura, tra cui scalabilità, applicazione di patch e ottimizzazione dei costi, senza rinunciare all'accesso alle funzionalità AWS e alle integrazioni di Amazon. EC2

Le istanze gestite da Amazon ECS supportano container Linux con ottimizzazioni e configurazioni di sicurezza specifiche della piattaforma. Per impostazione predefinita, le istanze gestite da Amazon ECS ottimizzano l'utilizzo dell'infrastruttura collocando più attività più piccole su istanze più grandi, contribuendo a ridurre i costi e migliorare i tempi di avvio delle attività.

In questo argomento vengono descritti i diversi componenti dei processi e dei servizi delle istanze gestite da Amazon ECS oltre a particolari considerazioni sull'utilizzo delle istanze gestite da Amazon ECS con Amazon ECS.

Nozioni di base

Per iniziare a usare Amazon ECS Managed Instances, devi creare i ruoli IAM richiesti e abilitare Amazon ECS Managed Instances nel tuo account. AWS. Quindi è possibile creare un provider di capacità e avviare attività o servizi utilizzando il provider di capacità delle istanze gestite da Amazon ECS.

Per istruzioni dettagliate sulle operazioni di base, consultare:

- [Ulteriori informazioni su come creare un'attività per Amazon ECS Managed Instances](#)
- [Scopri come creare un'attività per Amazon ECS Managed Instances con AWS CLI](#)

fornitori di capacità

Istanze gestite da Amazon ECS provider di capacità per gestire la capacità di calcolo dei carichi di lavoro. È possibile utilizzare il provider di capacità predefinito o creare fornitori di capacità personalizzati con requisiti di istanza specifici.

Sono disponibili le seguenti opzioni di provider di capacità:

- Provider di capacità predefinito: seleziona automaticamente i tipi di istanze per uso generico più ottimizzati in termini di costi per i requisiti dei carichi di lavoro.
- Provider di capacità personalizzati: consentono di specificare gli attributi dell'istanza utilizzando la selezione del tipo di istanza basata sugli attributi, tra cui il numero di vCPU, la memoria, i produttori di CPU, i tipi di acceleratori e tipi di istanza specifici.

Una strategia di provider di capacità può contenere solo un tipo di provider di capacità dal seguente elenco:

- Istanze gestite da Amazon ECS
- Gruppo Auto Scaling
- Fargate/Fargate_SPOT

Selezione e ottimizzazione delle istanze

Amazon ECS sceglie i tipi di istanze per i carichi di lavoro delle Istanze gestite da Amazon ECS utilizzando uno dei seguenti metodi:

- Selezione automatica: quando si utilizza il provider di capacità predefinito, Amazon ECS seleziona automaticamente i tipi di istanze per uso generico più ottimizzati in termini di costi che soddisfano i requisiti di CPU e memoria specificati nella definizione dell'attività.
- Selezione basata sugli attributi: quando si utilizzano i provider di capacità personalizzati, è possibile specificare gli attributi delle istanze, tra cui il numero di vCPU, la dimensione della memoria, i produttori di CPU, i tipi di acceleratori e i tipi di istanza specifici. Amazon ECS seleziona tra tutti i tipi di istanza che corrispondono agli attributi specificati.

Istanze gestite da Amazon ECS ottimizza l'utilizzo e i costi dell'infrastruttura attraverso diversi meccanismi:

- **Posizionamento multi-task:** per impostazione predefinita, Amazon ECS colloca diverse attività più piccole su istanze più grandi per massimizzare l'utilizzo e ridurre i costi.
- **Consolidamento attivo del carico di lavoro:** Amazon ECS identifica quando le istanze di container sono realmente inattive, cercando al contempo di evitare un'interruzione prematura che potrebbe influire sulla disponibilità delle applicazioni o sulle prestazioni di implementazione. Il sistema rispetta il numero minimo e massimo di attività impostate per un servizio, il comportamento di avvio prima dell'interruzione e quello di protezione delle attività.
- **Dimensionamento corretto:** man mano che i requisiti del carico di lavoro cambiano, Amazon ECS lancia istanze sostitutive di dimensioni adeguate alle esigenze attuali.

Amazon ECS utilizza le finestre EC2 degli eventi di Amazon per pianificare le attività di manutenzione durante i periodi di tempo preferiti. Le finestre degli eventi consentono di definire periodi di tempo ricorrenti in cui AWS può eseguire la manutenzione sulle istanze, aiutandoti a ridurre al minimo le interruzioni dei carichi di lavoro allineando la manutenzione alla pianificazione operativa. Per ulteriori informazioni, consulta [Eventi pianificati per le tue istanze](#) nella Amazon EC2 User Guide.

Se è necessario un forte isolamento, è possibile configurare Istanze gestite da Amazon ECS in modo da eseguire ogni attività su un'istanza separata con limiti di isolamento di sicurezza a livello di macchina virtuale.

Definizioni di processi

I processi che utilizzano Istanze gestite da Amazon ECS supportano la maggior parte dei parametri di definizione delle attività Amazon ECS. Istanze gestite da Amazon ECS è compatibile con le definizioni di attività di Fargate esistenti utilizzando la versione della piattaforma 1.4.0, semplificando la migrazione.

Per utilizzare Istanze gestite da Amazon ECS, imposta il parametro di definizione dell'attività `requiresCompatibilities` in modo da includere `MANAGED_INSTANCES`. Le definizioni delle attività possono specificare la compatibilità tra Fargate e Istanze gestite da Amazon ECS per una maggiore flessibilità nelle opzioni di implementazione.

Architettura del sistema operativo e della CPU

Sono supportati i seguenti sistemi operativi:

- Bottlerocket

Sono disponibili 2 architetture per la definizione di attività Amazon ECS, ARM e X86_64.

Quando esegui contenitori Linux su Amazon ECS Managed Instances, puoi utilizzare l'architettura CPU X86_64 o l'architettura per le tue applicazioni basate su ARM ARM64 .

Funzionalità principali

Di seguito sono riportate le caratteristiche principali di Istanze gestite da Amazon ECS:

- Seleziona tipi di EC2 istanze specifici per soddisfare i requisiti della tua applicazione, consentendo l'accesso a funzionalità hardware specializzate come elaborazione accelerata da GPU, funzionalità CPU specifiche e grandi dimensioni di memoria.
- Ottimizzare l'utilizzo delle risorse e i costi con più attività su una singola istanza per impostazione predefinita, a differenza di Fargate che esegue ogni attività nel proprio ambiente isolato.
- Garantisci la conformità alla sicurezza e l'applicazione regolare delle patch alle istanze. ECS Managed Instances avvia il drenaggio delle istanze dopo 14 giorni e sostituisce automaticamente le attività basate sui servizi con nuove istanze.
- Abilitare funzioni avanzate di rete e amministrazione del sistema all'interno dei container utilizzando funzionalità Linux privilegiate, tra cui CAP_NET_ADMIN, CAP_SYS_ADMIN e CAP_BPF.

Ruoli IAM

Istanze gestite da Amazon ECS richiede due ruoli IAM:

- Ruolo dell'infrastruttura: questo ruolo consente di AWS gestire le istanze gestite di Amazon ECS per tuo conto.
- Profilo dell'istanza: un profilo di istanza è un modo per trasferire un ruolo IAM a Istanze gestite da Amazon ECS. Questo profilo viene utilizzato per:
 - Definire le autorizzazioni IAM per le istanze gestite da Amazon ECS che eseguono i carichi di lavoro dei container.
 - Consenti AWS di gestire queste istanze per tuo conto.
 - Abilita le istanze ad accedere ai AWS servizi in base alle autorizzazioni definite nel profilo.

Conformità e sicurezza

Istanze gestite da Amazon ECS implementa più livelli di sicurezza per proteggere i carichi di lavoro:

- **Configurazione sicura:** le istanze gestite di Amazon ECS seguono le migliori pratiche AWS di sicurezza, tra cui nessun accesso SSH, file system root immutabile e controlli di accesso obbligatori a livello di kernel tramite SELinux
- **Patch automatico:** aggiorna AWS regolarmente Amazon ECS Managed Instances con le patch di sicurezza più recenti, rispettando le finestre di manutenzione configurate.
- **Durata limitata dell'istanza:** ECS avvia automaticamente il drenaggio delle istanze dopo 14 giorni, garantendo che le applicazioni vengano eseguite su istanze configurate in modo appropriato con patch di sicurezza. up-to-date
- **Funzionalità privilegiate:** opzionalmente, è possibile abilitare le funzionalità privilegiate di Linux per i carichi di lavoro che le richiedono, come le soluzioni di monitoraggio e osservabilità della rete.

Le istanze gestite da Amazon ECS supportano gli stessi programmi di conformità di Amazon ECS, tra cui PCI-DSS, HIPAA e FedRAMP. Nelle regioni supportate, Istanze gestite da Amazon ECS rispetta le impostazioni degli endpoint FIPS a livello di account per contribuire a raggiungere la conformità FedRAMP.

Rete

Le Istanze gestite da Amazon ECS supportano le modalità di rete `awsvpc` e `host`. La modalità di rete `awsvpc` fornisce a ciascuna attività una propria interfaccia di rete elastica e un indirizzo IP privato all'interno del VPC. Ciò consente controlli granulari dei gruppi di sicurezza e degli ACL di rete a livello di attività. Nella modalità rete `host`, le attività condividono il namespace di rete dell'host delle Istanze gestite da Amazon ECS. Per ulteriori informazioni sulle reti di processi nelle Istanze gestite da Amazon ECS, consultare [Rete di attività di Amazon ECS per istanze gestite Amazon ECS](#).

Archiviazione dell'istanza

Le Istanze gestite da Amazon ECS supportano la configurazione della dimensione del volume di dati Amazon EBS collegato all'istanza. Questo archivio è condiviso tra tutte le attività eseguite sull'istanza e può essere utilizzato per i montaggi vincolati. Il volume può essere montato e condiviso tra container che utilizzano i parametri `volumes`, `mountPoint` e `volumesFrom` nella definizione di attività.

Il volume è collegato durante la creazione dell'istanza. Puoi specificare la dimensione del volume, in GiB, quando crei un provider di capacità delle Istanze gestite da Amazon ECS utilizzando il parametro `storageConfiguration`.

```

{
  ...

  "managedInstancesProvider": {
    "infrastructureRoleArn": "arn:aws:iam::123456789012:role/
ecsInfrastructureRole",
    "instanceLaunchTemplate": {
      "ec2InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/
ecsInstanceProfile",
      "networkConfiguration": {
        "subnets": [
          "subnet-abcdef01234567",
          "subnet-bcdefa98765432"
        ],
        "securityGroups": [
          "sg-0123456789abcdef"
        ]
      },
      "storageConfiguration": {
        "storageSizeinGiB" : 100
      }
    }
  }
  ...
}

```

La dimensione minima di questo volume è 30 GiB e quella massima è 16.384 GiB. Per impostazione predefinita, la dimensione di questo volume è 80 GiB.

L'immagine del container estratta, compressa e non compressa per il processo viene memorizzata nel volume. Per determinare la quantità totale di archiviazione dell'istanza che il processo deve utilizzare come montaggio vincolante, è necessario sottrarre la quantità di archiviazione utilizzata dall'immagine del container dall'importo totale di archiviazione dell'istanza in cui è allocato il processo.

Le prestazioni dei volumi Amazon EBS collegati alle istanze gestite di Amazon ECS corrispondono alle prestazioni delle istanze Amazon corrispondenti, come indicato nella documentazione sulle EC2 istanze [ottimizzate per Amazon EBS nella Amazon](#) User Guide. EC2

È possibile creare snapshot del volume per eseguire un'analisi forense dei problemi di sicurezza o effettuare il debug dell'applicazione. Per ulteriori informazioni sulla creazione di snapshot dei volumi Amazon EBS, consultare [Amazon EBS snapshots](#) nella Guida per l'utente di Amazon EBS. Se la crittografia Amazon EBS è abilitata per impostazione predefinita, il volume verrà crittografato con la AWS KMS chiave specificata per la crittografia per impostazione predefinita. Per ulteriori informazioni sulla crittografia per impostazione predefinita, consultare [Enable Amazon EBS encryption by default](#) nella Guida per l'utente di Amazon EBS.

Oltre a utilizzare il volume di dati collegato all'istanza, è possibile anche configurare i volumi di dati per ogni attività eseguita su Istanze gestite da Amazon ECS. Per ulteriori informazioni sulle opzioni di archiviazione a livello di task disponibili, consultare [Opzioni di archiviazione per le attività di Amazon ECS](#).

Bilanciamento del carico nel servizio

È possibile scegliere di configurare i servizi Amazon ECS usando le Istanze gestite da Amazon ECS per l'utilizzo di Elastic Load Balancing per l'implementazione uniforme del traffico fra i processi del servizio.

I servizi di Amazon ECS sulle Istanze gestite da Amazon ECS supportano i tipi di bilanciatori del carico Application Load Balancer, Network Load Balancer e Gateway Load Balancer. Gli Application Load Balancer indirizzano il traffico HTTP/HTTPS (livello 7), mentre i Network Load Balancer instradano il traffico TCP o UDP (livello 4).

Inoltre, quando si crea gruppi di destinazione per questi servizi, 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 direttamente a un' EC2 istanza Amazon.

Monitoraggio e osservabilità

Amazon ECS Managed Instances offre funzionalità di monitoraggio complete tramite CloudWatch metriche e integrazione con strumenti di osservabilità:

- CloudWatch metriche: monitora l'utilizzo di CPU, memoria, rete e storage sia a livello di attività che di istanza.
- Container Insights: ottengono metriche e log dettagliati delle prestazioni per le applicazioni containerizzate.

- Integrazioni di terze parti: con funzionalità privilegiate abilitate, è possibile eseguire soluzioni avanzate di monitoraggio e osservabilità che richiedono autorizzazioni Linux elevate.

Ottimizzazione dei prezzi e dei costi

Con Amazon ECS Managed Instances, ti viene fatturata l'intera EC2 istanza Amazon che esegue le tue attività. Il prezzo dipende dai tipi di istanze selezionati per i propri carichi di lavoro.

Le Istanze gestite da Amazon ECS offrono diverse funzionalità di ottimizzazione dei costi:

- Ottimizzazione multi-task: massimizza l'utilizzo delle istanze eseguendo più attività su istanze di dimensioni adeguate.

I Savings Plans per il calcolo e le istanze si applicano anche ai carichi di lavoro delle Istanze gestite da Amazon ECS.

Service Quotas

I carichi di lavoro di Amazon ECS Managed Instances sono soggetti alle quote del servizio di istanze Amazon EC2 On-Demand. I servizi Amazon ECS che utilizzano le Istanze gestite da Amazon ECS sono soggetti alle quote di servizio di Amazon ECS.

Per ulteriori informazioni sulle service quotas, consultare:

- [EC2 Endpoint e quote Amazon](#) nel Riferimenti generali di Amazon Web Services
- [Service Quotas di Amazon ECS](#)

Considerazioni sulla migrazione

La migrazione alle Istanze gestite da Amazon ECS è semplice per la maggior parte dei carichi di lavoro:

- Da Fargate: richiede solo una modifica e una ridistribuzione della configurazione del provider di capacità. Le definizioni delle attività esistenti che utilizzano la versione della piattaforma 1.4.0 sono completamente compatibili.
- Da EC2 - Simile alla migrazione a Fargate, ma puoi mantenere l'accesso alle funzionalità di EC2 Amazon come tipi di istanze specifici.

Nella pianificazione della migrazione, tenere conto di quanto segue:

- Le applicazioni devono tollerare la durata dell'istanza di 14 giorni e gli intervalli di manutenzione pianificata.
- Le attività di lunga durata (che superano i 14 giorni) non sono adatte per le Istanze gestite da Amazon ECS.
- Le istanze personalizzate non AMIs sono supportate: le istanze gestite di Amazon ECS utilizzano istanze gestite e AWS ottimizzate per la sicurezza. AMIs

Considerazioni e limitazioni

Le seguenti limitazioni si applicano alle Istanze gestite da Amazon ECS:

- Personalizzato AMIs : l'AMI è di proprietà e gestito da AWS
- Durata dell'istanza: il runtime massimo è di 14 giorni per istanza per garantire l'applicazione di patch di sicurezza e la conformità.
- Accesso SSH: non disponibile per motivi di sicurezza. Usa Amazon ECS Exec per il debug e la risoluzione dei problemi. Operazioni di gestione APIs solo tramite Amazon ECS.

Controlli organizzativi

Alcuni controlli dell'organizzazione possono impedire il corretto funzionamento delle istanze gestite di Amazon ECS. In tal caso, devi aggiornare questi controlli per consentire ad Amazon ECS di disporre delle autorizzazioni necessarie per gestire le EC2 istanze per tuo conto.

Amazon ECS utilizza un ruolo di infrastruttura per avviare le EC2 istanze che supportano Amazon ECS Managed Instances. Questo ruolo di infrastruttura è un [ruolo IAM](#) creato nel tuo account che consente ad Amazon ECS di gestire le istanze gestite di Amazon ECS per tuo conto. [Le policy di controllo dei servizi](#) (SCPs) si applicano sempre alle azioni eseguite con i ruoli dell'infrastruttura. Ciò consente a un SCP di inibire le operazioni di Amazon ECS Managed Instances. L'evento più comune si verifica quando viene utilizzato un SCP per limitare le Amazon Machine Images (AMIs) che possono essere lanciate. Per consentire il funzionamento delle istanze gestite di Amazon ECS, modifica SCP per consentire l'avvio AMIs dagli account AMI di Amazon ECS Managed Instances.

Puoi anche utilizzare la AMIs funzionalità [EC2 Consentito](#) per limitare la visibilità di altri account. AMIs Se utilizzi questa funzionalità, devi espandere i criteri relativi alle immagini per includere anche gli account AMI Amazon ECS Managed Instances nelle regioni di interesse.

Esempio di SCP per bloccare tutte le istanze AMIs ad eccezione di Amazon ECS Managed Instances AMIs

L'SCP riportato di seguito impedisce le chiamate `ec2:RunInstances` a meno che l'AMI non appartenga all'account AMI Amazon ECS Managed Instances per us-west-2 o us-east-1.

Note

È importante non utilizzare la chiave di contesto. `ec2:Owner` Amazon possiede gli account AMI Amazon ECS Managed Instances e il valore di questa chiave sarà sempre lo stesso. `amazon` La creazione di un SCP che consenta l'avvio AMIs se `ec2:Owner` è `amazon` consentirà il lancio di qualsiasi AMI di proprietà di Amazon, non solo quelle per Amazon ECS Managed Instances.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyAMI",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": "arn:*:ec2:*:image/ami-*",
      "Condition": {
        "StringNotEquals": {
          "aws:ResourceAccount": [
            "187296253231",
            "260073348889"
          ]
        }
      }
    }
  ]
}
```

Account AMI Amazon ECS Managed Instances

AWS gli account che variano in base alla regione ospitano Amazon ECS Managed Instances public. AMIs

AWS Regione	Account
af-south-1	070957084703
ap-east-1	587573215167
ap-northeast-1	679336465495
ap-northeast-2	309903600357
ap-northeast-3	384570461223
ap-south-1	062344138989
ap-south-2	624198668379
ap-southeast-1	832199679391
ap-southeast-2	552073033681
ap-southeast-3	368903466070
ap-southeast-4	696793786439
ap-southeast-5	003457290689
ap-southeast-6	465836752572
ap-southeast-7	622515864387
ca-central-1	853167153192
ca-west-1	899469777611
eu-central-1	832570432258
eu-central-2	041659148495
eu-north-1	851563870067
eu-south-1	766433696616

AWS Regione	Account
eu-south-2	003380494496
eu-west-1	986619735082
eu-west-2	591706807364
eu-west-3	108582616801
il-central-1	009537862704
me-central-1	540883425316
me-south-1	181438624895
mx-central-1	210749644920
sa-east-1	591338347621
us-east-1	260073348889
us-east-2	292185169523
us-west-1	187296253231
us-west-2	491085424538

Tipi di Istanze gestite da Amazon ECS

Amazon ECS Managed Instances ti consente di selezionare tipi di EC2 istanze specifici per le tue applicazioni containerizzate.

Famiglie di istanze delle Istanze gestite da Amazon ECS

Sono supportati i seguenti tipi di istanza:

Uso generico

- m5, m5a, m5ad, m5d, m5dn, m5n, m5zn: elaborazione, memoria e rete bilanciate

- m6a, m6g, m6gd, m6i, m6id, m6idn, m6in: ultima generazione con prestazioni migliorate
- m7a, m7g, m7gd, m7i, m7i-flex: istanze per uso generico di nuova generazione
- m8g, m8gd: istanze per uso generico ARM di ultima generazione
- t3, t3a, t4g: istanze con prestazioni espandibili (escluse le dimensioni delle istanze nano e micro)

Ottimizzata per il calcolo

- c5, c5a, c5ad, c5d, c5n: processori ad alte prestazioni per applicazioni a elaborazione intensiva
- c6a, c6g, c6gd, c6i, c6id, c6in: istanze ottimizzate per il calcolo di ultima generazione
- c7a, c7g, c7gd, c7gn, c7i, c7i-flex: istanze ottimizzate per il calcolo di nuova generazione
- c8g, c8gd, c8gn: istanze ARM di ultima generazione ottimizzate per il calcolo
- hpc6a, hpc6id, hpc7a: istanze di calcolo ad alte prestazioni

Ottimizzata per la memoria

- r5, r5a, r5ad, r5b, r5d, r5dn, r5n: rapporto elevato per applicazioni che richiedono molta memoria memory-to-vCPU
- r6a, r6g, r6gd, r6i, r6id, r6idn, r6in: istanze ottimizzate per la memoria di ultima generazione
- r7a, r7g, r7gd, r7i, r7iz: istanze ottimizzate per la memoria di nuova generazione
- r8g, r8gd: istanze ottimizzate per la memoria ARM di ultima generazione
- u-3tb1, u7i-6tb, u7i-8tb, u7i-12tb, u7in-24tb, u7in-32tb: istanze a memoria elevata con un massimo di 32 TB di RAM
- x2gd, x2idn, x2iedn, x2iezn: memoria estrema per database e analisi in memoria
- x8g: istanze di memoria estrema di ultima generazione
- z1d: archiviazione SSD e ad NVMe alta frequenza

Ottimizzata per l'archiviazione

- d3, d3en: archiviazione HDD ad alta densità per file system distribuiti
- i4g, i4i: istanze ottimizzate per l'archiviazione di ultima generazione
- i7i, i7ie, i8g: istanze di archiviazione ad alte prestazioni di nuova generazione
- im4gn, is4gen: istanze di archiviazione ottimizzate per la rete

Calcolo accelerato

- g4dn: NVIDIA GPUs T4 per inferenza e grafica di apprendimento automatico
- g5, g5g: NVIDIA A10G per grafica e ML ad alte prestazioni GPUs
- g6, g6e, g6f: istanze GPU di ultima generazione
- gr6, gr6f: istanze GPU con NVIDIA L4 Tensor Core e rapporto 1:8 vCPU:RAM per carichi di lavoro grafici GPUs
- p3dn: GPUs NVIDIA V100 per formazione in deep learning e HPC
- p4d: NVIDIA A100 per la formazione ML ad alte prestazioni GPUs
- p5: Ultima generazione con NVIDIA H100 GPUs
- p6-b200: nuova generazione con NVIDIA B200 GPUs

Metodi di selezione delle istanze

Le Istanze gestite da Amazon ECS offrono due metodi per selezionare i tipi di istanze:

- Selezione del tipo di istanza specifico: si specifica esplicitamente il tipo di EC2 istanza da utilizzare per le attività.
- Selezione del tipo di istanza basata sugli attributi: specifichi gli attributi (come vCPU, memoria e architettura) richiesti dall'applicazione e le Istanze gestite da Amazon ECS selezionano un tipo di istanza appropriato.

Selezione del tipo di istanza specifico

Con la selezione del tipo di istanza specifico, specifichi esplicitamente il tipo di EC2 istanza da utilizzare per le attività di Amazon ECS Managed Instances. Ciò è utile quando l'applicazione richiede un tipo di istanza specifico con caratteristiche hardware particolari.

Selezione del tipo di istanza basata su attributi

Con la selezione del tipo di istanza basata sugli attributi, si specificano gli attributi richiesti dall'applicazione, e le Istanze gestite da Amazon ECS selezionano un tipo di istanza appropriato che soddisfa questi requisiti. Ciò offre maggiore flessibilità e può contribuire a garantire che le attività vengano eseguite correttamente anche se non sono disponibili tipi di istanze specifici.

I seguenti attributi sono supportati per la selezione del tipo di istanza basata su attributi:

- **Architettura della CPU:** l'architettura della CPU (X86_64 o). ARM64
- **instanceGeneration:** la generazione dell'istanza (corrente, precedente o tutte).
- **burstablePerformance:** se includere istanze a prestazioni espandibili (incluse, escluse o obbligatorie).
- **CPUManufacturer:** il produttore della CPU (intel, amd o or). amazon-web-services
- **networkBandwidth:** la larghezza di banda della rete minima in Gbps.
- **networkInterfaceCount:** Il numero minimo di interfacce di rete.
- **localStorage:** se è richiesta l'archiviazione locale (inclusa, esclusa o richiesta).
- **localStorageType:** il tipo di archiviazione locale (hdd o ssd).

Ottimizzazione dei costi

Le Istanze gestite da Amazon ECS supportano diverse funzionalità che aiutano a ottimizzare il costo dei carichi di lavoro containerizzati:

- **Savings Plans (SPs):** le istanze gestite di Amazon ECS beneficiano dei Savings Plans che hai acquistato per i tipi di istanza utilizzati dalle tue attività.
- **Istanze riservate (RIs):** le attività di Amazon ECS Managed Instances possono trarre vantaggio da RIs ciò che hai acquistato per i tipi di istanze utilizzati dalle tue attività.
- **Istanze Spot:** puoi configurare il provider di capacità di Amazon ECS Managed Instances per utilizzare EC2 le istanze Spot.
- **Posizionamento multi-task:** le Istanze gestite da Amazon ECS collocano più attività su una singola istanza per uso generico per impostazione predefinita, ottimizzando l'utilizzo delle risorse e i costi.

Best practice per la selezione delle istanze per le Istanze gestite da Amazon ECS

La selezione della giusta configurazione dell'istanza per i carichi di lavoro delle Istanze gestite da Amazon ECS è fondamentale per ottimizzare le prestazioni, i costi e l'utilizzo delle risorse. Amazon ECS offre opzioni flessibili di selezione delle istanze che consentono di bilanciare i requisiti delle applicazioni con l'efficienza dei costi. Le seguenti best practice ti aiutano a prendere decisioni informate sulla selezione delle istanze per i tuoi carichi di lavoro containerizzati.

1. Utilizzare il provider di capacità predefinito delle Istanze gestite da Amazon ECS

Amazon ECS sceglie le istanze più convenienti che soddisfano i seguenti requisiti relativi alla definizione delle attività e ai parametri di servizio:

Definizione di attività

- `operatingSystemFamily`
- `cpuArchitecture`
- `cpu`
- `memory`

Definizioni di servizi

- `placementConstraints`
- `placementStrategy`

2. Utilizzare la selezione basata sugli attributi per la maggior parte dei carichi di lavoro per offrire flessibilità e migliorare le percentuali di successo del posizionamento

La selezione delle istanze basata sugli attributi consente ad Amazon ECS di scegliere tra una gamma più ampia di tipi di istanze che soddisfano i requisiti specificati. Questo approccio aumenta la probabilità di successo nell'inserimento delle attività e fornisce una migliore ottimizzazione dei costi, consentendo ad Amazon ECS di selezionare le istanze più convenienti disponibili al momento del lancio.

3. Utilizzare tipi di istanze specifici solo quando le applicazioni hanno requisiti hardware specifici

Riservare una selezione specifica del tipo di istanza per carichi di lavoro che richiedono caratteristiche hardware particolari, come accelerazione GPU, processori ad alta frequenza o funzionalità di rete specializzate. Per le applicazioni generiche, la selezione basata sugli attributi offre in genere una maggiore flessibilità e ottimizzazione dei costi.

4. Scegliere risorse bilanciate per evitare un eccesso di approvvigionamento e costi inutili

Selezionare configurazioni di istanze che soddisfino al meglio i requisiti di CPU e memoria dell'applicazione. Evitare un eccessivo approvvigionamento di risorse, poiché ciò comporta costi più elevati e una riduzione dell'efficienza. Utilizzare i dati di monitoraggio per comprendere i modelli di utilizzo effettivo delle risorse e modificare di conseguenza la selezione delle istanze.

5. Combinare tipi di istanze per applicazioni con carichi di lavoro diversi per bilanciare prestazioni e costi

Per applicazioni con requisiti prestazionali diversi o modelli di carico di lavoro diversi, prendere in considerazione l'utilizzo di più provider di capacità con diverse configurazioni di istanze. Questo approccio consente di ottimizzare i costi utilizzando tipi di istanze appropriati per i diversi componenti dell'applicazione, mantenendo al contempo le prestazioni laddove necessario.

Comportamento dell'immagine del container Amazon ECS Managed Instances

Il tempo impiegato da un container per l'avvio varia in base alla sua immagine sottostante. Ad esempio, un'immagine più grande (versioni complete di Debian, Ubuntu e Amazon2) 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 Amazon ECS avvia un'attività che viene eseguita su Amazon ECS Managed Instances, Amazon ECS recupera l'immagine 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 di pulizia automatica delle immagini. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza gestita di Amazon ECS. In questo modo si ha la garanzia che non vengano eseguite estrazioni di immagini non necessarie.

Applicazione di patch nelle Istanze gestite da Amazon ECS

In Amazon ECS Managed Instances, l'applicazione di patch è un processo di manutenzione fondamentale in cui gestisce e aggiorna AWS automaticamente il software sulle istanze di container gestite per garantire sicurezza e conformità mantenendo al contempo la disponibilità del carico di lavoro attraverso un processo controllato e configurabile.

Ciclo di vita dell'istanza

Per impostazione predefinita, le istanze di container gestite da Amazon ECS operano con un ciclo di vita standardizzato di 14-21 giorni. Amazon ECS avvia un corretto drenaggio del carico di lavoro il giorno 14 dal lancio dell'istanza, con la chiusura finale entro il giorno 21. Amazon ECS supporta il draining precoce in circostanze specifiche:

- Rilevamento di vulnerabilità di sicurezza sul software

- Peggioramento dello stato di salute dell'hardware
- Per rispettare le finestre degli eventi configurate dal cliente

Questo approccio mantiene la conformità del sistema rispettando al contempo i requisiti di manutenzione definiti dal cliente.

Finestre degli eventi e pianificazione della manutenzione

AWS gestisce il ciclo di vita di un'istanza di container gestita attraverso processi automatizzati in background che monitorano il timestamp di creazione e le pianificazioni di manutenzione di un nodo. All'avvio dell'istanza, AWS imposta un programma di drenaggio predefinito di 14 giorni e valuta tutte le finestre degli eventi configurate dal cliente.

Puoi pianificare le attività di manutenzione per Amazon ECS Managed Instances configurando le finestre degli eventi. Puoi associare una o più istanze gestite di Amazon ECS a una finestra di evento utilizzando tag di istanza IDs o di istanza. Quando una finestra di evento è etichettata con un valore specifico, Amazon ECS mappa questi tag alle istanze gestite Amazon ECS corrispondenti dei cluster corrispondenti e pianifica la manutenzione dell'istanza durante i periodi di tempo definiti con la massima diligenza.

Per ulteriori informazioni sulle finestre degli eventi, consulta [Crea finestre di eventi personalizzate per gli eventi pianificati che influiscono sulle tue EC2 istanze Amazon](#) nella Amazon EC2 User Guide.

Se esistono finestre di eventi, AWS modifica il programma di esaurimento per allinearli a queste finestre, il che può comportare un esaurimento anticipato rispetto al periodo predefinito di 14 giorni per rispettare la finestra dell'evento specificata. Le modifiche alla finestra degli eventi riguardano solo le istanze di container gestite appena lanciate, garantendo una pianificazione della manutenzione predittiva.

Fino all'orario di esaurimento programmato, Amazon ECS continua le normali operazioni di posizionamento delle attività sulle istanze di container gestite secondo la configurazione del cliente.

Sequenza di manutenzione

All'ora di manutenzione designata, Amazon ECS inizia la sequenza di manutenzione richiamando l'API `UpdateContainerInstancesState` per avviare un corretto drenaggio del carico di lavoro. Durante il periodo di interruzione regolare, Amazon ECS tenta di interrompere il carico di lavoro sull'istanza contrassegnata per il drenaggio.

Amazon ECS utilizza una start-before-stop strategia per le attività di servizio (o secondo la configurazione del servizio Amazon ECS), garantendo l'avvio delle attività sostitutive prima di interrompere quelle esistenti, riducendo al minimo l'interruzione del servizio. Nel corso di questo processo, i servizi Amazon ECS rispettano tutte le configurazioni di implementazione del servizio, continuando i tentativi di esaurimento fino al giorno 21 dal lancio dell'istanza.

Se il drenaggio non viene completato entro il giorno 21, Amazon ECS esegue l'API `DeregisterContainerInstance` per arrestare l'istanza di container gestito e i relativi carichi di lavoro rimanenti per mantenere la conformità e applicare all'istanza gestita le patch più recenti.

Considerazioni sulla sicurezza per le Istanze gestite da Amazon ECS

Amazon ECS Managed Instances offre un'esperienza di calcolo in container completamente gestita che ti consente di eseguire carichi di lavoro su tipi di EC2 istanze Amazon specifici, delegando al contempo le responsabilità di sicurezza. AWS Questo argomento descrive il modello di sicurezza, le caratteristiche e le considerazioni relative all'utilizzo delle Istanze gestite da Amazon ECS.

Modello di sicurezza

Le Istanze gestite da Amazon ECS implementano un modello di sicurezza completo che bilancia flessibilità e protezione:

- **AWS-infrastruttura gestita:** AWS controlla il ciclo di vita delle istanze gestite e gestisce le patch di sicurezza, eliminando la possibilità di errori umani e manomissioni.
- **Nessun accesso amministrativo:** il modello di sicurezza è bloccato e vieta l'accesso amministrativo alle istanze gestite.
- **Posizionamento multi-task:** per impostazione predefinita, le Istanze gestite da Amazon ECS collocano più attività su una singola istanza per ottimizzare costi e utilizzo, il che allenta il vincolo di isolamento del carico di lavoro rispetto a Fargate.
- **Isolamento dei dati:** sebbene AWS controlli il ciclo di vita delle istanze e il posizionamento delle attività, AWS non può accedere alle istanze gestite o accedere ai dati dei clienti.

Funzionalità di sicurezza

Le Istanze gestite da Amazon ECS includono diverse funzionalità di sicurezza integrate progettate per proteggere i carichi di lavoro e mantenere una solida posizione di sicurezza. Queste funzionalità

vanno dall'applicazione automatica delle patch di sicurezza al supporto di funzionalità Linux privilegiate quando necessario.

Best practice di sicurezza

Le istanze gestite sono configurate in base alle migliori pratiche di AWS sicurezza, tra cui:

- Nessun accesso SSH: l'accesso remoto alla shell è disabilitato per impedire accessi non autorizzati.
- Filesystem del root immutabile: il filesystem del root non può essere modificato, garantendo l'integrità del sistema.
- Controlli di accesso obbligatori a livello di kernel: SELinux forniscono un'ulteriore applicazione della sicurezza a livello di kernel.

Applicazione automatica delle patch di sicurezza

Le Istanze gestite da Amazon ECS aiutano a migliorare il livello di sicurezza dei carichi di lavoro attraverso l'applicazione di patch automatizzate:

- Aggiornamenti di sicurezza regolari: le istanze vengono regolarmente aggiornate con le ultime patch di sicurezza entro le AWS finestre di manutenzione configurate.
- Durata limitata delle istanze: la durata massima di un'istanza in esecuzione è limitata a 14 giorni per garantire che le applicazioni vengano eseguite su istanze configurate in modo appropriato con patch di sicurezza. up-to-date
- Controllo delle finestre di manutenzione: puoi utilizzare la funzionalità delle finestre EC2 degli eventi di Amazon per specificare quando Amazon ECS deve sostituire le istanze con istanze con patch.

Funzionalità Linux privilegiate

Le Istanze gestite da Amazon ECS supportano software che richiedono privilegi Linux elevati, abilitando soluzioni di monitoraggio e sicurezza avanzate:

- Funzionalità supportate: è possibile attivare tutte le funzionalità Linux privilegiate, tra cui CAP_NET_ADMIN, CAP_SYS_ADMIN e CAP_BPF.
- Soluzioni popolari: ciò consente di eseguire le più diffuse soluzioni di monitoraggio e osservabilità della rete come Wireshark e Datadog.

- Configurazione esplicita richiesta: è necessario configurare in modo esplicito il provider di capacità delle Istanze gestite da Amazon ECS per abilitare funzionalità Linux privilegiate, poiché ciò potrebbe comportare rischi di sicurezza aggiuntivi per le applicazioni.

Important

L'attivazione delle funzionalità privilegiate di Linux può esporre le attività a rischi di sicurezza aggiuntivi. Abilitare queste funzionalità solo quando richiesto dalle applicazioni e assicurarsi di comprendere le implicazioni sulla sicurezza.

Conformità e supporto normativo

Le Istanze gestite da Amazon ECS mantengono lo stesso livello di conformità di Amazon ECS:

- Programmi di conformità: Amazon ECS Managed Instances rientra nell'ambito degli stessi programmi di AWS garanzia di Amazon ECS, inclusi PCI-DSS, HIPAA e FedRAMP.
- Endpoint FIPS: Amazon ECS Managed Instances rispetta le impostazioni a livello di account per l'utilizzo degli endpoint FIPS nelle regioni per contribuire a raggiungere la conformità FedRAMP. AWS
- Chiavi gestite dal cliente: supporta le funzionalità di sicurezza necessarie per il raggiungimento della conformità, come le chiavi gestite dal cliente per la crittografia.

Considerazioni FIPS-140 sulle istanze gestite di Amazon ECS

Considera quanto segue quando utilizzi la conformità FIPS-140 su istanze gestite di Amazon ECS:

- Le istanze gestite conformi allo standard FIPS-140 sono disponibili solo nelle regioni. AMIs AWS GovCloud (US)
- Le istanze gestite di Amazon ECS supportano FIPS-140-3
- La conformità FIPS-140 è abilitata per impostazione predefinita nelle regioni. AWS GovCloud (US)
Se devi eseguire carichi di lavoro senza la conformità FIPS, disattiva la conformità FIPS nella configurazione Managed Instances Capacity Provider.
- Le `cpuArchitecture` tue attività devono essere `X86_64` conformi allo standard FIPS-140.

Disattiva FIPS sulle istanze gestite di Amazon ECS

Per impostazione predefinita, i fornitori di capacità di Amazon ECS Managed Instances in AWS GovCloud (US) Regions avviano la versione conforme a FIPS. AMIs Scegli di disabilitare la conformità FIPS-140 quando crei un nuovo Amazon ECS Managed Instances Capacity Provider. Segui questi passaggi per creare un nuovo Capacity Provider senza conformità FIPS.

1. Disabilita la conformità FIPS-140 su Capacity Provider.

```
aws ecs create-capacity-provider \
  --cluster cluster-name \
  --name capacity-provider-name \
  --managed-instances-provider '{
    "infrastructureRoleArn": "infrastructure-role-arn",
    "instanceLaunchTemplate": {
      "ec2InstanceProfileArn": "instance-profile-arn",
      "fipsEnabled": false,
      "networkConfiguration": {
        "subnets": [subnet-id],
        "securityGroups": [security-group-id]
      }
    }
  }'
```

2. Facoltativamente, è possibile utilizzare ECS Exec per eseguire il seguente comando per verificare lo stato di conformità FIPS-140 per un provider di capacità.

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"
```

Considerazioni relative alla sicurezza

Quando si utilizzano le Istanze gestite da Amazon ECS, ci sono diverse importanti considerazioni sulla sicurezza da comprendere e pianificare. Queste considerazioni ti aiutano a prendere decisioni informate sull'architettura del carico di lavoro e sui requisiti di sicurezza.

Modello di sicurezza multitasking

Il modello di posizionamento multi-task predefinito nelle Istanze gestite da Amazon ECS è diverso dall'isolamento a singola attività di Fargate:

- **Risorse condivise di istanze:** è possibile eseguire più attività sulla stessa istanza, esponendo potenzialmente un'attività alle vulnerabilità di altre attività in esecuzione sulla stessa istanza o nello stesso cluster ECS.
- **Opzione single-task:** è possibile configurare le Istanze gestite da Amazon ECS per utilizzare la modalità single-task per i clienti che richiedono il modello di sicurezza Fargate predefinito con limite di isolamento di sicurezza a livello di macchina virtuale.
- **Compromesso tra costi e sicurezza:** la modalità multi-task consente di ottimizzare i costi e velocizzare i tempi di startup delle attività, mentre la modalità single-task offre un maggiore isolamento.

Gestione delle interruzioni dell'istanza

È importante progettare le applicazioni in modo da tollerare le interruzioni quando si utilizzano le Istanze gestite da Amazon ECS:

- **Tolleranza all'interruzione:** utilizzare le Istanze gestite da Amazon ECS con applicazioni che tollerano l'interruzione dei servizi o delle attività sottostanti.
- **Carichi di lavoro basati sui servizi:** utilizzare i servizi Amazon ECS per la sostituzione automatica delle attività o eseguire carichi di lavoro con una durata controllata e limitata non superiore a 14 giorni su attività autonome.
- **Arresto regolare:** configurare il periodo di tolleranza per la chiusura delle attività per controllare l'impatto delle interruzioni.

Accesso e privacy dei dati

Le Istanze gestite da Amazon ECS mantengono rigorosi controlli di accesso ai dati:

- Nessun accesso ai dati dei clienti: sebbene AWS controlli il ciclo di vita delle istanze gestite e il posizionamento delle attività sulle istanze, AWS non può accedere alle istanze gestite o accedere ai dati dei clienti.
- Solo parametri e log: AWS acquisisce solo i parametri e i log correlati necessari per fornire le funzionalità di Amazon ECS Managed Instances.
- Modello di sicurezza bloccato: il modello di sicurezza vieta l'accesso amministrativo, eliminando la possibilità di errori umani e manomissioni.

Best practice di sicurezza

Seguire queste best practice quando si utilizzano le Istanze gestite da Amazon ECS:

- Valutare il modello di sicurezza: prendere una decisione consapevole sull'adozione delle Istanze gestite da Amazon ECS in base ai requisiti di sicurezza, in particolare per quanto riguarda il modello di posizionamento multi-task.
- Usare la modalità single-task quando necessario: se i carichi di lavoro richiedono un isolamento più forte, configurare le Istanze gestite da Amazon ECS per utilizzare la modalità single-task.
- Ridurre al minimo le funzionalità privilegiate: abilitare le funzionalità privilegiate di Linux solo quando assolutamente necessario e comprendere i rischi per la sicurezza associati.
- Pianificare le interruzioni: progettare le applicazioni per gestire le sostituzioni delle istanze in modo corretto, soprattutto considerando la durata massima delle istanze di 14 giorni.
- Configura le finestre di manutenzione: utilizza le finestre EC2 degli eventi per controllare quando avvengono le sostituzioni delle istanze per ridurre al minimo l'impatto sui carichi di lavoro.
- Monitoraggio e verifica: esaminare regolarmente la configurazione delle Istanze gestite da Amazon ECS e monitorare eventuali eventi o modifiche relativi alla sicurezza.

Abilitazione del controllo della crittografia VPC per le istanze gestite di Amazon ECS

Amazon ECS Managed Instances supporta VPC Encryption Controls, una funzionalità di sicurezza e conformità che fornisce un controllo centralizzato per monitorare e applicare la crittografia in transito per tutti i flussi di traffico all'interno e all'interno di una regione. VPCs Quando VPC Encryption Controls è abilitato sulla tua sottorete, puoi specificare i tipi di istanze che supportano la crittografia in

transito nel tuo provider di capacità personalizzata Amazon ECS Managed Instances, garantendo che i carichi di lavoro di Amazon ECS Managed Instances vengano eseguiti con crittografia in transito.

Prerequisiti

Prima di iniziare è necessario disporre di quanto segue:

- Un VPC con crittografia in transito abilitata sulle sottoreti. Per ulteriori informazioni, consulta la documentazione dei [controlli di crittografia VPC](#).
- Un provider di capacità personalizzato Amazon ECS Managed Instances. Per ulteriori informazioni, consulta [Architettura per le istanze gestite da Amazon ECS](#).

Identifica i tipi di istanze compatibili

I tipi di EC2 istanze Amazon devono soddisfare due requisiti:

1. Supporta la crittografia VPC in transito: utilizza il seguente AWS CLI comando per elencare i tipi di EC2 istanze Amazon che supportano la crittografia in transito:

```
aws ec2 describe-instance-types \
  --filters Name=network-info.encryption-in-transit-supported,Values=true \
  --query "InstanceTypes[*].[InstanceType]" \
  --output text | sort
```

2. Supportato da Amazon ECS Managed Instances - Tutti i tipi di EC2 istanze Amazon supportati da Amazon ECS Managed Instances sono documentati in. [Tipi di Istanze gestite da Amazon ECS](#)

Se hai requisiti aggiuntivi (come esigenze specifiche di CPU, memoria o architettura), filtra ulteriormente i tipi di istanze compatibili in base ai requisiti del carico di lavoro.

Crea un cluster con supporto per la crittografia VPC

Per configurare Amazon ECS Managed Instances per la crittografia VPC in transito:

1. Crea un nuovo cluster e seleziona Fargate e Istanze gestite per l'infrastruttura.
2. Seleziona Usa personalizzato — avanzato per accedere a parametri di configurazione aggiuntivi.
3. Nei tipi di istanze consentite, aggiungi solo i tipi di istanza specifici che supportano la crittografia VPC in transito.

Se configurate in questo modo, Amazon ECS Managed Instances avvierà solo tipi di EC2 istanze Amazon che supportano la crittografia VPC in transito.

Considerazioni

- Istanze a prestazioni stabili: i tipi di istanze T3, T3a e T4g non supportano la crittografia VPC in transito e non possono essere utilizzate nelle sottoreti con il controllo della crittografia abilitato in modalità Enforced.
- Transizioni di modalità: puoi trasferire la sottorete VPC dalla modalità Monitor alla modalità Enforced solo se tutte le istanze in esecuzione supportano la crittografia VPC in transito.
- Errori di avvio delle attività: in modalità applicata, le attività non verranno avviate se si specificano tipi di istanze che non supportano la crittografia in transito.

Risoluzione dei problemi

Errori di avvio delle attività in modalità applicata

Se le attività non vengono avviate, verifica che tutti i tipi di istanza specificati supportino la crittografia VPC in transito utilizzando il AWS CLI comando fornito sopra.

Impossibile passare alla modalità Enforced

Utilizza la console o il `GetVpcResourcesBlockingEncryptionEnforcement` comando per identificare le risorse che non applicano la crittografia in transito.

Per ulteriori informazioni sui controlli di crittografia VPC, consulta la documentazione dei controlli di crittografia [VPC](#).

Ottimizzazione dell'infrastruttura delle Istanze gestite da Amazon ECS

Amazon ECS Managed Instances effettua automaticamente il provisioning delle EC2 istanze della giusta dimensione in base alla configurazione del Capacity Provider e alle attuali richieste di carico di lavoro, assicurando che le applicazioni containerizzate dispongano delle risorse di elaborazione appropriate sin dal momento in cui vengono distribuite. Man mano che i modelli di traffico delle applicazioni si evolvono e i requisiti del carico di lavoro cambiano nel tempo, Amazon ECS Managed Instances monitora e ottimizza continuamente l'infrastruttura regolando in modo

intelligente le dimensioni delle istanze in base alle esigenze attuali, sostituendo in modo proattivo le istanze che si sono allontanate da configurazioni ottimali e bilanciando dinamicamente l'efficienza dei costi, le prestazioni delle applicazioni e l'affidabilità del sistema. Questo sistema di gestione delle risorse funziona senza alcun intervento manuale, riducendo i costi dell'infrastruttura e mantenendo un'elevata disponibilità per le applicazioni.

L'ottimizzazione dell'infrastruttura presenta i seguenti vantaggi:

- **Ottimizzazione dei costi:** riduce i costi dell'infrastruttura massimizzando l'utilizzo delle risorse ed eliminando la capacità inattiva
- **Miglioramento delle prestazioni:** ottimizza il posizionamento dei carichi di lavoro in base ai requisiti delle risorse e alle caratteristiche prestazionali
- **Semplicità operativa:** automatizza le decisioni complesse sulla gestione delle risorse senza richiedere l'intervento manuale
- **Miglioramento dell'affidabilità:** mantiene un'elevata disponibilità attraverso la distribuzione intelligente del carico di lavoro e il monitoraggio dello stato

Amazon ECS Managed Instances esegue due tipi di ottimizzazioni dell'infrastruttura per massimizzare l'efficienza e ridurre i costi:

Rilevamento delle istanze inattive

Identifica e rimuove le EC2 istanze che non hanno attività in esecuzione, eliminando i costi di infrastruttura non necessari dovuti alla capacità inutilizzata. Quando viene rilevata un'istanza inattiva, il processo di ottimizzazione contrassegna l'istanza del contenitore come **ANNULLATA REGISTRAZIONE**, che avvia la sequenza di pulizia che termina in modo sicuro l'istanza sottostante. EC2

Rilevamento delle istanze sottoutilizzate

Analizza la distribuzione delle attività tra le istanze per identificare le opportunità per una migliore allocazione delle risorse. Quando le attività vengono eseguite in modo non ottimale su più istanze, Amazon ECS Managed Instances consolida i carichi di lavoro su un numero inferiore di istanze utilizzate in modo più efficiente, riducendo i costi complessivi e mantenendo le prestazioni. Il processo di ottimizzazione contrassegna le istanze di container sottoutilizzate come **DRAINING**, il che attiva la sostituzione delle attività per spostare i carichi di lavoro su istanze esistenti o nuove e più efficienti. Una volta che tutte le attività sono state migrate in modo sicuro, l'istanza passa allo stato **ANNULLAMENTO DELLA REGISTRAZIONE** e viene ripulita. Questa ottimizzazione si applica

alle istanze che eseguono attività di servizio e garantisce un consolidamento sicuro rispettando i limiti minimi e massimi delle attività del servizio, rispettando il comportamento di start-before-stop implementazione e mantenendo le impostazioni di protezione delle attività durante tutto il processo di esaurimento. Le istanze che eseguono attività autonome non vengono prese in considerazione ai fini dell'ottimizzazione poiché le istanze gestite da ECS non sostituiscono le attività autonome.

Queste ottimizzazioni collaborano per garantire che l'infrastruttura si adatti continuamente alle effettive richieste del carico di lavoro, eliminando automaticamente gli sprechi e migliorando l'utilizzo delle risorse senza influire sulla disponibilità delle applicazioni. Entrambi i meccanismi utilizzano il monitoraggio basato sugli eventi che risponde agli eventi del ciclo di vita delle attività e delle istanze per identificare opportunità di ottimizzazione in tempo reale. Amazon ECS Managed Instances rileva quando l'ultima attività si interrompe su un'istanza di container, indicando una potenziale condizione di inattività per l'ottimizzazione dei costi. Per le istanze sottoutilizzate, qualsiasi interruzione di un'attività o il lancio di una nuova istanza innesca un'analisi per identificare opportunità di consolidamento del carico di lavoro e maggiore efficienza delle risorse.

ScaleInAfter

Entrambe le ottimizzazioni dell'infrastruttura cercano opportunità per terminare le istanze in esecuzione per migliorare l'utilizzo e ridurre i costi. Puoi controllare la tempistica di queste azioni utilizzando la `ScaleInAfter` configurazione nelle impostazioni del capacity provider di Amazon ECS Managed Instances, che si applica sia alle istanze inattive che a quelle sottoutilizzate. `ScaleInAfter` consente di specificare il ritardo, in secondi, tra l'inattività o il sottoutilizzo dell'istanza e il momento in cui Amazon ECS Managed Instances inizia a ottimizzare l'infrastruttura. È possibile impostare il ritardo tra 0 e 3600 secondi. È inoltre possibile specificare -1 per disabilitare l'ottimizzazione dell'infrastruttura.

Istanze inattive

- ECS attende la durata specificata dopo l'interruzione dell'ultima attività prima di annullare la registrazione dell'istanza
- Se una nuova attività inizia durante il periodo di attesa, l'istanza non viene più considerata inattiva e la terminazione viene annullata

Istanze sottoutilizzate

- ECS attende la durata specificata dopo un evento di interruzione dell'attività che causa il sottoutilizzo dell'istanza prima di esaurire l'istanza
- Se viene avviata una nuova attività o un'attività esistente viene interrotta su una particolare istanza durante il periodo di attesa, il timer viene reimpostato dall'ultima interruzione dell'attività

o dall'ora di creazione di una nuova attività e Amazon ECS Managed Instances valuta nuovamente le inefficienze e interviene, se necessario, dopo la scadenza del nuovo periodo di attesa.

Questa configurazione è opzionale. Se non specificato, ECS Managed Instances determina automaticamente la tempistica ottimale in base alla configurazione predefinita di ECS Managed Instances.

Esegui la migrazione AWS Fargate da istanze gestite di Amazon ECS

È possibile eseguire la migrazione dei carichi di lavoro esistenti da Fargate alle Istanze gestite da Amazon ECS. Questa migrazione ti consente di accedere all'intera gamma di tipi di EC2 istanze Amazon, prenotazioni di capacità e funzionalità avanzate mantenendo al contempo l'infrastruttura AWS gestita.

Considerazioni sulla migrazione

Tenere a mente le considerazioni seguenti durante la migrazione da Fargate alle Istanze gestite da Amazon ECS:

Compatibilità delle attività

Le definizioni delle attività esistenti configurate per Fargate sono per lo più compatibili con Amazon ECS Managed Instances. Per ulteriori informazioni sulle differenze nelle definizioni delle attività, consulta [Differenze nella definizione dell'attività di Amazon ECS per istanze gestite da Amazon ECS](#)

Modifiche al modello di sicurezza

Per impostazione predefinita, le Istanze gestite da Amazon ECS consentono di eseguire più attività per istanza. Prendere in considerazione l'attivazione della modalità a task singolo se i carichi di lavoro richiedono un isolamento più forte.

Ciclo di vita dell'istanza

Le Istanze gestite da Amazon ECS hanno una durata massima di 14 giorni. Pianificare la sostituzione delle attività e utilizzare i servizi Amazon ECS per la gestione automatica delle attività.

Modifiche ai prezzi

Con le Istanze gestite da Amazon ECS, paghi l'intera istanza più una commissione di gestione, anziché le risorse per attività come con Fargate.

Finestre di manutenzione

Configura le finestre di manutenzione utilizzando le finestre EC2 degli eventi di Amazon per controllare quando le istanze gestite di Amazon ECS vengono sostituite per l'applicazione di patch.

Prerequisiti

Prima di eseguire la migrazione alle Istanze gestite da Amazon ECS, assicurati di avere:

- Processi Fargate che utilizzano la versione della piattaforma 1.4.0 o successiva
- Si dispone dei ruoli IAM richiesti per le Istanze gestite da Amazon ECS. Questo include:
 - Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

- Comprensione delle differenze del modello di sicurezza tra Fargate e le Istanze gestite da Amazon ECS

Important

Le Istanze gestite da Amazon ECS utilizzano un modello di sicurezza diverso da Fargate. Per impostazione predefinita, possono essere eseguite più attività sulla stessa istanza, il che potrebbe esporre le attività alle vulnerabilità di altre attività. Verifica i requisiti di sicurezza prima della migrazione.

Passaggio 1: aggiornare un cluster per usare le Istanze gestite da Amazon ECS

Creazione di un provider di capacità Quando si crea un provider di capacità con le Istanze gestite da Amazon ECS, questo diventa disponibile solo all'interno del cluster specificato.

Per ulteriori informazioni, consulta [Creazione di un provider di capacità per le istanze gestite da Amazon ECS](#).

Passaggio 2: aggiornare la definizione dell'attività per disporre della funzionalità delle Istanze gestite da Amazon ECS

Aggiornare la definizione del task in modo che abbia le `requiresCapabilities` per le Istanze gestite da Amazon ECS.

Per ulteriori informazioni, consulta [Aggiornamento di una definizione di attività Amazon ECS attraverso la console](#).

Passaggio 3: aggiornare il servizio per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS

Aggiornare il servizio Amazon ECS esistente per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS.

Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS per utilizzare un provider di capacità](#).

Passaggio 4: eseguire la migrazione delle attività autonome

Per le attività autonome, specificare il provider di capacità delle Istanze gestite da Amazon ECS durante l'esecuzione dell'attività.

Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).

Esegui la migrazione da Amazon EC2 a Amazon ECS Managed Instances

Esegui la migrazione dei carichi di lavoro esistenti da Amazon a EC2 Amazon ECS Managed Instances. Questa migrazione ti consente di accedere all'intera gamma di tipi di EC2 istanze Amazon, prenotazioni di capacità e funzionalità avanzate mantenendo al contempo l'infrastruttura AWS gestita.

Considerazioni sulla migrazione

Tieni a mente le seguenti considerazioni durante la migrazione da Amazon a EC2 Amazon ECS Managed Instances:

Compatibilità delle attività

Le definizioni di attività esistenti configurate per Amazon EC2 sono per lo più compatibili con Amazon ECS Managed Instances. Per un elenco delle differenze nella definizione delle attività, consulta [Differenze nella definizione dell'attività di Amazon ECS per istanze gestite da Amazon ECS](#)

Modifiche al modello di sicurezza

Per impostazione predefinita, le Istanze gestite da Amazon ECS consentono di eseguire più attività per istanza. Prendere in considerazione l'attivazione della modalità a task singolo se i carichi di lavoro richiedono un isolamento più forte.

Ciclo di vita dell'istanza

Le Istanze gestite da Amazon ECS hanno una durata massima di 14 giorni. Pianificare la sostituzione delle attività e utilizzare i servizi Amazon ECS per la gestione automatica delle attività.

Modifiche ai prezzi

Con Amazon ECS Managed Instances, paghi l'intera istanza più una commissione di gestione, con la gestione del sovraccarico di AWS gestione dell'infrastruttura.

Finestre di manutenzione

Configura le finestre di manutenzione utilizzando le finestre EC2 degli eventi di Amazon per controllare quando le istanze gestite di Amazon ECS vengono sostituite per l'applicazione di patch.

Prerequisiti

Prima di eseguire la migrazione alle Istanze gestite da Amazon ECS, assicurarsi di avere:

- Si dispone dei ruoli IAM richiesti per le Istanze gestite da Amazon ECS. Questo include:
 - Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

- Comprensione delle differenze dei modelli di sicurezza tra Amazon EC2 e Amazon ECS Managed Instances

Passaggio 1: aggiornare un cluster per usare le Istanze gestite da Amazon ECS

Creazione di un provider di capacità Quando si crea un provider di capacità con le Istanze gestite da Amazon ECS, questo diventa disponibile solo all'interno del cluster specificato.

Per ulteriori informazioni, consulta [Creazione di un provider di capacità per le istanze gestite da Amazon ECS](#).

Passaggio 2: aggiornare la definizione dell'attività per disporre della funzionalità delle Istanze gestite da Amazon ECS

Aggiornare la definizione del task in modo che abbia le `requiresCapabilities` per le Istanze gestite da Amazon ECS.

Per ulteriori informazioni, consulta [Aggiornamento di una definizione di attività Amazon ECS attraverso la console](#).

Passaggio 3: aggiornare il servizio per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS

Aggiornare il servizio Amazon ECS esistente per utilizzare il provider di capacità delle Istanze gestite da Amazon ECS.

Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS per utilizzare un provider di capacità](#).

Passaggio 4: eseguire la migrazione delle attività autonome

Per le attività autonome, specificare il provider di capacità delle Istanze gestite da Amazon ECS durante l'esecuzione dell'attività.

Per ulteriori informazioni, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).

Progettazione per 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 dimensionare 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 le attività e i servizi con 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 attività 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'altra attività. Configura le definizioni delle attività per Fargate impostando il parametro di definizione delle attività di `requiresCompatibilities` su FARGATE. Per ulteriori informazioni, consulta [Capacity](#).

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 si applicano a tutte le piattaforme di Fargate.

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

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

Procedure guidate

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

- [Ulteriori informazioni su come creare un'attività di Amazon ECS Linux per Fargate](#)
- [Ulteriori informazioni su come creare un'attività di Amazon ECS Windows per Fargate](#)

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

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

fornitori di capacità

Sono disponibili i seguenti provider di capacità:

- Fargate
- Fargate Spot: esegui le attività di Amazon ECS con tolleranza alle interruzioni a una tariffa scontata rispetto al prezzo di AWS Fargate. 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

Le attività di Fargate non supportano tutti i parametri di definizione delle attività 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

Puoi scegliere di configurare il servizio Amazon ECS su AWS Fargate per l'utilizzo di Elastic Load Balancing per l'implementazione uniforme del traffico fra i processi del tuo servizio.

I servizi di Amazon ECS su AWS Fargate supportano i tipi di bilanciatori del carico Application Load Balancer, Network Load Balancer e Gateway Load Balancer. Gli Application Load Balancer vengono utilizzati per instradare il traffico HTTP/HTTPS (o di 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](#).

L'utilizzo di un Network Load Balancer per instradare il traffico UDP alle attività Amazon ECS su AWS Fargate è supportato solo se utilizzi la piattaforma versione 1.4 o successiva.

Parametri di utilizzo

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.

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 sulle quote di servizio di AWS Fargate, consulta [Amazon ECS endpoints and quotas](#) in Riferimenti generali di Amazon Web Services.

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

Considerazioni sulla sicurezza di Amazon ECS su quando utilizzare 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 tali carichi di lavoro containerizzati non condividono le interfacce di rete, lo spazio di archiviazione temporaneo, la CPU o la memoria di Fargate con altre attività. Per ulteriori informazioni, consulta [Panoramica sulla sicurezza di AWS Fargate](#).

Best practice sulla 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 crittografare lo storage temporaneo con una delle chiavi gestite dal cliente o con chiavi gestite dal cliente. AWS KMS Le attività ospitate su Fargate che utilizzano la versione della piattaforma 1.4.0 o successiva ricevono almeno 20 GiB di spazio di archiviazione temporaneo. Per ulteriori informazioni, consulta l'argomento relativo alla [chiave gestita dal cliente \(CMK\)](#). 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 le attività avviate a partire dal 28 maggio 2020 (compreso), lo spazio di archiviazione temporaneo viene crittografato con un algoritmo di crittografia AES-256 utilizzando una chiave di crittografia gestita da Fargate.

Per ulteriori informazioni, [Storage options for Amazon ECS tasks](#).

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

Il comando seguente avvierà un'attività sulla versione della piattaforma 1.4 di Fargate. Dal momento che viene avviata nell'ambito del cluster, l'attività utilizza 20 GiB di spazio di archiviazione temporaneo crittografato in automatico.

```
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 di 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 seguente video 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 relative 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 avvio 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](#).

Specifica la versione della piattaforma quando esegui un'attività o implementi un servizio.

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

- Puoi specificare un numero di versione in particolare, ad esempio 1.4.0 o LATEST.

La versione della piattaforma Linux PIÙ RECENTE è 1.4.0.

La versione della piattaforma Windows PIÙ RECENTE 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 allo scopo di eseguire attività sulla versione 1.4.0 della piattaforma Linux, aggiorna il servizio e specifica 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 della versione della piattaforma. Ciò garantisce che le attività siano sempre eseguite su un'infrastruttura protetta e aggiornata.
- 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 il processo funzioni correttamente sulla versione 1.4.0 della piattaforma prima di eseguire la migrazione delle 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 su Fargate ricevono un'unica interfaccia di rete elastica (denominata interfaccia di rete elastica (ENI) dell'attività) e tutto il traffico di rete passa attraverso tale ENI all'interno del VPC. Questo traffico viene registrato nei log di flusso VPC. Per ulteriori informazioni, consulta [Opzioni di rete di attività di Amazon ECS per Fargate](#).
- Se utilizzi endpoint VPC dell'interfaccia, prendi in considerazione le seguenti informazioni.

- 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 del gateway Amazon S3
- Quando la definizione delle attività fa riferimento a segreti di Secrets Manager per recuperare dati sensibili per i container, è necessario creare gli endpoint VPC dell'interfaccia per Secrets Manager. Per ulteriori informazioni, consulta [Utilizzo di Secrets Manager con endpoint VPC](#) nella Guida per l'utente di Gestione dei segreti AWS .
- Quando si utilizza una definizione di attività che fa riferimento ai parametri dell'archivio parametri di Systems Manager per recuperare dati sensibili per i container, 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 nell'interfaccia di rete elastica (ENI) associato all'attività richiede delle regole del gruppo di sicurezza per consentire il traffico tra l'attività e gli endpoint VPC.

Log 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 [Trasferimento di dati sensibili a un container Amazon ECS](#).

- Specifica le variabili di ambiente in blocco utilizzando il parametro di definizione del container `environmentFiles`. Per ulteriori informazioni, consulta [Passare una singola variabile di ambiente a un container 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 maggiori informazioni, consulta la sezione [Opzioni di networking delle attività di Amazon ECS per 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 [Endpoint metadati delle attività 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 temporanea delle attività di Fargate per Amazon ECS](#) e [Opzioni di archiviazione 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 [Usare 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 temporanea delle attività di 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 EC2](#). Per ulteriori informazioni sul networking per Fargate, consultare [Opzioni di rete di attività di Amazon ECS per 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 container di 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 [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#) e [Endpoint metadati delle attività Amazon ECS versione 4 per le 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 informazioni, consulta [Messaggi di errore delle attività di Amazon ECS interrotte](#)
- Basato su Amazon Linux 2.

AWS Deprecazione della versione della piattaforma Fargate Linux

Important

Il 2 marzo 2026, AWS inizierà l'aggiornamento forzato di tutti i servizi Amazon ECS che eseguono la versione 1.3.0 della piattaforma Fargate che non sono migrati alla versione 1.4.0 della piattaforma. Ciò potrebbe influire potenzialmente sui tuoi carichi di lavoro. A partire dal 16 marzo 2026, ritireremo la versione 1.3.0. Da quel momento, non sarà possibile avviare nuove attività o creare nuovi servizi configurati con la versione 1.3.0 della piattaforma, ma le attività esistenti continueranno a essere eseguite. Il 31 marzo 2026, interromperemo tutte le attività in esecuzione rimanenti configurate con la versione 1.3.0 della piattaforma.

Per informazioni su come eseguire la migrazione alla versione 1.4 della piattaforma, consulta [Migrazione alla versione 1.4.0 della piattaforma Linux su Amazon ECS](#).

La tabella seguente elenca le versioni della piattaforma Linux che AWS Fargate ha dichiarato 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.

Per informazioni su come eseguire la migrazione alla versione della piattaforma più recente, 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 nuove attività 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
1.3.0	2 marzo 2026	16 marzo 2026

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

Log delle modifiche delle versioni obsolete di Fargate Linux

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, consultare [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 Gestione dei segreti AWS segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container 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 [Trasferimento di dati sensibili a un container 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 `awslogs`. 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 container di 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.

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 Gestione dei segreti AWS. 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 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 dell'integrità](#).
- Aggiunto il supporto per il rilevamento servizi di Amazon ECS. Per ulteriori informazioni, consulta [Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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.

Log 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

Considerazioni relative ai container Windows su Fargate per Amazon ECS

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

Per eseguire attività su container Linux e Windows, devi creare definizioni delle 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 [Utilizzo di montaggio vincolato con Amazon ECS](#).

- I parametri della CPU e della memoria a livello di attività vengono ignorati per i container Windows. Ti consigliamo di specificare risorse a livello di container per i container Windows.
- memoria per attività

- mermoryReservation sui container
- Riavvia le policy sui container
- Impossibile aggiornare la platformFamily per un servizio.

Comportamento dell'estrazione delle immagini dei container Linux sui container Fargate per Amazon ECS

Ogni attività di Fargate viene eseguita su una propria istanza monouso con tenant singolo. Quando si eseguono container Linux su Fargate, le immagini o i livelli di immagini dei container non vengono memorizzati nella cache dell'istanza. Pertanto, per ogni immagine del container definita nell'attività, l'intera immagine del container deve essere estratta dal relativo registro per ogni attività di Fargate. Il tempo necessario per estrarre le immagini è direttamente correlato al tempo impiegato per avviare un'attività di Fargate.

Prendi in considerazione le seguenti informazioni per ottimizzare il tempo di estrazione dell'immagine.

Prossimità dell'immagine del container

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. Consigliamo di archiviare l'immagine del container nella stessa Regione in cui verrà eseguita l'attività. Se memorizzi l'immagine del container in Amazon ECR, utilizza un endpoint dell'interfaccia VPC per ridurre ulteriormente il tempo di estrazione dell'immagine. Per ulteriori informazioni, consulta [Amazon ECR interface VPC endpoints \(AWS PrivateLink\)](#) nella Guida per l'utente di Amazon ECR.

Riduzione delle dimensioni dell'immagine del container

Le dimensioni di un'immagine del container influiscono direttamente sul tempo di download. La riduzione delle dimensioni o del numero di livelli dell'immagine del container può ridurre il tempo necessario per il download di un'immagine. Le immagini di base leggere (come l'immagine minima del container Amazon Linux 2023) possono essere significativamente più piccole di quelle basate sulle immagini 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 dell'immagine del container vengono spesso compressi quando si inseriscono in un registro di immagini del container. La compressione del livello dell'immagine del container

riduce la quantità di dati che devono essere trasferiti attraverso la rete e archiviati nel registro di immagini del container. Dopo che un livello dell'immagine del container è stato scaricato su un'istanza dal runtime di quest'ultimo, 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](#).

Caricamento lento delle immagini del container

Per immagini del container di grandi dimensioni (> 250 MB), potrebbe essere utile caricare lentamente l'immagine del container anziché scaricarla tutta. Su Fargate, puoi utilizzare Seekable OCI (SOCI) per caricare lentamente un'immagine del container da un registro di immagini del container. Per ulteriori informazioni, consulta [soci-snapshotter](#) on GitHub e [Lazy loading](#) container images using Seekable OCI (SOCI).

Comportamento dell'estrazione delle immagini dei container Windows sui container Fargate 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 al numero di patch aggiornate ogni Patch Tuesday. KB/Build 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 WINDOWS_SERVER_2019_FULL le seguenti immagini del container sono state memorizzate 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 `WINDOWS_SERVER_2022_FULL` le seguenti immagini del container sono state memorizzate nella cache:

- `mcr.microsoft.com/windows/servercore:ltsc2022`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2402`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2340`

Archiviazione temporanea delle attività di 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 [Utilizzo di montaggio vincolato con Amazon ECS](#).

Chiavi gestite dal cliente per l'archiviazione temporanea di AWS Fargate 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 dell'archiviazione autogestita ai revisori della conformità. Sebbene Fargate utilizzi la crittografia dell'archiviazione temporanea gestita da Fargate per impostazione predefinita, i clienti possono anche utilizzare le proprie chiavi autogestite per crittografare i dati sensibili, come le informazioni finanziarie o relative alla salute.

Puoi importare le tue chiavi o crearle in AWS KMS. Queste chiavi autogestite vengono archiviate ed eseguono azioni standard 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 dal cliente. Ti viene addebitato solo il prezzo standard per l'utilizzo AWS KMS delle chiavi per l'archiviazione e le richieste API.

Argomenti

- [Creazione di una chiave di crittografia per l'archiviazione temporanea di Fargate per Amazon ECS](#)
- [Gestione delle AWS KMS chiavi per lo storage temporaneo Fargate per Amazon ECS](#)

Creazione di una chiave di crittografia per l'archiviazione temporanea di Fargate per Amazon ECS

Crea una chiave gestita dal cliente per crittografare i dati nell'archiviazione temporanea di Fargate.

Note

La crittografia dell'archiviazione temporanea di Fargate con chiavi gestite dal cliente non è disponibile per i cluster di attività di Windows.

La crittografia dell'archiviazione temporanea di Fargate con chiavi gestite dal cliente non è disponibile sulle `platformVersions` precedenti alla `1.4.0`.

Fargate riserva spazio sull'archiviazione temporanea utilizzata solo da Fargate e non viene addebitato alcun costo per lo spazio. L'allocazione potrebbe differire dalle attività chiave non gestite dal cliente, ma lo spazio totale rimane invariato. 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 sviluppatori di AWS Key Management Service](#)
3. Durante la creazione della AWS KMS chiave, assicuratevi di fornire al servizio Fargate le autorizzazioni AWS KMS operative pertinenti nelle politiche chiave. Per utilizzare la chiave gestita dal cliente con le risorse del cluster di Amazon ECS, le seguenti operazioni API devono essere concesse nella policy della chiave gestita:
 - `kms:GenerateDataKeyWithoutPlainText`- Chiama `GenerateDataKeyWithoutPlainText` per generare una chiave dati crittografata dalla chiave fornita AWS KMS .
 - `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 delle concessioni](#), consulta la [Guida per gli sviluppatori di 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 effimeri.
- Configurare un principale ritirato per consentire al servizio di RetireGrant.
- kms:DescribeKey: fornisce i dettagli della chiave gestita dal cliente per consentire ad Amazon ECS di convalidarla se è simmetrica e abilitata.

L'esempio seguente mostra una politica AWS KMS chiave da applicare alla chiave di destinazione per la crittografia. Per usare gli esempi di dichiarazioni della policy, sostituire *user input placeholders* con le proprie 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 di Fargate utilizzano le chiavi contestuali di crittografia `aws:ecs:clusterAccount` e `aws:ecs:clusterName` per le operazioni di crittografia con la chiave. I clienti devono aggiungere queste autorizzazioni per limitare l'accesso a un cluster di account specifico. and/or Utilizza 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 crei o aggiorni un cluster, puoi utilizzare una chiave di condizione `fargateEphemeralStorageKmsKeyId`. Tale chiave di condizione consente ai clienti di avere un controllo più granulare delle policy IAM. Gli aggiornamenti alla configurazione `fargateEphemeralStorageKmsKeyId` 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 uno specifico set di chiavi approvate AWS KMS .

JSON

```
{
  "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-EXAMPLE11111"
        }
      }
    }
  ]
}
```

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

JSON

```
{
  "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 sulle chiavi, consulta [Condition keys for AWS KMS](#) nella [Guida per gli sviluppatori di AWS KMS](#).

Console di gestione AWS

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
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
}

```

CloudFormation

Di seguito è riportato un modello di esempio per la creazione di un cluster e la configurazione dello storage temporaneo Fargate utilizzando (sostituisci 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

Puoi abilitare la rotazione automatica delle chiavi o ruotarle manualmente. La rotazione automatica della chiave ruota 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 della chiave è facoltativa e può essere abilitata o disabilitata in qualsiasi momento.

Disattivazione o revoca delle chiavi AWS KMS

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'attività ha esito negativo poiché non può accedere alla chiave. Dovresti impostare un CloudWatch allarme o qualcosa di simile per assicurarti 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 deve essere eseguita solo se hai la certezza che la chiave eliminata non sarà più necessaria. Le nuove attività che tentano di utilizzare la chiave eliminata falliranno perché non potranno 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-EXAMPLE2222",
  "eventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLE3333",
  "readOnly": false,
  "resources": [
    {
      "accountId": "AWS Internal",
      "type": "AWS::KMS::Key",
      "ARN": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE1111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": true,
  "recipientAccountId": "account-id",
  "sharedEventID": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaa",
  "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-EXAMPLE1111"
  },
  "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-EXAMPLEaaaaa",
"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 piattaforma supportate per introdurre una nuova revisione contenente aggiornamenti al software runtime di Fargate e alle dipendenze sottostanti come il sistema operativo e il runtime del container. 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 di 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 del servizio sono implementate nell'ambito 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 devi effettuare alcuna operazione in risposta al ritiro delle attività di servizio poiché il pianificatore 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 ritirarle, a meno che non si desideri sostituirle prima dell'esecuzione. AWS Quando il pianificatore di Amazon ECS arresta le attività, quest'ultimo utilizza `maximumPercent` e avvia nuove attività nel tentativo di mantenere il numero di servizi desiderato. Segui le best practice per ridurre al minimo l'impatto del ritiro delle attività. Il valore predefinito `maximumPercent` per un servizio che utilizza il pianificatore di servizi 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 valore `maximumPercent` al 100%, Amazon ECS arresta 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 arrestare le attività in esecuzione e avviarne una 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.

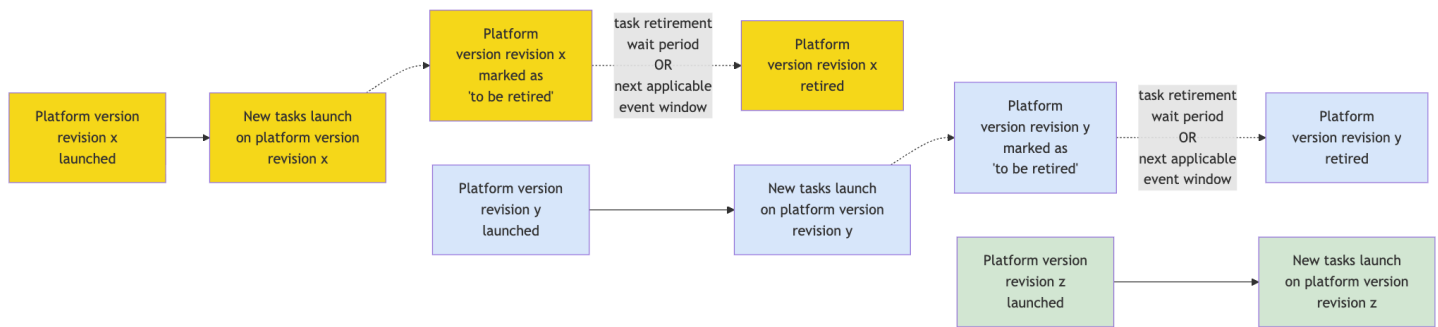
Quando un'attività viene interrotta in uno degli scenari seguenti, puoi eseguire il comando `describe-tasks`. Il `stoppedReason` nella risposta è `ECS is performing maintenance on the underlying infrastructure hosting the task`.

La manutenzione delle attività viene applicata quando una revisione della versione della piattaforma deve essere sostituita con una nuova. In caso di problemi con un host Fargate sottostante, Amazon ECS sostituisce l'host senza un avviso di ritiro delle attività.

Panoramica sull'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.

La figura seguente mostra il ciclo di vita di una revisione della versione della piattaforma di Fargate, dall'avvio di una nuova revisione al ritiro della versione della piattaforma.



Le seguenti informazioni forniscono dettagli.

- Dopo l'avvio di una nuova revisione della versione della piattaforma, tutte le nuove attività vengono pianificate in base a tale 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 a quella nuova.
- 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 dell'avvio.

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 per ciascuna attività del servizio e attività autonome in ogni 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 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$ aws ecs update-
service -service service_name \\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(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": "arn:aws:ecs:eu-west-1:111222333444:task/
examplecluster/00805ce1d81940b5a37398e5a2c23333"
      },
      {
        "entityValue": "arn:aws:ecs:eu-west-1:111222333444:task/
examplecluster/00805ce1d81940b5a37398e5a2c25555"
      }
    ]
  }
}

```

- Email

Viene inviata un'e-mail all'indirizzo di posta elettronica registrato per l' Account AWS ID.

Per informazioni su come preparare il ritiro delle attività, consulta [Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS](#).

Posso disabilitare il 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. In questo modo, puoi concentrarti sulle applicazioni, piuttosto che 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 AWS servizi?

AWS invia una notifica di ritiro dell'attività al Health Dashboard e al contatto di posta elettronica principale su Account 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 [Solution overview: Capturing task retirement notifications](#).

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 estendere il periodo di attesa a 14 giorni. Per ulteriori informazioni, consulta [Fase 2: registrazione delle notifiche di ritiro delle attività per avvisare i team ed effettuare azioni](#).

A partire dal 18/12/2025, Amazon ECS ti consente di configurare [finestre di eventi EC2 Amazon per le tue attività](#) a Fargate. Se hai bisogno di un controllo preciso sulla tempistica esatta del ritiro delle attività, ad esempio programmandole durante i fine settimana per evitare interruzioni durante l'orario lavorativo, puoi configurare le finestre EC2 degli eventi di Amazon per le tue attività, servizi o cluster, vedi. [Passaggio 1: imposta il tempo di attesa dell'attività o utilizza le finestre EC2 degli eventi di Amazon](#) Tieni presente che la modifica di questa configurazione si applica ai pensionamenti che verranno programmati in futuro. I ritiri attualmente pianificati non sono influenzati. Inoltre, quando configuri una finestra di EC2 eventi Amazon per le tue attività su Fargate, questa ha la precedenza sulla configurazione del tempo di attesa per il ritiro dell'attività. Se avete ulteriori dubbi, contattateci. 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 tali attività prima dell' AWS esecuzione. Quando il pianificatore di Amazon ECS arresta le attività, quest'ultimo utilizza la percentuale minima di integrità e avvia nuove attività nel tentativo di mantenere il numero di servizi desiderato. Per ridurre al minimo l'impatto del ritiro delle attività di Fargate, i carichi di lavoro devono essere implementati 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` Il valore predefinito per la percentuale minima di integrità di un servizio è il 100%. Pertanto, quando Fargate inizia a ritirare le attività, Amazon ECS pianifica una nuova attività e attende che sia in esecuzione, prima di ritirarne una vecchia. Le attività del servizio vengono sostituite regolarmente come parte del ritiro delle attività allo stesso modo quando dimensioni il servizio, implementi le modifiche alla configurazione o le revisioni di definizioni delle attività. Per preparare il processo di ritiro delle attività, consulta [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.

Risoluzione dei problemi relativi alla disponibilità del servizio durante il ritiro delle attività

Se Amazon ECS non è in grado di avviare un'attività sostitutiva durante il ritiro delle attività, la disponibilità del servizio potrebbe risentirne. Ciò può verificarsi a causa di una configurazione errata del cliente, ad esempio:

- Ruoli IAM mancanti o configurati in modo non corretto
- Capacità insufficiente nelle sottoreti di destinazione
- Configurazioni errate del gruppo di sicurezza
- Errori nelle definizioni delle attività

Quando Amazon ECS non è in grado di avviare attività sostitutive, le attività ritirate vengono arrestate senza sostituzioni, riducendo la capacità disponibile del servizio e causando potenzialmente interruzioni del servizio. Monitora il numero di attività del tuo servizio e i `CloudWatch` parametri di Amazon per assicurarti che le attività sostitutive vengano lanciate con successo durante gli eventi di pensionamento.

Preparati per il ritiro delle attività di AWS Fargate su Amazon ECS

Per preparare il ritiro delle attività, esegui le seguenti operazioni:

1. Imposta il periodo di attesa per il ritiro dell'attività o utilizza le finestre EC2 degli eventi di Amazon.
2. Registra le notifiche di ritiro delle attività per avvisare i membri del team.
3. Puoi assicurarti che tutte le attività dei tuoi servizi vengano eseguite sulla versione più recente della piattaforma aggiornando il servizio con l'opzione di implementazione forzata. Questa fase è facoltativa.

Passaggio 1: imposta il tempo di attesa dell'attività o utilizza le finestre EC2 degli eventi di Amazon

Sono disponibili due opzioni di impostazione dell'account per configurare l'ora in cui Fargate avvia il ritiro delle attività: `e. fargateTaskRetirementWaitPeriod` `fargateEventWindows`

Utilizzo delle impostazioni dell'account `fargateTaskRetirement WaitPeriod`

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Il periodo di attesa predefinito è di 7 giorni. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (0). Se hai bisogno di più tempo, configura l'opzione 7, o 14 day.

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. Questa è l'impostazione predefinita.
- 14- AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

Per ulteriori informazioni, consulta [put-account-setting-default](#) e consulta il riferimento [put-account-setting](#) all'API di Amazon Elastic Container Service.

Utilizzo `fargateEventWindows` delle impostazioni dell'account

A partire dal 18/12/2025, Amazon ECS ti consente di configurare [finestre di eventi EC2 Amazon per le tue attività](#) a Fargate. Se hai bisogno di un controllo preciso sulla tempistica esatta del ritiro delle attività, ad esempio programmandole durante i fine settimana per evitare interruzioni durante l'orario lavorativo, puoi configurare le finestre degli EC2 eventi di Amazon per le tue attività, i tuoi servizi o i tuoi cluster.

Quando si utilizzano le finestre degli eventi, Fargate assicura che le attività vengano eseguite per almeno 3 giorni prima di essere ritirate entro la finestra successiva disponibile, a meno che

non vengano interrotte da azioni avviate dall'utente o da eventi critici di salute come il degrado dell'hardware sottostante.

Impostare le impostazioni dell'account `fargateEventWindows` su `enabled`. È possibile utilizzare uno dei seguenti APIs: `put-account-setting-default` o `put-account-setting` come utente `root` o utente amministrativo.

Ogni finestra di EC2 evento Amazon deve essere aperta per almeno 4 ore a settimana e ogni intervallo di tempo deve essere lungo almeno 2 ore. Per cluster e servizi di grandi dimensioni, consigliamo di configurare finestre di eventi con durate lunghe (8 ore o più) o intervalli di tempo più frequenti che si verificano almeno una volta ogni 3 giorni. Puoi esaminare ulteriormente le considerazioni relative alle finestre degli EC2 eventi di Amazon nella [guida per l'utente](#) per assicurarti che le tue attività AWS Fargate vengano eseguite per almeno 3 giorni prima di essere ritirate, a meno che non vengano interrotte da azioni avviate dall'utente o da eventi critici di salute come il degrado dell'hardware sottostante.

 Important

La sostituzione delle attività all'interno della finestra degli eventi è la soluzione migliore. Se noti che le attività vengono ritirate al di fuori delle finestre degli eventi, valuta la possibilità di aumentare la durata (8 ore o più) o di aumentare la frequenza (almeno una volta ogni 3 giorni).

Per applicare le finestre EC2 degli eventi di Amazon ai ritiri delle attività di Fargate:

- Impostare le impostazioni dell'account `fargateEventWindows` su `enabled`. Puoi utilizzare uno dei seguenti APIs: `put-account-setting-default` o `put-account-setting` come utente `root` o utente amministrativo. Tieni presente che si tratta di un'attivazione unica per l'utilizzo della funzionalità Amazon EC2 Event Windows per le tue attività su Fargate.
- Crea una finestra di EC2 eventi Amazon tramite la console AWS o l'AWS CLI. Per creare una finestra di eventi utilizzando la CLI, utilizza l' `EC2create-instance-event-window` API con intervalli di tempo o espressioni cron. Prendi nota di quanto riportato nella `InstanceEventWindowId` risposta.

```
aws ec2 create-instance-event-window \
    --time-range
    StartWeekDay=monday,StartHour=2,EndWeekDay=wednesday,EndHour=8 \
```

```
--tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1" \
--name myEventWindowName
```

In alternativa, puoi usare le espressioni cron durante la creazione di finestre di EC2 eventi.

```
aws ec2 create-instance-event-window \
--cron-expression "* 21-23 * * 2,3" \
--tag-specifications "ResourceType=instance-event-
window,Tags=[{Key=K1,Value=V1" \
--name myEventWindowName
```

- Puoi quindi associare la finestra dell'evento a servizi specifici, cluster o tutte le attività del tuo account utilizzando l' EC2 associate-instance-event-window API.
- Per le attività di servizio ECS

```
aws ec2 associate-instance-event-window \
--instance-event-window-id iew-0abcdef1234567890 \
--association-target "InstanceTags=[{Key=aws:ecs:serviceArn,Value=your-service-
arn}]"
```

- Per i cluster ECS

```
aws ec2 associate-instance-event-window \
--instance-event-window-id iew-0abcdef1234567890 \
--association-target "InstanceTags=[{Key=aws:ecs:clusterArn,Value=your-cluster-
arn}]"
```

- Per associare una finestra dell'evento a tutte le attività dell'account

```
aws ec2 associate-instance-event-window \
--instance-event-window-id iew-0abcdef1234567890 \
--association-target "InstanceTags=[{Key=aws:ecs:fargateTask,Value=true}]"
```

È possibile utilizzare più di una coppia chiave-valore per associare una finestra dell'evento a più servizi o cluster.

Fargate sceglierà la finestra dell'evento per ogni attività nel seguente ordine:

- Se è presente una finestra di eventi associata al servizio dell'attività, verrà utilizzata. Questo non è applicabile alle attività autonome o non gestite.
- Se è presente una finestra di eventi associata al cluster dell'attività, verrà utilizzata.
- Se è impostata una finestra di eventi per tutte le attività di Fargate, verrà utilizzata.
- L'`fargateTaskRetirementWaitPeriod` impostazione verrà utilizzata se nessuna delle finestre degli eventi corrisponde all'attività.

Configurazione delle finestre degli eventi per la manutenzione delle attività di Fargate

Prendi in considerazione un caso in cui esegui più servizi ECS su Fargate con requisiti di disponibilità diversi. Desiderate un controllo preciso sulla cessazione delle attività. È possibile configurare più finestre di eventi come segue:

- Manutenzione predefinita per tutte le attività Fargate: create una finestra evento per la manutenzione ordinaria durante le ore non di punta (tutti i giorni dalle 12:00 alle 4:00) e associatela a tutte le attività Fargate utilizzando il tag `aws:ecs:fargateTask`
- Manutenzione solo nel fine settimana per il cluster di sviluppo: per un cluster di sviluppo con servizi in grado di tollerare interruzioni nei fine settimana, crea una finestra di 24 ore per il fine settimana (sabato e domenica, tutto il giorno) e associala al cluster utilizzando il tag con l'ARN del `aws:ecs:clusterArn` cluster.
- Intervallo limitato per il servizio mission critical: per un servizio di elaborazione dei pagamenti di importanza critica che richiede un'elevata operatività durante i giorni feriali, limita la manutenzione alle prime ore mattutine del fine settimana (sabato e domenica, dalle 00:00 alle 04:00) e associala al servizio specifico utilizzando il tag con l'ARN del servizio. `aws:ecs:serviceArn`

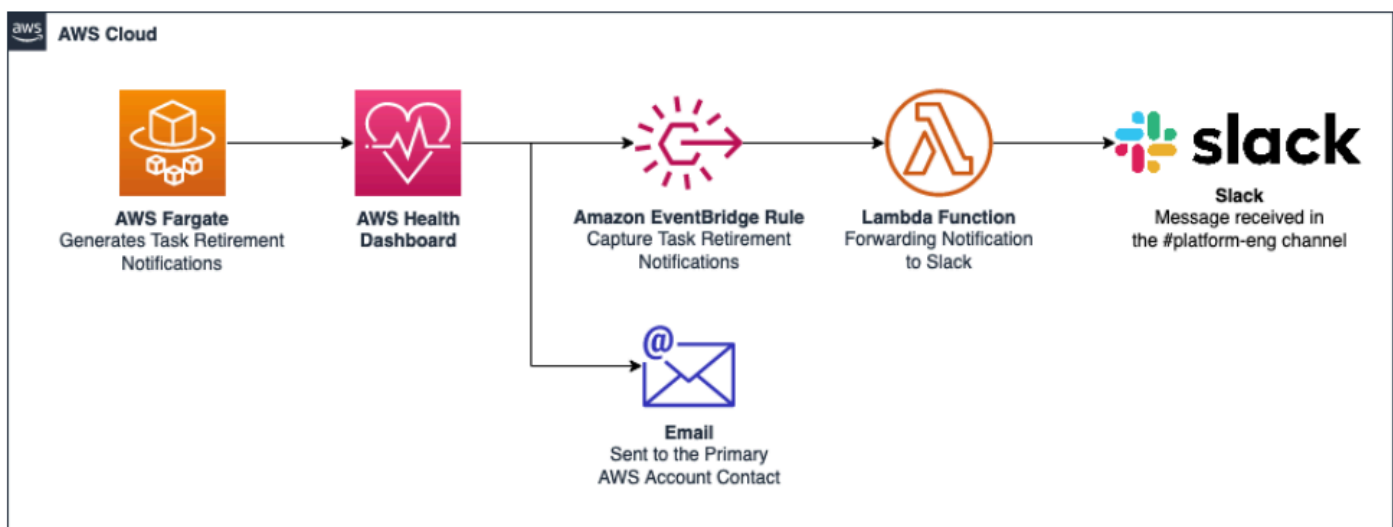
Con questa configurazione, il servizio di pagamento utilizza la sua finestra specifica solo per il fine settimana, i servizi e le attività del cluster di sviluppo utilizzano la finestra di 24 ore del fine settimana e tutte le altre attività di Fargate utilizzano la finestra di manutenzione giornaliera predefinita.

Per ulteriori informazioni, consulta [put-account-setting-defaults](#) e consulta il riferimento [put-account-setting](#) all'API di Amazon Elastic Container Service.

Fase 2: registrazione delle notifiche di ritiro delle attività per avvisare i team ed effettuare 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 in 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 delle attività a un'applicazione di chat, come Slack.

La figura seguente illustra la panoramica della soluzione.



Le seguenti informazioni forniscono dettagli.

- Fargate invia la notifica di ritiro dell'attività a Dashboard AWS Health .
- La AWS Health Dashboard invia posta elettronica al contatto di posta elettronica principale Account AWS su e notifica. EventBridge
- EventBridge ha una regola che registra la notifica di pensionamento.

La regola che cerca eventi con il tipo di dettagli dell'evento: "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 un webhook in ingresso di Slack. Per ulteriori informazioni, consulta [Webhook in ingresso](#).

Per un esempio di codice, vedi [Acquisizione delle notifiche di ritiro delle AWS Fargate attività su Github](#).

Fase 3: controllo della sostituzione delle attività

Non puoi controllare l'ora esatta del ritiro di un'attività, tuttavia puoi definire un tempo di attesa. Se desideri controllare la sostituzione delle attività in base alla tua pianificazione, puoi registrare l'avviso di ritiro delle attività per comprendere innanzitutto la data di ritiro. Quindi, puoi implementare nuovamente il servizio per avviare attività sostitutive e, allo stesso modo, sostituire qualsiasi attività autonoma. Per i servizi che utilizzano l'implementazione progressiva, è possibile aggiornare il servizio utilizzando `update-service` con l'opzione `force-deployment` prima dell'inizio del ritiro.

Nell'esempio `update-service` seguente viene utilizzata l'opzione `force-deployment`.

```
aws ecs update-service --service service_name \  
  --cluster cluster_name \  
  --force-new-deployment
```

Per i servizi che utilizzano la blue/green distribuzione, è necessario creare una nuova distribuzione in AWS CodeDeploy. Per informazioni su come creare l'implementazione, consulta [create-deployment](#) nella Guida di riferimento di 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>)

Nome della regione	Regione
Stati Uniti occidentali (Oregon)	us-west-2
Canada (Centrale)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Messico (centrale)	mx-central-1
Africa (Città del Capo)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacifico (Tokyo)	ap-northeast-1 (solo apne1-az1 , apne1-az2 e apne1-az4)
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Hyderabad)	ap-south-2
Asia Pacifico (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Thailandia)	ap-southeast-7
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacifico (Malesia)	ap-southeast-5
Canada (Centrale)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (solo cnn1-az1 e cnn1-az2)

Nome della regione	Regione
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Milan)	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

Nome della regione	Regione
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)
Stati Uniti occidentali (Oregon)	us-west-2
Africa (Città del Capo)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacifico (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)
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
Europe (Milan)	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

Architetto per la EC2 capacità di Amazon ECS

Usa la EC2 capacità 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 EC2, si consiglia di considerare quali processi devono essere eseguiti insieme e come è possibile ridimensionare ciascun componente.

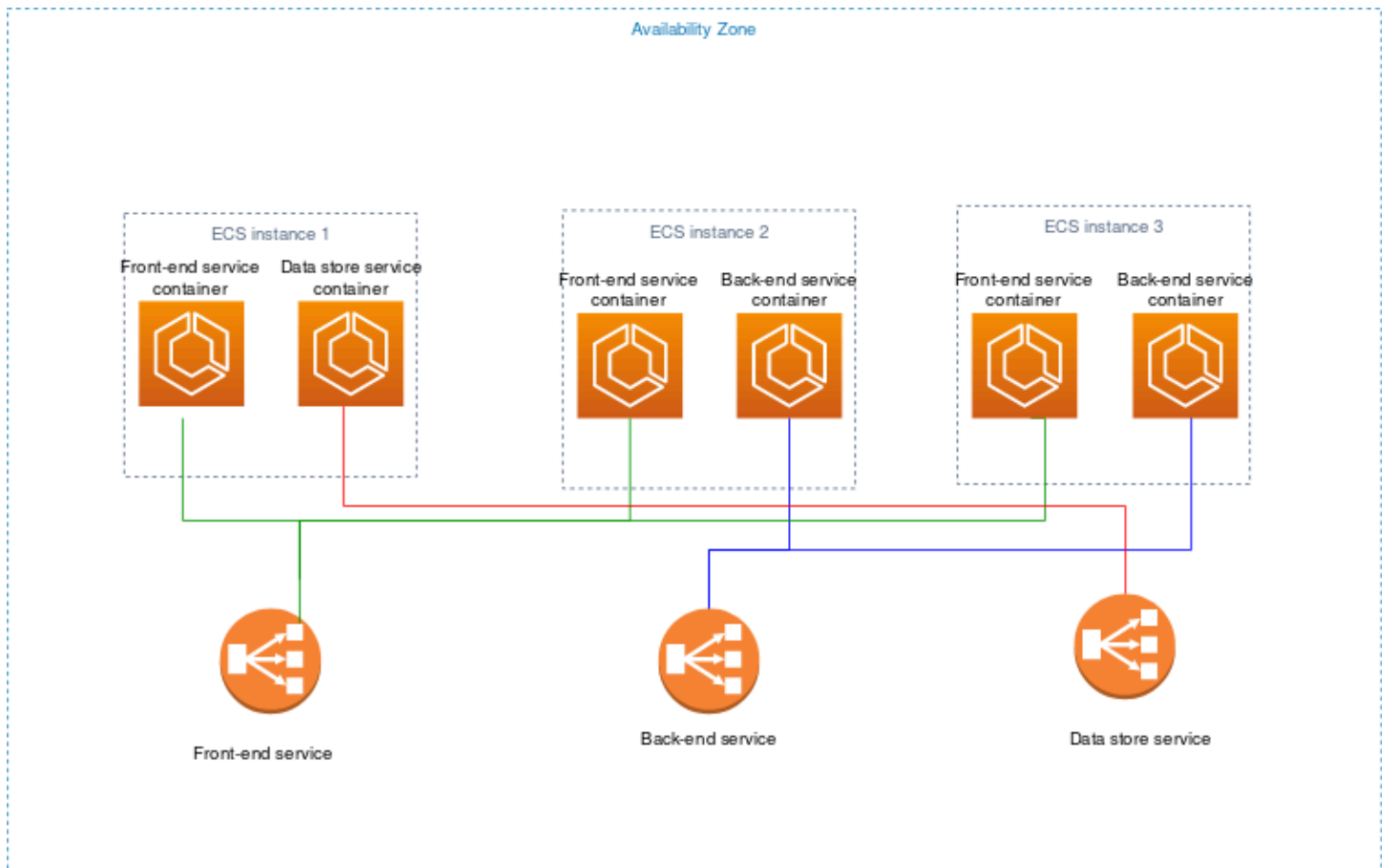
Ad esempio, immagina un'applicazione che comprende i seguenti componenti:

- Un servizio frontend 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. Il seguente esempio di cluster presenta tre istanze di container su cui vengono eseguiti tre container per i servizi frontend, due container per i servizi backend 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 frontend, 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'implementazione di aggiornamenti continui di Amazon ECS](#). Nei servizi, puoi associare i container mediante i bilanciatori del carico 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](#).



Comportamento di estrazione dell'immagine del contenitore per EC2 istanze esterne su Amazon ECS

Il tempo impiegato da un container per l'avvio varia in base alla sua immagine sottostante. Ad esempio, un'immagine più grande (versioni complete di Debian, Ubuntu e Amazon1/2) potrebbe richiedere più tempo per l'avvio per la presenza di più servizi in esecuzione nei container rispetto alle rispettive versioni minori (di Debian, Ubuntu e Amazon) o alle immagini di base più piccole (Alpine).

Quando l'agente Amazon ECS avvia un'attività, estrae l'immagine Docker dal registro remoto e 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 dell'agente `ECS_IMAGE_PULL_BEHAVIOR` 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 ha esito negativo, lo ha anche l'operazione.

Per accelerare l'implementazione, imposta il parametro dell'agente Amazon ECS su uno dei seguenti valori:

- `ECS_IMAGE_PULL_BEHAVIOR: once`

L'immagine viene estratta in remoto solo se non è stata estratta da un'attività precedente nella stessa istanza di container o se l'immagine memorizzata nella cache è stata rimossa dal processo di pulizia dell'immagine automatizzato. 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`

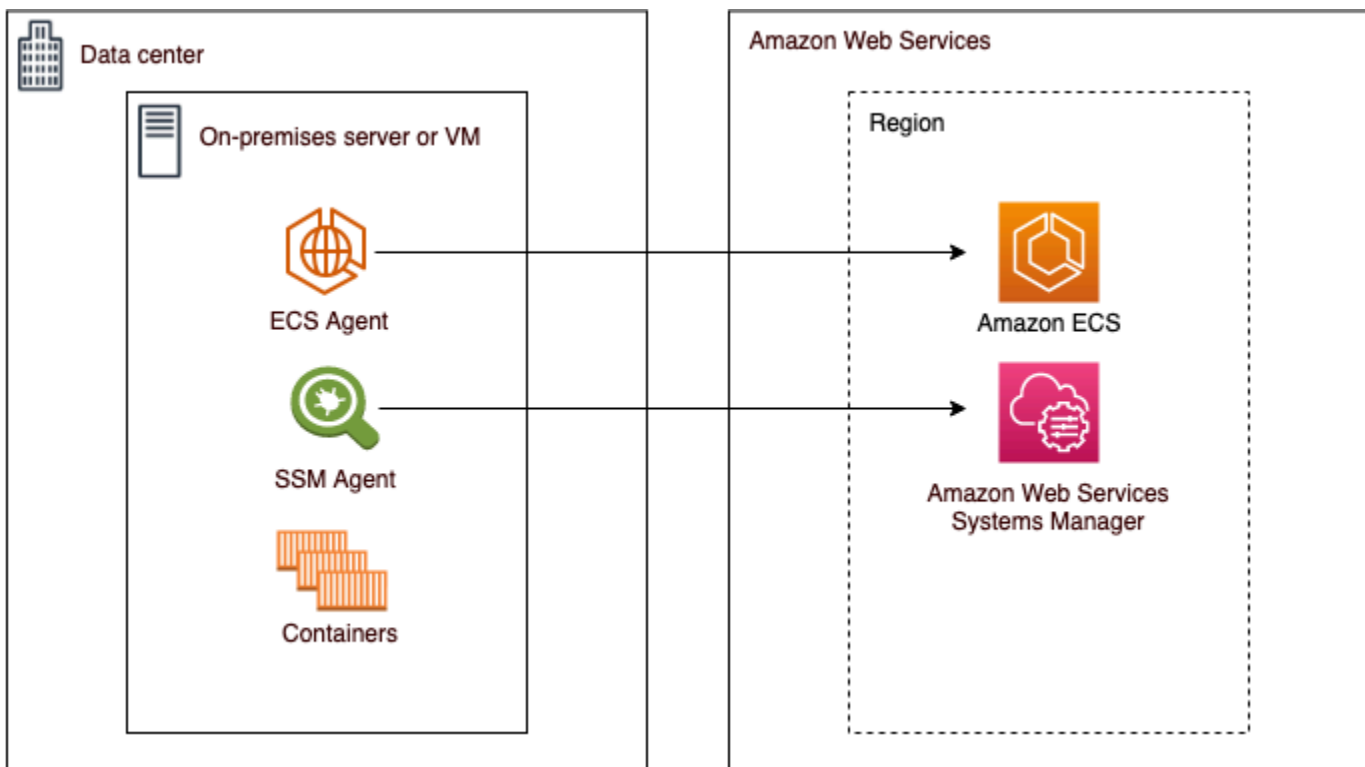
L'immagine viene estratta da remoto se non esiste un'immagine memorizzata nella cache. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. La pulizia automatizzata dell'immagine è disattivata per il container, per avere la certezza che l'immagine memorizzata nella cache non venga rimossa.

L'impostazione del parametro `ECS_IMAGE_PULL_BEHAVIOR` 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 download per il trasferimento dalla rete potrebbe richiedere 10-20 secondi.

Esterno (Amazon ECS Anywhere) per Amazon ECS

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premises 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 on-premises sono installati sia l'agente Amazon ECS che SSM.



Per ulteriori informazioni, consulta [Cluster Amazon ECS per istanze esterne](#).

Modalità Amazon ECS Express

Un servizio Amazon ECS Express Mode riduce la complessità della distribuzione di applicazioni containerizzate fornendo impostazioni predefinite sensibili e automatizzando la configurazione dei servizi di supporto. AWS Invece di gestire i parametri di configurazione su più servizi, un servizio in modalità Express richiede solo tre cose per iniziare:

- Un'immagine del contenitore
- Un ruolo di esecuzione di un'attività
- Un ruolo nell'infrastruttura

La modalità Amazon ECS Express orchestra e configura tutta l'infrastruttura necessaria: un servizio ECS basato su Fargate con un URL accessibile univoco, bilanciamento del carico con SSL/TLS, politiche di scalabilità automatica, monitoraggio e componenti di rete.

Utilizza i servizi in modalità Amazon ECS Express nei seguenti scenari:

- Applicazioni Web e APIs - Applicazioni containerizzate senza stato che servono richieste HTTP
- Prototipazione rapida: implementa e testa rapidamente le applicazioni senza sovraccaricare la configurazione dell'infrastruttura
- Produttività degli sviluppatori: consenti ai team applicativi di implementare le applicazioni in modo indipendente senza conoscenze approfondite AWS
- Efficienza del team di piattaforma: riduci il sovraccarico di manutenzione fornendo funzionalità di implementazione self-service

I servizi Amazon ECS Express Mode supportano applicazioni HTTPS pubbliche o private e scalano automaticamente in base all'utilizzo o al traffico.

Vantaggi dei servizi Amazon ECS Express Mode

I servizi Amazon ECS Express Mode offrono diversi vantaggi chiave per la distribuzione e la gestione di applicazioni containerizzate su AWS.

- Distribuzione semplificata: distribuisce applicazioni containerizzate con impostazioni predefinite pronte per la produzione su più servizi AWS.

- Nessun compromesso sulle funzionalità: tutte le AWS risorse sottostanti rimangono accessibili per la gestione diretta quando è necessario un controllo preciso o funzionalità avanzate.
- Ottimizzazione dei costi: condivide gli Application Load Balancer su più servizi Express Mode utilizzando la stessa configurazione di rete per ridurre i costi.
- Infrastruttura trasparente: tutte le risorse vengono create nel tuo AWS account con visibilità e accessibilità complete tramite la AWS console e. APIs

Prezzi

Non sono previsti costi aggiuntivi per l'utilizzo di un servizio Amazon ECS Express Mode. Paghi solo per le AWS risorse sottostanti create per eseguire l'applicazione, tra cui:

- Risorse di calcolo Fargate
- Application Load Balancer
- CloudWatch registri e metriche
- Costi per il trasferimento dei dati

Disponibilità

La modalità Amazon ECS Express è disponibile in tutte le AWS regioni in cui sono supportati Amazon ECS e Fargate. Puoi creare e gestire i servizi in modalità Express utilizzando uno dei seguenti strumenti:

- La console Amazon ECS
- AWS CLI
- AWS SDKs
- CloudFormation
- Terraform
- AWS Server MCP Labs per Amazon ECS

Creazione di un servizio Amazon ECS Express Mode

Un servizio Amazon ECS Express Mode riduce la complessità della distribuzione di applicazioni containerizzate fornendo impostazioni predefinite sensibili e automatizzando la configurazione dei

servizi di supporto. AWS Invece di gestire i parametri di configurazione su più servizi, un servizio in modalità Express richiede solo un'immagine del contenitore, un ruolo di esecuzione delle attività e un ruolo di infrastruttura per iniziare.

Una volta completata la distribuzione, ti verrà restituito un URL univoco per la tua applicazione. Puoi trovare il servizio che hai appena distribuito in Clusters e, se non hai specificato un cluster, lo troverai nel cluster. Default

Prerequisiti

Prima di creare un servizio in modalità Express, assicurati di disporre di uno dei seguenti:

- Un'immagine del contenitore archiviata in Amazon ECR.
- Oppure un'immagine del contenitore memorizzata in un registro privato. Per utilizzare un registro privato, configura un segreto di Secrets Manager. Per ulteriori informazioni, consulta [Creare un Gestione dei segreti AWS segreto](#) nella Guida Gestione dei segreti AWS per l'utente.

Note

Anche i ruoli IAM sono obbligatori, ma sono trattati separatamente nelle guide [Console](#) e [CLI](#). La console Amazon ECS dispone di flussi automatizzati per la creazione di ruoli.

Procedure guidate

- Per un'esperienza semplificata di prima esecuzione con la console, consulta. [the section called “Crea il tuo primo servizio Express Mode nella console”](#)
- Per creare un servizio Express Mode utilizzando il AWS CLI, vedere. [the section called “Crea il tuo primo servizio Express Mode utilizzando il AWS CLI”](#)
- Per ulteriori informazioni su cosa crea Express Mode e su come funziona, consulta [the section called “Risorse create dai servizi Express Mode”](#).

Personalizzazione

Quando crei un servizio in modalità Express, Amazon ECS seleziona i valori predefiniti per Amazon ECS e altre risorse. AWS Personalizza l'applicazione quando necessario utilizzando i parametri opzionali. Sfrutta un servizio Express Mode e personalizza quanto segue:

- Nome del servizio
- Cluster Amazon ECS
- Impostazioni del contenitore: porta e percorso di controllo dello stato
- Segreti e variabili di ambiente
- Comandi personalizzati
- Ruolo dell'attività: questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi.

Questo ruolo è necessario quando l'applicazione accede ad altri AWS servizi, come Amazon S3.

- Risorse di calcolo: allocazione di CPU e memoria
- Configurazione di rete: sottoreti e gruppo di sicurezza del servizio
- Politiche di dimensionamento automatico delle applicazioni: metriche, valori target e limiti delle attività
- CloudWatch Configurazione dei registri
- Tag delle risorse

Crea il tuo primo servizio Amazon ECS Express Mode nella console

L'esperienza di console per il servizio Express Mode offre un modo semplificato per distribuire un'applicazione containerizzata con una configurazione minima richiesta. Scopri di più sui prerequisiti in [Modalità Amazon ECS Express](#). Per ulteriori informazioni su cosa crea la modalità Express e su come funziona, consulta [the section called “Risorse create dai servizi Express Mode”](#).

Procedura

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nel pannello di navigazione, scegli la modalità Express.
3. In Configuriamo la tua app:
 - a. Specificate l'immagine da usare per l'applicazione. Per l'URI dell'immagine, inserisci l'URI dell'immagine. Per sfogliare le tue immagini Amazon ECR, scegli Browse ECR images, quindi procedi come segue:

- i. Per Archivio privato, scegli l'archivio privato Amazon ECR.
- ii. Per Image, scegli la tua immagine.
- iii. Scegli come identificare l'immagine. Per Seleziona immagine per, scegli una delle seguenti opzioni:
 - AWS consiglia di scegliere Image digest.
 - Per usare il tag, scegli il tag Image, quindi scegli il tag.
- b. Se utilizzi un registro privato, seleziona Registro privato. Quindi, per Secrets Manager ARN o name, inserisci l'ARN di Secrets Manager che hai creato nei prerequisiti.
- c. Per Ruolo di esecuzione delle attività e Ruolo di infrastruttura, scegli i ruoli. Se non hai mai creato questi ruoli in Amazon ECS prima, vedrai l'opzione Crea nuovo ruolo nel menu a discesa. Quando selezioni questa opzione, il ruolo verrà creato automaticamente dopo aver fatto clic su Crea. Per il ruolo di esecuzione delle attività, nei dettagli dell'autorizzazione verrà visualizzato il collegamento per ulteriori dettagli su [Amazon ECSTask ExecutionRolePolicy](#) nella AWS Managed Policy Reference Guide. Per il ruolo Infrastruttura, nei dettagli dell'autorizzazione verrà visualizzato il collegamento per ulteriori dettagli sulla [Amazon ECSInfrastructure RoleforExpressGatewayServices](#) Managed Policy nella AWS Managed Policy Reference Guide.

Se preferisci personalizzare questi ruoli, il pulsante Crea nuovo ruolo ti indirizza alla console IAM dove puoi creare il ruolo, visualizzare e modificare le autorizzazioni associate al ruolo. Quando torni alla console Amazon ECS Express Mode, aggiorna per visualizzare il ruolo che hai creato.

Note

La prima volta che crei un servizio nella console Amazon ECS, alcuni utenti potrebbero visualizzare un messaggio `Invalid Parameter Exception: Unable to assume the service linked role`. Se ciò si verifica, attendi qualche secondo e riprova.

4. Scegli Create (Crea). Hai appena creato il tuo primo servizio in modalità Amazon ECS Express!

Fasi successive

Dopo aver creato il tuo primo servizio Express Mode:

- Accedi all'applicazione utilizzando l'URL dell'applicazione fornito una volta completata la distribuzione.
- Monitora l'avanzamento della distribuzione nella scheda Risorse della console. Una volta completata l'implementazione, il servizio è pronto a ricevere traffico. Per ulteriori informazioni, consulta [the section called “Visualizzazione dei dettagli di un servizio in modalità Express”](#).
- Per tornare al servizio, seleziona uno dei Cluster di navigazione, quindi seleziona il default Cluster se non ne hai specificato uno durante la creazione. Amazon ECS organizza i servizi in raggruppamenti logici chiamati cluster. Questo è un modo semplice per organizzare le applicazioni nel tuo account.
- In caso di problemi durante la distribuzione, consulta [the section called “Risoluzione dei problemi”](#).

Personalizzazione

Facoltativamente, puoi anche personalizzare la configurazione del servizio Express Mode. Apri la sezione intitolata Configurazioni aggiuntive - opzionale.

1. Seleziona un cluster dal menu a discesa se esiste un cluster Amazon ECS specifico in cui desideri distribuire il servizio. Se non viene specificato alcun cluster, Express Mode utilizzerà il Default cluster. Se lasci la console Express Mode per creare un cluster, utilizza il pulsante di aggiornamento per compilare il menu a discesa.
2. Inserisci un nome per il servizio. Se non viene specificato alcun nome, Express Mode ne genererà uno dal nome dell'immagine del contenitore. Questo nome di servizio verrà utilizzato in diverse risorse, tra cui l'URL dell'applicazione, il servizio Amazon ECS, la definizione delle attività di Amazon ECS, il certificato, l'obiettivo di scalabilità e la politica di scalabilità.
3. Facoltativamente, specifica i dettagli sul tuo contenitore:
 - a. Per Container port, inserisci la porta su cui l'applicazione è in ascolto (l'impostazione predefinita è 80).
 - b. Per Health check path, inserire il percorso per i controlli sanitari (ad esempio, `/health`). Il valore predefinito è `/`.
 - c. In Variabili di ambiente, aggiungete coppie chiave-valore per le variabili di ambiente necessarie all'applicazione. Per Key, inserisci il nome della variabile di ambiente. Per Tipo di valore, scegliete Variabile di ambiente o Segreto. Per Valore o valore da, inserite il valore o il riferimento. Scegliete Aggiungi variabile di ambiente per aggiungere altre variabili, se necessario. Non vengono fornite impostazioni predefinite.

- d. Per Command, inserisci facoltativamente un comando personalizzato per sovrascrivere l'istruzione Docker. CMD Non viene fornito alcun valore predefinito.
 - e. Per il ruolo Task, scegli un ruolo IAM che conceda le autorizzazioni alla tua applicazione. Non viene fornito alcun valore predefinito.
4. Se lo desideri, specifica i dettagli relativi a Compute e Auto scaling:
 - a. Per CPU, scegli l'allocazione vCPU per le tue attività (l'impostazione predefinita è 1 vCPU).
 - b. Per Memoria, scegli l'allocazione di memoria per le tue attività (l'impostazione predefinita è 2 GB).
 - c. Per la metrica del servizio ECS, scegli la metrica in base alla quale scalare (l'impostazione predefinita è Utilizzo medio della CPU).
 - d. Per Valore Target, scegliete la percentuale target per la scalabilità (l'impostazione predefinita è 60).
 - e. Per Numero minimo di attività e Numero massimo di attività, imposta i limiti di scalabilità. (I valori predefiniti sono 1 e 20)
5. Seleziona la casella Personalizza le configurazioni di rete. Se non personalizzi queste configurazioni, Express Mode utilizzerà il VPC predefinito.
 - a. Seleziona un VPC per aiutarti a navigare rapidamente verso le sottoreti in cui verranno eseguiti i tuoi servizi. Facoltativamente, esci dalla console Amazon ECS per creare un nuovo VPC, quindi torna ad aggiornare e trova il VPC nel menu a discesa.
 - b. Per i gruppi di sicurezza, scegli o crea gruppi di sicurezza per consentire un accesso aggiuntivo alla rete in entrata al tuo servizio.
6. In Registri:
 - a. Per Amazon CloudWatch log group, inserisci il nome del gruppo di log preferito per i log delle tue applicazioni. Il gruppo di log predefinito è denominato in base ai nomi del cluster e del servizio.
 - b. Per il prefisso Amazon CloudWatch log stream, inserisci un prefisso preferito per i flussi di log. Il prefisso predefinito dello stream è `ecs/Main/`.
7. In Tag, aggiungi coppie chiave-valore per etichettare le tue risorse. In Key (Chiave), inserire il nome della chiave. In Value (Valore), inserire il valore del tag. Scegli Aggiungi nuovo elemento per aggiungere altri tag, se necessario. I tag possono essere aggiunti solo durante la creazione.

Crea il tuo primo servizio Express Mode utilizzando il AWS CLI

Questo tutorial mostra come creare un'applicazione di servizio Express Mode utilizzando AWS CLI. Implementerai una semplice applicazione web che dimostra le funzionalità principali del servizio Express Mode.

Prerequisiti

Prima di iniziare, assicurati di disporre dei seguenti elementi:

- Un AWS account con le autorizzazioni appropriate
- L'AWS CLI installato e configurato
- Un'immagine del contenitore archiviata in Amazon ECR o in un registro privato

Fase 1: creazione di ruoli IAM

Un servizio Express Mode richiede due ruoli IAM. Il Task Execution Role consente ad Amazon ECS di estrarre immagini di container e scrivere log per tuo conto. Crea un ruolo di esecuzione delle attività e un ruolo di infrastruttura con le seguenti politiche:

```
#Create the roles with ECS trust policies
aws iam create-role --role-name ecsTaskExecutionRole \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
      {
        "Effect": "Allow",
        "Principal": {
          "Service": "ecs-tasks.amazonaws.com"
        },
        "Action": "sts:AssumeRole",
      }
    ]
  }'

aws iam create-role --role-name ecsInfrastructureRoleForExpressServices \
  --assume-role-policy-document '{
    "Version": "2012-10-17",
    "Statement": [
```

```
{
  "Sid": "AllowAccessInfrastructureForECSExpressServices",
  "Effect": "Allow",
  "Principal": {
    "Service": "ecs.amazonaws.com"
  },
  "Action": "sts:AssumeRole"
}
]
```

```
#Attach the AWS managed policies
aws iam attach-role-policy --role-name ecsTaskExecutionRole \
  --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
aws iam attach-role-policy --role-name ecsInfrastructureRoleForExpressServices \
  --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSInfrastructureRoleforExpressGatewayServices
```

Per ulteriori informazioni, consulta [the section called “Ruolo IAM di esecuzione di attività”](#).

Fase 2: Creare la prima applicazione di servizio Express Mode

Creare un'applicazione di servizio Express Mode con i parametri minimi richiesti:

```
aws ecs create-express-gateway-service \
  --image "public.ecr.aws/nginx/nginx:latest" \
  --execution-role-arn arn:aws:iam::123456789012:role/ecsTaskExecutionRole \
  --infrastructure-role-arn arn:aws:iam::123456789012:role/
ecsInfrastructureRoleForExpressServices \
  --monitor-resources
```

Questo comando crea un'applicazione di servizio Express Mode con:

- Un nome di servizio univoco generato dal nome dell'immagine
- 1 vCPU e 2 GB di memoria
- Porta 80 per il traffico HTTPS
- Scalabilità automatica basata sull'utilizzo della CPU
- Un Application Load Balancer connesso a Internet nel VPC predefinito e nelle sottoreti pubbliche
- Un URL univoco per questo servizio nel formato `servicename.ecs.region.on.aws`

Il comando monitora continuamente le risorse man mano che vengono fornite e ne restituisce lo stato. Una volta completata la distribuzione del servizio, l'URL dell'applicazione è pronto per ricevere traffico. Quando si interrompe il monitoraggio, vengono restituite le informazioni sul servizio creato, inclusi l'ARN e l'URL del servizio:

```
{
  "service": {
    "cluster": "arn:aws:ecs:region:123456789012:cluster/default",
    "serviceName": "nginx-1234",
    "serviceArn": "arn:aws:ecs:region:123456789012:service/default/nginx-1234",
    "infrastructureRoleArn": "arn:aws:iam::123456789012:role/ecsInfrastructureRoleForExpressServices",
    "status": {
      "statusCode": "ACTIVE"
    },
    "createdAt": "UNIXTIMESTAMP"
  }
}
```

Fase 3: Creare un'applicazione di servizio Express Mode con impostazioni personalizzate

È possibile personalizzare l'applicazione di servizio Express Mode specificando parametri aggiuntivi:

```
aws ecs create-express-gateway-service \
  --execution-role-arn arn:aws:iam::123456789012:role/ecsTaskExecutionRole \
  --infrastructure-role-arn arn:aws:iam::123456789012:role/ecsInfrastructureRoleForExpressServices \
  --primary-container \
    '{"image":"123456789012.dkr.ecr.region.amazonaws.com/my-app:latest", \
    "containerPort":8080, \
    "environment":[{"name":"ENV","value":"production"}, \
    {"name":"DEBUG","value":"false"}]}' \
  --service-name "my-web-app" \
  --cpu 2 \
  --memory 4 \
  --health-check-path "/health" \
  --scaling-target '{"minTaskCount":3,"maxTaskCount":100}' \
  --monitor-resources
```


Questo crea un'applicazione con:

- Un nome personalizzato "my-web-app»
- 2 vCPU e 4 GB di memoria
- Porta 8080 per il traffico delle applicazioni
- Endpoint personalizzato per il controllo dello stato
- Variabili di ambiente
- Minimo 3 attività, massimo 100 attività per la scalabilità automatica

Fase 4: Monitora la distribuzione

Il `--monitor-resources` flag funziona su qualsiasi chiamata di creazione, aggiornamento o eliminazione ai servizi in modalità Express. Inoltre, è possibile monitorare le risorse di un servizio in qualsiasi momento, indipendentemente da un'azione di mutazione. I tempi di implementazione possono variare a seconda delle risorse da fornire. Una volta modificato lo stato `ACTIVE`, l'applicazione è pronta a ricevere traffico.

```
aws ecs monitor-express-gateway-service --service-arn
arn:aws:ecs:region:123456789012:service/app-23d97h88
```

È inoltre possibile trovare la configurazione e lo stato correnti dell'applicazione di servizio Express Mode:

```
aws ecs describe-express-gateway-service --service-arn
arn:aws:ecs:region:123456789012:service/app-23d97h88
```

Fase 5: Accedere all'applicazione

Accedi all'applicazione del servizio Express Mode utilizzando l'URL fornito quando diventa attiva. Il formato dell'URL è:

```
https://<service-name>.ecs.<region>.on.aws/
```

Esempio:

```
https://app-23d97h88.ecs.us-west-2.on.aws/
```

L'applicazione è ora in esecuzione con:

- SSL/TLS Terminazione automatica
- Bilanciamento del carico su più zone di disponibilità
- Scalabilità automatica basata sull'utilizzo della CPU
- CloudWatch registrazione e monitoraggio
- 5XX Rollback Alarms e Canary Deployments per aggiornamenti futuri

Risorse create dai servizi Amazon ECS Express Mode

Un servizio in modalità Express riduce la complessità della distribuzione di applicazioni containerizzate fornendo impostazioni predefinite ragionevoli e automatizzando la configurazione dei servizi di supporto. AWS Invece di gestire centinaia di parametri di configurazione su più servizi, un servizio in modalità Express richiede solo un'immagine del contenitore, un ruolo di esecuzione delle attività e un ruolo di infrastruttura per iniziare.

I servizi in modalità Express hanno procedure semplificate per il provisioning e la configurazione di un servizio Amazon ECS che raggruppa l'infrastruttura preferita per l'esecuzione di un'applicazione Web di produzione. I servizi in modalità Express incorporano automaticamente le migliori pratiche operative e di configurazione. Vengono create le seguenti risorse:

- Il cluster ECS predefinito (se non esiste già) con i provider di capacità Fargate
- Una definizione di attività con configurazioni di container, logging e rete
- Un servizio con implementazione canaria e scalabilità automatica
- Application Load Balancer con listener HTTPS, regole di listener e gruppi target
- Gruppi di sicurezza con ingresso minimo richiesto
- Service Linked Roles per l'auto-scaling e il bilanciamento del carico
- Politica di scalabilità degli obiettivi e del tracciamento dei target di Application Auto Scaling.

La politica di scalabilità aggiunge o rimuove le attività di servizio necessarie per mantenere la metrica pari o vicina al valore target specificato. Oltre a mantenere la metrica vicina al valore target, una politica di scalabilità basata sul tracciamento degli obiettivi si adatta anche alle fluttuazioni della metrica dovute a uno schema di carico fluttuante e riduce al minimo le rapide fluttuazioni nel numero di attività in esecuzione nel servizio.

- Gruppo di log specifico del servizio

- Allarme metrico per il rilevamento di implementazioni difettose
- Certificato ACM

Utilizza le impostazioni predefinite per iniziare a utilizzare i servizi Express Mode o configura l'applicazione utilizzando un set minimo di parametri.

Impostazioni predefinite del cluster

Le seguenti impostazioni predefinite sono configurabili dalla modalità Express per il cluster:

- `ClusterName`: utilizza il cluster. `default`

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o specificando `create-express-gateway-service --cluster`

Le seguenti impostazioni predefinite sono impostate da Express Mode ma possono essere aggiornate direttamente nel Cluster.

- `CapacityProviders`: `["FARGATE"]` - Utilizza Fargate

Impostazioni predefinite per la definizione delle attività

Le seguenti impostazioni predefinite sono configurabili in modalità Express per la definizione dell'attività:

- `cpu`: 1024 - 1 unità vCPU allocata all'attività

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --cpu`

- `memoria`: 2048 - 2 GB di memoria allocata all'attività

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --memory`

Le seguenti impostazioni predefinite sono impostate da Express Mode ma possono essere aggiornate direttamente nella Task Definition.

- `NetworkMode`: `«awsvpc»`: utilizza la modalità di rete `awsvpc`

- `operatingSystemFamily`: «LINUX» - Funziona sul sistema operativo Linux
- `CPUArchitecture`: «X86_64» - Utilizza l'architettura del processore x86_64
- `requiresCompatibilities`: ["FARGATE"] - Compatibile con il tipo di lancio Fargate
- `PlatformVersion`: «LATEST» - Utilizza l'ultima versione della piattaforma Fargate

Impostazioni predefinite per la definizione del contenitore

Le seguenti impostazioni predefinite sono configurabili dalla modalità Express per il contenitore:

- `port`: 80 - Porta predefinita per il traffico di container

Le seguenti impostazioni predefinite sono impostate dalla modalità Express ma possono essere aggiornate direttamente in Task Definition - Container Definition:

- `essential`: true - Il contenitore è contrassegnato come essenziale per lo stato delle attività
- `protocol`: «tcp» - Protocollo predefinito per la comunicazione con il bilanciamento del carico
- `name`: «Main» - Nome predefinito per il contenitore principale

La modifica del nome del contenitore predefinito può influire sulla capacità di Express Mode di apportare aggiornamenti successivi al servizio. Questa operazione non è consigliata se si intende continuare a utilizzare la console Express Mode o APIs

- `VersionConsistency`: abilitato: risolve i tag di immagine da digest per distribuzioni coerenti IDs
- `initProcessEnabled`: true - Abilita il supporto del processo di inizializzazione di Docker.
- `stopTimeout`: 30 secondi - Tempo tra i segnali SIGTERM e SIGKILL.

Impostazioni predefinite di registrazione

Le seguenti impostazioni predefinite sono configurabili dalla modalità Express per i registri: CloudWatch

- `<cluster><name>awslogs-group`: «/aws/ecs/ -####» - Nome del gruppo di log derivato dal nome del cluster e del servizio

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --primary-container "awsLogsConfiguration"=[{"logGroup"="/my/logGroup/"}]`

- `awslogs-stream-prefix`: «ecs» - Prefisso per i nomi dei flussi di log

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --primary-container "awsLogsConfiguration"=[{"logStreamPrefix"="myprefix"}]`

Le seguenti impostazioni predefinite sono impostate da Express Mode ma possono essere aggiornate direttamente in Task Definition - Log Configuration:

- `LogDriver`: «awslogs» - Utilizza i log CloudWatch
- `awslogs-region`: current Region: utilizza la stessa regione del servizio Express Mode
- `mode`: «non-blocking» - Modalità di registrazione impostata su non bloccante
- `max-buffer-size`: «25m» - Dimensione massima del buffer per le scritture di log

Impostazioni predefinite del servizio

Le seguenti impostazioni predefinite sono configurabili dalla modalità Express per il servizio Amazon ECS:

- `ServiceName`: servizio Express Mode → Nome - Nome del servizio (fornito dal cliente o derivato dal nome dell'immagine)

Il nome può essere configurato solo al momento della creazione e non è disponibile per la configurazione negli aggiornamenti.

- `cluster`: servizio Express Mode → Cluster - Cluster ECS per questo servizio (fornito dal cliente o predefinito)

Il cluster può essere configurato solo al momento della creazione e non è disponibile per la configurazione in caso di aggiornamenti.

- `DesiredCount`: servizio Express Mode → `MinTasks` - Conteggio desiderato per il servizio (il valore predefinito è 1)

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --scaling-target '{"minTaskCount"=3}'`,

- `tag`: servizio Express Mode Tag per l' AWS etichettatura (forniti dal cliente)

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o eseguendo `create-express-gateway-service --tags`. I tag possono essere aggiunti solo al momento della creazione di una nuova risorsa.

Le seguenti impostazioni predefinite sono impostate dalla modalità Express ma possono essere aggiornate direttamente nel servizio Amazon ECS:

- `availabilityZoneRebalancing`: `true` - Abilita il ribilanciamento automatico AZ
- `CapacityProviderStrategy`: `{"base": 1, "capacityProvider": "FARGATE", "weight": 1}` - Utilizza Fargate per la capacità di calcolo
- `DeploymentConfiguration`: Canary per impostazione predefinita: il servizio Express Mode utilizza le distribuzioni Canary

Tieni presente che la strategia di distribuzione non può essere aggiornata sui servizi Express Mode.

- `enable ECSTaskManaged Tags`: `true` - Abilita l'etichettatura delle risorse gestite da ECS
- `healthCheckGracePeriodSeconds`: 300 - Periodo di grazia prima che Scheduler esamini i controlli di integrità ELB o Lattice (si allinea al periodo di grazia predefinito per il controllo dello stato di salute ELB)
- `<Not set>LaunchType`: - Utilizza fornitori di capacità
- `LoadBalancers`: configurazione del bilanciamento del carico gestita da ECS

Tieni presente che le configurazioni del load balancer non possono essere aggiornate sui servizi Express Mode.

- Strategia di posizionamento: non impostata: utilizza Fargate all'avvio (deve sempre essere AZ Spread)
- `PlatformVersion`: ULTIMA - Versione della piattaforma Fargate (codificata secondo la versione PIÙ RECENTE)
- `propagateTags`: «SERVICE» - Propaga i tag dal servizio alle attività
- `SchedulingStrategy`: «REPLICA» - I servizi Express Mode sono servizi REPLICA
- `TaskDefinition`: servizio Express Mode creato: la definizione dell'attività da utilizzare (fornita dal servizio Express Mode)

Impostazioni predefinite della configurazione di rete

Le seguenti impostazioni predefinite sono configurabili dalla modalità Express per il servizio Amazon ECS - Configurazione di rete:

- `NetworkConfiguration.subnets`: se non ne viene fornita nessuna, Express Mode utilizzerà le sottoreti pubbliche predefinite nel VPC predefinito.

Il VPC predefinito deve avere almeno due sottoreti pubbliche, in almeno due zone di disponibilità, di cui almeno 8 IPs disponibili gratuitamente, per blocco CIDR assegnato, per sottorete.

Se fornisci sottoreti pubbliche personalizzate, Express Mode fornirà un ALB con accesso a Internet e attiverà `AssignPublicIP` per le tue attività. Se fornite sottoreti private (sottoreti senza un gateway Internet nella tabella di routing), Express Mode fornirà un ALB interno.

Se si specificano sottoreti, il primo servizio Express Mode per un VPC definisce le sottoreti associate al bilanciamento del carico con accesso a Internet o interno per quel VPC. I servizi successivi in modalità Express che vengono avviati nello stesso VPC devono disporre di sottoreti che corrispondono alle zone di disponibilità supportate dal load balancer. Per questo motivo, consigliamo di creare servizi Express Mode con sottoreti provenienti da tutte le zone di disponibilità.

- Configurazione di rete. `SecurityGroups`: Se non ne viene fornito nessuno, Express Mode crea sia un Service Security Group che un Load Balancer Security Group

Il Service Security Group consente il traffico in uscita verso la rete Internet pubblica, ma ciò può essere limitato a seconda di altri aspetti della rete, come la configurazione della sottorete.

Il gruppo di sicurezza Load Balancer Il gruppo di sicurezza Load Balancer consente il traffico in entrata per la risoluzione HTTPS e il traffico in uscita verso il gruppo di sicurezza del servizio sulla porta container specificata (impostazione predefinita 80). E viene aggiornato automaticamente in base alla porta container fornita nelle chiamate Express Mode `Create` o `Update`.

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o `create-express-gateway-service --network-configuration '{"securityGroup": ["sg-xxxxxxx"]}'` eseguendo. Quando si fornisce un gruppo di sicurezza, si fornisce un percorso di ingresso aggiuntivo al servizio.

Le seguenti impostazioni predefinite sono impostate da Express Mode ma possono essere aggiornate direttamente in Amazon ECS Service - Network Configuration:

- `assignPublicIp`: In base al tipo di sottorete: abilitato per le sottoreti pubbliche per garantire che gli utenti del VPC predefinito possano accedere alla rete Internet pubblica, abilitiamo il pubblico per ogni attività per impostazione predefinita. Questa opzione è disabilitata se fornisci una sottorete privata e sei quindi responsabile della configurazione di un gateway NAT se le tue attività richiedono l'accesso a Internet.

Impostazioni predefinite dei ruoli IAM

I seguenti ruoli IAM sono configurabili per un servizio in modalità Express con creazione automatica di ruoli collegati ai servizi, ove appropriato.

- `executionRoleArn`: Ruolo di esecuzione dell'attività (parametro obbligatorio)

Sebbene le autorizzazioni richieste siano incluse nella politica gestita, è possibile allegare politiche e autorizzazioni aggiuntive. Ulteriori dettagli su [Amazon ECSTask ExecutionRolePolicy](#) nella AWS Managed Policy Reference Guide

- `infrastructureRoleArn`: Ruolo dell'infrastruttura per Express Gateway Services (parametro obbligatorio)

Sebbene le autorizzazioni richieste siano incluse nella policy gestita, è possibile allegare policy e autorizzazioni aggiuntive. Ulteriori dettagli su [Amazon ECSInfrastructure RoleforExpressGatewayServices](#) Managed Policy nella AWS Managed Policy Reference Guide

- `taskRoleArn`: Task Role per le chiamate dal codice dell'applicazione del contenitore (opzionale)

È possibile personalizzare questa opzione utilizzando l'opzione Configurazioni aggiuntive nella console o `create-express-gateway-service --task-role-arn` eseguendo.

È necessario creare il nuovo ruolo IAM prima di creare o aggiornare il servizio Express Mode.

I seguenti ruoli IAM vengono configurati automaticamente da Express Mode.

- `ecsServiceRoleForECS`: consente ad Amazon ECS di gestire il cluster
- `AWSServiceRoleForElasticLoadBalancing`: richiama altri AWS servizi per tuo conto, creati automaticamente da Elastic Load Balancing

- `AWSServiceRoleForApplicationAutoScaling_ECSService`: chiama altri AWS servizi per conto dell'utente, creati automaticamente da Application Auto Scaling

Impostazioni predefinite di Application Auto Scaling

Le seguenti impostazioni di Application Auto Scaling possono essere configurate in modalità Express:

- `autoScalingTargetValore`: 60 - Percentuale di utilizzo della CPU prevista per la scalabilità
- `autoScalingMetric`: "CPUUtilization" - Metrica utilizzata per le decisioni relative alla scalabilità automatica
- `desiredMinTaskNumero`: 1 - Numero minimo di attività da mantenere
- `desiredMaxTaskNumero`: 20 - Numero massimo di attività a cui è possibile adattare (può essere aumentato o diminuito)

Le seguenti impostazioni possono essere configurate direttamente in Application Auto Scaling:

- `PolicyType`: "TargetTrackingScaling" - Utilizza la politica di ridimensionamento del tracciamento degli obiettivi
- `disableScaleIn`: false: consente il ridimensionamento quando la domanda diminuisce

Impostazioni predefinite del target di scalabilità delle applicazioni

Gli obiettivi Application Auto Scaling definiscono i limiti scalabili di risorse e capacità per le operazioni di scalabilità automatica. In modalità Express è possibile modificare quanto segue:

- `MaxCapacity`: Servizio in modalità Express → `MaxTasks` - Numero massimo di attività a cui il servizio sarà esteso (impostazione predefinita `CreateLoadBalancedService` o fornita dal cliente)
- `MinCapacity`: Servizio Express Mode → `MinTasks` - Attività minime a cui il servizio verrà ridimensionato (impostazione predefinita `CreateLoadBalancedService` o fornita dal cliente)
- `ScalableDimension`: `ecs:service: DesiredCount` - La proprietà da scalare
- `Tag`: derivati dai tag passati `CreateLoadBalancedService` - Tag di risorsa per questa risorsa

Le seguenti impostazioni predefinite possono essere modificate nel target scalabile di Application Auto Scaling:

- `DynamicScalingInSuspended`: false - Indica se lo scaling in è sospeso
- `DynamicScalingOutSuspended`: false - Indica se la scalabilità orizzontale è sospesa
- `ScheduledScalingSuspended`: false - Indica se il ridimensionamento pianificato è sospeso

Impostazioni predefinite dei criteri di scalabilità delle applicazioni

Le politiche di scalabilità implementano algoritmi di tracciamento degli obiettivi che monitorano l'utilizzo della CPU e regolano il numero di attività per mantenere prestazioni ottimali. Dalla modalità Express è possibile modificare quanto segue:

- `TargetValue`: 60% - Il valore obiettivo per la politica di tracciamento (servizio Express Mode → `scaling-target-value`)
- `PredefinedMetricType`: `ECSService MediaCPUUtilization` : il nome, il tipo di metrica da tracciare (servizio Express Mode → `scaling-metric`)

La modalità Express fornisce metriche di scalabilità per l'utilizzo medio `CPUUtilization`, medio della memoria e il numero di richieste per destinazione. Il numero di richieste per target proviene dall'Application Load Balancer e può essere impostato su 65536 richieste per target al secondo, ovvero il servizio Amazon ECS nel gruppo target Application Load Balancer.

Le seguenti impostazioni predefinite sono impostate da Express Mode ma possono essere modificate nella politica Application Auto Scaling:

- `PolicyName`: `< ServiceName >< TargetMetric >` - Nome di questa politica di scalabilità (derivato dal nome del servizio ECS e da Target Metric)
- `PolicyType`: `TargetTrackingScaling` - Il tipo di ridimensionamento da eseguire (impostazione predefinita)
- `ScalableDimension`: `«ecs:service:DesiredCount»` - La proprietà da ridimensionare (impostazione predefinita)
- `DisableScaleIn`: false - Indica se la scalabilità in è disabilitata

Impostazioni predefinite di Application Load Balancer

Le seguenti impostazioni predefinite di Application Load Balancer possono essere configurate in modalità Express:

- **schema:** dipende dalle sottoreti - Crea un sistema di bilanciamento del carico rivolto a Internet con indirizzi IP pubblici quando vengono fornite sottoreti pubbliche. Crea un sistema di bilanciamento del carico interno con indirizzi IP privati quando vengono fornite sottoreti private.
- **ip-address-type:** dipende dalle sottoreti: crea un Application Load Balancer IPv4 solo quando gli indirizzi lo sono. IPv4 Quando sono presenti sottoreti IPv6 abilitate, verrà creato un Application Load Balancer dual-stack. Tieni presente che se disponi di sottoreti IPv6 abilitate, il primo servizio Express Mode in un VPC definisce il sistema di bilanciamento del carico interno o connesso a Internet per quel VPC. Per questo motivo, ti consigliamo di creare prima le tue IPv6 sottoreti solo o in un nuovo VPC.

Le seguenti impostazioni predefinite sono impostate in modalità Express e possono essere configurate in Application Load Balancer:

- **desync-mitigation-mode:** Disattivata: la mitigazione della desincronizzazione HTTP è disabilitata
- **access-logs.enabled:** false - La registrazione degli accessi è disabilitata
- **https - Uses HTTPS protocol for secure comlistener-configurations.protocol:** comunicazione
- **listener-configurations.port:** 443 - Ascolta sulla porta HTTPS standard
- **listener-configurations.rule-type:** host-header: indirizza il traffico in base alle regole dell'intestazione dell'host

Impostazioni predefinite del gruppo target

Le seguenti impostazioni predefinite di Amazon EC2 Target Group possono essere configurate in modalità Express:

- **health-check-path:** (impostazione predefinita «/») Percorso di controllo dello stato del servizio in modalità Express - percorso URL per le richieste di controllo sanitario
- **port:** (default 80) - Porta su cui le destinazioni ricevono traffico
- **health-check-port:** (impostazione predefinita 80) - Porta per le richieste di controllo sanitario

Le seguenti impostazioni predefinite sono impostate dalla modalità Express e possono essere configurate in Amazon EC2 Target Groups:

- **protocollo:** HTTP - Protocollo per l'instradamento del traffico verso gli obiettivi
- **protocol-version:** HTTP1 - Versione del protocollo HTTP per la comunicazione

- `vpc-id`: l'impostazione predefinita è l'identificatore VPC - Virtual Private Cloud predefinito per il gruppo target, ma verrà derivato dalle sottoreti fornite
- `health-check-protocol`: Uguale a `Protocol`: Protocollo per le richieste di controlli sanitari
- `health-check-enabled`: Sempre abilitato: i controlli Health vengono abilitati automaticamente
- `health-check-interval-seconds`: 30 - Intervallo tra i controlli sanitari dei singoli obiettivi
- `health-check-timeout-seconds`: 5 - Durata del timeout per le risposte ai controlli sanitari
- `healthy-threshold-count`: 5 - Sono necessari controlli sanitari consecutivi con esito positivo per verificare lo stato di salute
- `unhealthy-threshold-count`: 2 - Sono necessari controlli sanitari consecutivi non riusciti per verificare lo stato di salute
- `target-type`: `ip` - Le destinazioni sono registrate in base all'indirizzo IP
- `ip-address-type`: `ipv4` - Utilizza IPv4 gli indirizzi per la registrazione degli obiettivi

Condivisione delle risorse e ottimizzazione dei costi

I servizi Express Mode condividono automaticamente le risorse quando possibile per ottimizzare i costi:

- **Condivisione del Load Balancer**: fino a 25 servizi Express Mode nello stesso VPC possono condividere un Application Load Balancer. Express Mode fornirà ulteriori Application Load Balancer, se necessario, a seconda della quantità di servizi Express Mode forniti. Inoltre, riducendo la quantità di servizi Express Mode nel VPC, Express Mode eliminerà anche gli Application Load Balancer non utilizzati. Questa condivisione riduce il costo effettivo per applicazione man mano che si implementano più servizi Express Mode.
- **Condivisione di cluster**: i servizi in modalità Express possono essere raggruppati in Amazon ECS Clusters. I servizi in modalità Express possono anche condividere Amazon ECS Clusters con servizi Amazon ECS non gestiti da Amazon ECS Mode.

Visualizzazione dei dettagli di un servizio in modalità Amazon ECS Express

Puoi visualizzare i dettagli sul tuo servizio Express Mode per monitorarne lo stato, la configurazione e i parametri prestazionali. La pagina dei dettagli del servizio fornisce informazioni complete sullo stato

attuale del servizio, sulle attività in esecuzione, sulla cronologia della distribuzione e sulle metriche operative.

Per visualizzare un servizio in modalità Express (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 Cluster.
3. Nella pagina Cluster, scegli il nome del cluster che contiene il tuo servizio Express Mode.
4. Nella pagina dei dettagli del cluster, scegli la scheda Servizi.
5. Configura un filtro per visualizzare i servizi in modalità Express. Per il tipo di gestione delle risorse Filter, scegli ECS.

Un servizio Express Mode ha un badge Express accanto al nome.

6. Seleziona il nome del tuo servizio Express Mode per accedere alla pagina dei dettagli del servizio Express Mode.
7. Controlla le informazioni sul servizio visualizzate nella pagina dei dettagli del servizio:
 - a. **Panoramica del servizio Express:** visualizza le informazioni di base sul servizio, tra cui l'URL dell'applicazione, lo stato del servizio, le attività in sospeso e in esecuzione e le distribuzioni attive.
 - b. **Osservabilità:** visualizza le CloudWatch metriche relative all'utilizzo della CPU (minimo, massimo, medio), all'utilizzo della memoria (minimo, massimo, medio), al tempo di risposta previsto (p50, p90, p99), agli stati 4xx e agli stati 5xx. RequestCount
 - c. **Registri:** visualizza i log dei contenitori relativi alle tue attività.
 - d. **Risorse:** le AWS risorse create per il servizio Express Mode. La visualizzazione Cronologia si aggiorna automaticamente dopo una creazione o un aggiornamento e mostra lo stato e gli eventi. La visualizzazione Tabella fornisce una tabella ricercabile con lo stato e il numero completo ARNs di ogni risorsa.
 - e. **Implementazioni:** monitora la cronologia delle distribuzioni e lo stato attuale della distribuzione, include la revisione del servizio e le attività che forniscono informazioni preziose durante gli aggiornamenti. Il pannello degli eventi consente di sapere quando il servizio Express Mode ha raggiunto uno stato stazionario.
 - f. **Configurazione:** visualizza la configurazione del servizio, inclusi il cluster, i ruoli, la configurazione del contenitore, l'elaborazione, la scalabilità automatica e la configurazione di rete.

- g. Tag: gestisci i tuoi tag.

Aggiornamento di un servizio in modalità Amazon ECS Express

Il servizio Express Mode semplifica il processo di aggiornamento del servizio fornendo opzioni configurabili per il networking, il bilanciamento del carico e l'Application Auto Scaling e orchestrando le modifiche tra questi servizi.

È possibile aggiornare il servizio Express Mode per modificare le immagini dei contenitori, regolare l'allocazione delle risorse o modificare le impostazioni di configurazione. Gli aggiornamenti vengono distribuiti utilizzando distribuzioni canarie con allarmi di rollback basati sugli allarmi per mantenere la disponibilità.

Comportamento di distribuzione

Per impostazione predefinita, un servizio Express Mode utilizza le distribuzioni Canary per garantire aggiornamenti sicuri e ripristini rapidi:

- Viene creato un nuovo ambiente e vengono distribuite le attività con le modifiche
- Il 5% del traffico viene spostato verso il nuovo ambiente
- I rollback basati sugli allarmi si attivano se la somma della percentuale di errori 4xx e 5xx è > 1 per 2 punti dati entro 3 minuti
- I controlli sanitari verificano che le nuove attività siano corrette
- Dopo 3 minuti di cottura, il 100% del traffico viene spostato verso un nuovo ambiente
- Dopo 3 minuti di tempo di cottura per il monitoraggio, le vecchie attività vengono gradualmente interrotte e sostituite

Per ulteriori informazioni, consulta [Amazon ECS Canary](#) deployments

Procedura

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 Cluster.
3. Nella pagina Cluster, scegli il nome del cluster che contiene il tuo servizio Express Mode.
4. Nella pagina dei dettagli del cluster, scegli la scheda Servizi.

5. Configura un filtro per visualizzare i servizi in modalità Express. Per il tipo di gestione delle risorse Filter, scegli ECS.

Un servizio Express Mode ha un badge Express accanto al nome.

6. In Configurazione:
 - a. Specificate l'immagine da utilizzare per l'applicazione. Per l'URI dell'immagine, inserisci l'URI dell'immagine. Per sfogliare le tue immagini Amazon ECR, scegli Browse ECR images, quindi procedi come segue:
 - i. Per Archivio privato, scegli l'archivio privato Amazon ECR.
 - ii. Per Image, scegli la tua immagine.
 - iii. Scegli come identificare l'immagine. Per Seleziona immagine per, scegli una delle seguenti opzioni:
 - AWS consiglia di scegliere Image digest.
 - Per usare il tag, scegli il tag Image, quindi scegli il tag.
 - b. Per utilizzare un registro privato, seleziona Registro privato. Quindi, per Secrets Manager ARN o name, inserisci l'ARN di Secrets Manager che hai creato nei prerequisiti.
 - c. Per il ruolo di esecuzione delle attività, scegli i ruoli o crea un nuovo ruolo e aggiorna. È possibile aggiornare il ruolo di esecuzione dell'attività quando è necessario aggiungere autorizzazioni aggiuntive.
7. In Configurazioni aggiuntive, personalizza il tuo servizio.
 - a. In Contenitore:
 - i. Per Container port, aggiorna la porta su cui l'applicazione è in ascolto (l'impostazione predefinita è 80).
 - ii. Per Health check path, aggiorna il percorso per i controlli sanitari (ad esempio, /health).
 - b. In Variabili di ambiente, aggiungi coppie chiave-valore per le variabili di ambiente di cui la tua applicazione ha bisogno:
 - i. Per Key, inserisci il nome della variabile di ambiente.
 - ii. Per Tipo di valore, scegliete Variabile di ambiente o Segreto.
 - iii. Per Valore o valore da, inserite il valore o il riferimento.

- iv. Scegliete Aggiungi variabile di ambiente per aggiungere altre variabili, se necessario.
 - c. Per Command, inserisci facoltativamente un comando personalizzato per sovrascrivere l'istruzione Docker CMD.
 - d. Per il ruolo Task, aggiungi un ruolo IAM che concede le autorizzazioni all'applicazione in esecuzione nei tuoi contenitori. Ciò consente all'applicazione di effettuare chiamate API ai AWS servizi.
 - e. In Compute:
 - i. Per la CPU, aggiorna l'allocazione vCPU per le tue attività (ad esempio, 1 vCPU).
 - ii. Per Memoria, aggiorna l'allocazione di memoria per le tue attività (ad esempio, 2 GB).
 - f. In Auto Scaling:
 - i. Per la metrica del servizio ECS, scegli la metrica su cui basare la scalabilità (ad esempio, ECS Service Average Memory Utilization o Request count per target).
 - ii. Per Valore Target, inserisci l'obiettivo per la scalabilità (ad esempio, 60 o 1000).
 - iii. Per Numero minimo di attività e Numero massimo di attività, aggiorna i limiti di scalabilità.
 - g. In Registri:
 - i. Per i gruppi di CloudWatch log di Amazon, aggiorna il nome del gruppo di log per i log delle applicazioni. Tieni presente che questo non sposterà i log esistenti, ma inizierà a scrivere i log a partire dalla nuova revisione del servizio.
 - ii. Per il prefisso Amazon CloudWatch log stream, inserisci un nuovo prefisso per i flussi di log.
8. Scegli Aggiorna per aggiornare il tuo servizio Express Mode.

Eliminare i servizi in modalità Amazon ECS Express

Quando si elimina un servizio in modalità Express, si rimuove il servizio e si eliminano le risorse associate, tra cui il cluster, il sistema di bilanciamento del carico e altri componenti dell'infrastruttura.

Important

L'eliminazione di un servizio in modalità Express interromperà tutte le attività in esecuzione e renderà l'applicazione non disponibile. Questa operazione non può essere annullata.

A seconda della selezione, è possibile eliminare le seguenti risorse:

- Il cluster Amazon ECS (se non sono in esecuzione altri servizi)
- Il servizio Amazon ECS, la definizione delle attività e tutte le attività in esecuzione
- Gruppo di sicurezza del servizio
- CloudWatch gruppo di log
- Allarme del parametro
- Certificato ACM
- L'Application Load Balancer (se non sono configurati altri servizi), il gruppo di destinazione, il gruppo di sicurezza, il listener e la regola del listener
- Politica di Auto Scaling, obiettivo scalabile

Considerate quanto segue:

- Assicuratevi di aver eseguito il backup di tutti i dati importanti prima dell'eliminazione
- Valuta la possibilità di ridurre il servizio a zero attività prima dell'eliminazione per bloccare il traffico in modo corretto
- Esamina le dipendenze che potrebbero essere interessate dall'eliminazione

Differenza tra l'eliminazione di un servizio Express Mode e di un servizio

Quando elimini un servizio, Amazon ECS lo elimina se non ci sono attività in esecuzione. Amazon ECS non elimina i gruppi target associati al servizio o al cluster predefinito (se è stato utilizzato). Quando elimini un servizio in modalità Express, Amazon ECS eliminerà il servizio e tutte le risorse distinte dal servizio Express Mode. Ad esempio, se il servizio Express Mode ha creato il load balancer ed è l'unico servizio che utilizza il load balancer, eliminerà il load balancer associato. Se il load balancer viene condiviso, anche con risorse non gestite da Express Mode, non verrà eliminato.

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 Cluster, scegliere il cluster.
4. Scegli la scheda Servizi.

5. Seleziona i servizi da eliminare.
6. Scegli Elimina.
7. Nella finestra di dialogo di conferma dell'eliminazione:
 - Invio **delete** per confermare l'eliminazione.
8. Seleziona Elimina per confermare.
9. Resta sulla pagina per monitorare l'avanzamento dell'eliminazione e visualizzare un elenco delle risorse che sono state rimosse, oltre a quelle che sono state conservate e che potrebbero richiedere una pulizia manuale.

Best practice per i servizi in modalità Amazon ECS Express

Scopri le best practice e i consigli per utilizzare efficacemente il servizio Express Mode negli ambienti di produzione.

Best practice di sicurezza

Gestione dei segreti

- Usa Secrets Manager per i segreti: archivia i dati sensibili in Secrets Manager (ad esempio repository privato o credenziali del database).

Per ulteriori informazioni sulle best practice di Secrets Manager, consulta le [best practice di Secrets Manager](#) nella Guida per l'utente di Secrets Manager

- Abilita la crittografia a riposo: assicurati che i segreti vengano crittografati quando vengono archiviati nei AWS servizi.

L'utilizzo di un servizio come Secrets Manager consente di crittografare utilizzando una chiave AWS gestita o fornita dal cliente.

- Implementa la rotazione segreta: utilizza la rotazione automatica per le password del database e le chiavi API.

Utilizzando un servizio come Secrets Manager puoi gestire la rotazione segreta per servizi come Amazon Aurora e Amazon RDS.

Esempio di utilizzo dei segreti nel servizio Express Mode:

```
aws ecs update-express-gateway-service \
  --primary-container \
    '{"environment":[{"name":"DB_PASSWORD","value":"arn:aws:secretsmanager:us-
west-2:123456789012:secret:prod/db/password"}, \
    {"name":"API_KEY","value":"arn:aws:ssm:us-west-2:123456789012:parameter/prod/
api-key"}]}' \
```

Sicurezza di rete

- Usa sottoreti private per applicazioni sensibili: distribuisce applicazioni che non richiedono l'accesso diretto a Internet in sottoreti private.

Per ulteriori informazioni sulle architetture consigliate, consulta l'[applicazione Connect Amazon ECS a Internet](#).

- Configura i gruppi di sicurezza in modo che siano minimamente permissivi: limita il traffico in entrata e in uscita solo alle porte e alle fonti necessarie.

Per limitare il traffico in uscita dell'Express Mode Service Security Group, puoi modificarlo direttamente nella console di Amazon EC2 Security Groups modificando le regole in uscita o utilizzando i seguenti comandi:

```
aws ec2 authorize-security-group-egress
  --group-id sg-xxxxxxx \
  --protocol tcp \
  --port 443 \
  --cidr 0.0.0.0/0

aws ec2 revoke-security-group-egress
  --group-id sg-xxxxxxx \
  --protocol tcp \
  --port 443 \
  --cidr 0.0.0.0/0
```

- Abilita Amazon VPC Flow Logs: monitora il traffico di rete per l'analisi della sicurezza e la risoluzione dei problemi.

È possibile abilitarlo in ogni sottorete utilizzata dalle applicazioni in modalità Express nella console di sottorete VPC, oppure utilizzare `aws ec2 create-flow-logs --resource-ids subnet-xxx`

- Utilizzo AWS WAF per applicazioni Web: protezione dagli exploit e dagli attacchi Web comuni.

È possibile abilitare questa funzionalità creando un ACL Web e quindi associandolo all'Application Load Balancer utilizzato dal servizio Express Mode. In Console, crea un ACL web nel servizio WAF & Shield e associalo al tuo Application Load Balancer. Oppure, usa `aws wafv2 create-web-acl` e `aws wafv2 associate-web-acl --resource-arn <alb>`

Ottimizzazione delle prestazioni e del calcolo

Dimensionamento delle risorse

- CPU e memoria di dimensioni corrette: monitora le prestazioni delle applicazioni e regola le allocazioni di CPU e memoria in base ai modelli di utilizzo effettivi.

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 .

- Verifica le prestazioni della tua applicazione: per assicurarti che l'applicazione funzioni su larga scala e con le soglie di scalabilità e le allocazioni di risorse specificate, esegui test di carico.

Configurazione con scalabilità automatica

- Imposta soglie di scalabilità appropriate: configura le soglie di CPU o memoria che attivino la scalabilità prima che le prestazioni peggiorino.

È possibile modificare il valore target della metrica di servizio nella console di servizio Express Mode.

Prendi in considerazione l'aggiunta di una politica di scalabilità predittiva, soprattutto se il traffico segue uno schema temporale. Per ulteriori informazioni, consulta [Predictive Auto Scaling](#).

- Utilizza più metriche di scalabilità: prendi in considerazione l'utilizzo sia della CPU o della memoria che della scalabilità basata su richiesta per una scalabilità più reattiva.

È possibile aggiungere più politiche a un servizio. La modalità Express ne aggiunge una per impostazione predefinita, ma è possibile allegare politiche aggiuntive direttamente al servizio.

- Configura i limiti minimi e massimi delle attività: stabilisci limiti ragionevoli per controllare i costi e garantire la disponibilità.

Per i carichi di lavoro di produzione, una volta completati i test iniziali, consigliamo di eseguire l'operazione in tre zone di disponibilità per seguire le best practice in materia di disponibilità. È possibile aggiornare il numero minimo di attività nella console Express Mode o utilizzando `update-express-gateway-service --scaling-target '{"minTaskCount":3}'`.

Controlli dell'integrità

- Implementa controlli di integrità significativi: crea endpoint di controllo dello stato che verifichino le dipendenze critiche delle applicazioni.

È possibile aggiornare il percorso di Health check nella Express Mode Console. Oppure utilizzando `update-express-gateway-service --health-check-path "/health"`.

Per ulteriori informazioni sulla creazione di controlli sanitari per la tua applicazione, consulta [Implementing Health Checks](#)

- Mantieni i controlli sanitari leggeri: evita operazioni costose sugli endpoint per i controlli sanitari.

Gli esempi potrebbero includere chiamate API esterne, operazioni che richiedono un uso intensivo della CPU o della memoria o operazioni di lunga durata con la possibilità di un timeout.

- Utilizza timeout appropriati: configura i timeout dei controlli sanitari che consentano tempi di risposta normali e rilevi rapidamente i guasti.

I timeout Health check per Express Mode possono essere configurati sul gruppo target Application Load Balancer. Nella EC2 console Amazon, vai alla sezione Target Groups e seleziona il tuo gruppo target in modalità Express. Seleziona la scheda Health Checks e fai clic su Modifica, in Impostazioni avanzate dei controlli sanitari puoi regolare il timeout. Oppure usa `aws elbv2 modify-target-group --target-group-arn <targetgroup> --health-check-timeout`.

- Restituisci i codici di stato HTTP corretti: usa 200 per gli stati integri e 4xx/5xx per gli stati non integri.

Migliori pratiche operative

Monitoraggio e registrazione dei log

- Abilita Enhanced Container Insights: utilizza CloudWatch; Enhanced Container Insights per il monitoraggio completo delle applicazioni di servizio Express Mode.

Per ulteriori informazioni, consulta [Configurazione di Container Insights su Amazon ECS](#).

- Configura parametri personalizzati: pubblica parametri specifici dell'applicazione su; per CloudWatch il monitoraggio della logica aziendale.

Per ulteriori informazioni, consulta [Metriche pubbliche personalizzate nella Guida](#) per l'utente CloudWatch .

- Configura la conservazione dei log: imposta i periodi di conservazione dei log appropriati per bilanciare costi e requisiti di conformità.

CloudWatch I gruppi di log creati da Express Mode sono configurati per non scadere mai e vengono conservati quando il servizio Express Mode viene eliminato. È possibile modificare questa impostazione nel CloudWatch Log Group.

- Crea dashboard e avvisi: configura CloudWatch; dashboard e allarmi per un monitoraggio proattivo.

Strategie di distribuzione

- Implementa i tempi di cottura: la modalità Express implementa un tempo di cottura canary per garantire che le implementazioni abbiano il tempo di stabilizzarsi, riducendo al contempo il raggio d'azione delle implementazioni problematiche. Se l'applicazione necessita di più tempo per stabilizzarsi, è possibile configurarla nella definizione del servizio Amazon ECS del servizio Express Mode. Per ulteriori dettagli, consulta [Creazione di una distribuzione Amazon ECS Canary](#).
- Implementa procedure di rollback: prepara un piano per ripristinare rapidamente le versioni precedenti in caso di problemi.

Controlli sanitari significativi e ripristini basati sugli allarmi possono entrambi aiutare con il rollback. La strategia di implementazione canary di Express Mode, combinata con i rollback basati sugli allarmi sul traffico 4xx e 5xx, consente di configurare le implementazioni per ripristini rapidi in caso di codice o configurazione dell'applicazione difettosi.

Risoluzione dei problemi dei servizi in modalità Amazon ECS Express

Questa sezione consente di identificare e risolvere i problemi più comuni durante la distribuzione e la gestione dei servizi Express Mode.

Problemi relativi alla distribuzione

Servizio bloccato nello stato ATTIVO o DRAINING

Sintomi: DescribeServiceRevisions mostra che le risorse sono ancora in fase di approvvigionamento o esaurimento. DescribeServices mostra una distribuzione non stabilizzata

Possibili cause e soluzioni:

- Autorizzazioni IAM insufficienti: verifica che il ruolo di esecuzione dell'attività e il ruolo di infrastruttura dispongano delle autorizzazioni necessarie, come indicato nelle rispettive politiche gestite.

```
# Check if the role has the required managed policy
aws iam list-attached-role-policies --role-name ecsTaskExecutionRole
```

- Errori di estrazione dell'immagine: assicurati che l'immagine del contenitore esista e sia accessibile.

```
# Test image pull manually
docker pull 123456789012.dkr.ecr.us-west-2.amazonaws.com/my-app:latest
```

- Problemi di connettività di rete: verifica che le sottoreti abbiano accesso a Internet o endpoint Amazon VPC per i servizi. AWS
- Limiti delle risorse: verifica che il tuo account abbia una capacità Fargate sufficiente e che non abbia raggiunto le quote di servizio.

Fasi diagnostiche:

1. `DescribeExpressGatewayService` Utilizzatelo per ottenere la revisione corrente del servizio, seguita da uno `DescribeServiceRevisions` per visualizzare lo `ServiceRevision` stato del provisioning o del deprovisioning
2. Controlla gli eventi del servizio nella console Amazon ECS per messaggi di errore dettagliati.
3. Verifica che la porta del container sia impostata correttamente
4. Controlla le quote AWS di servizio per Amazon ECS e Fargate.

Errori di avvio delle attività

Sintomi: le attività non vengono avviate o si interrompono immediatamente dopo l'avvio.

Cause comuni:

- Errori dell'applicazione: l'applicazione contenitore si chiude a causa di errori di configurazione o di runtime.
- Errori del controllo dello stato di salute: l'applicazione non risponde ai controlli di integrità sulla porta o sul percorso previsti.
- Limiti di risorse: allocazione insufficiente della CPU o della memoria per l'applicazione.
- Variabili o segreti di ambiente mancanti: la configurazione richiesta non è disponibile per l'applicazione.

Fasi di risoluzione:

1. Controlla i log delle applicazioni in CloudWatch Logs, ottieni il nome del gruppo di log da: `DescribeServiceRevisions`

```
aws logs describe-log-streams --log-group-name /ecs/express-service-my-app
aws logs get-log-events --log-group-name /ecs/express-service-my-app --log-stream-
name stream-name
```

2. Verificate che il percorso del controllo dello stato di salute restituisca lo stato HTTP 200.
3. Testa l'immagine del contenitore localmente per assicurarti che si avvii correttamente.
4. Esamina e modifica le allocazioni di CPU e memoria, se necessario.

Problemi di connettività

Applicazione irraggiungibile tramite il sistema di bilanciamento del carico

Sintomi: l'URL dell'applicazione restituisce timeout o errori di connessione.

Passaggi per la risoluzione dei problemi

1. Verifica che le risorse abbiano terminato il provisioning
2. Verifica che le attività siano in esecuzione e funzionino correttamente:

```
aws ecs describe-services --cluster my-cluster --services my-express-service
```

3. Verifica lo stato del gruppo target di Application Load Balancer:

```
aws elbv2 describe-target-health --target-group-arn  
arn:aws:elasticloadbalancing:region:account:targetgroup/name/id
```

4. Assicurati che l'applicazione sia in ascolto sulla porta corretta all'interno del contenitore.

Problemi di prestazioni

Tempi di risposta lenti

Sintomi: le risposte dell'applicazione sono più lente del previsto.

Approccio diagnostico:

1. Monitora l'utilizzo della CPU e della memoria:

```
# Check CloudWatch metrics for the service  
aws cloudwatch get-metric-statistics \  
  --namespace AWS/ECS \  
  --metric-name CPUUtilization \  
  --dimensions Name=ServiceName,Value=my-express-service Name=ClusterName,Value=my-  
cluster \  
  --start-time 2024-01-01T00:00:00Z \  
  --end-time 2024-01-01T00:00:00Z
```

```
--end-time 2024-01-01T01:00:00Z \  
--period 300 \  
--statistics Average
```

2. Esamina i registri delle applicazioni per eventuali errori o avvisi sulle prestazioni.
3. Verifica se il ridimensionamento automatico risponde in modo appropriato al carico.
4. Analizza le metriche del load balancer per la distribuzione delle richieste.

Strategie di ottimizzazione:

- Aumenta l'allocazione della CPU o della memoria se le risorse sono limitate.
- Modifica le soglie di scalabilità automatica per scalare prima.
- Ottimizza le query sul codice dell'applicazione e sul database.

La scalabilità automatica non funziona come previsto

Sintomi: il servizio non aumenta in caso di carico elevato o non si riduce in caso di carico basso.

Passaggi per la risoluzione dei problemi

1. Controlla le politiche di scalabilità automatica e la loro configurazione:

```
aws application-autoscaling describe-scaling-policies \  
  --service-namespace ecs \  
  --resource-id service/my-cluster/my-express-service
```

2. Rivedi le CloudWatch metriche per assicurarti che i trigger di scalabilità vengano soddisfatti.
3. Verifica che il servizio disponga dell'autorizzazione alla scalabilità (controlla i ruoli IAM).
4. Verifica eventuali attività di scalabilità e i relativi risultati.

Strumenti di monitoraggio e debug

Utilizzo di Container Insights CloudWatch

Abilita Container Insights per un monitoraggio completo:

```
aws ecs put-account-setting --name containerInsights --value enabled
```

Container Insights offre:

- Metriche relative a CPU, memoria, disco e rete
- Dashboard di monitoraggio delle prestazioni
- Correlazione e analisi dei log
- Rilevamento anomalie

Aggiornamento delle risorse al di fuori della modalità Express

Express Mode semplifica l'implementazione fornendo e gestendo automaticamente AWS le risorse per tuo conto, ma è importante comprendere il modello di responsabilità. Sebbene Express Mode crei e configuri risorse come i servizi Amazon ECS, i sistemi di bilanciamento del carico e le politiche di scalabilità automatica, tutte le risorse rimangono nel tuo AWS account e sono completamente accessibili per la gestione diretta. Puoi modificare queste risorse al di fuori della modalità Express utilizzando la AWS console AWS CLI APIs, ma ciò potrebbe influire sulla capacità di Express Mode di gestire il servizio. Questa sezione spiega quali risorse gestisce Express Mode, come apportare modifiche in modo sicuro al di fuori di Express Mode e cosa succede quando lo fai.

Comprensione del modello di responsabilità condivisa

Il limite di responsabilità in modalità Express è definito dall'utilizzo dell'API. Express Mode APIs gestisce le risorse per tuo conto con aggiornamenti coordinati su più AWS servizi. AWS Il servizio standard APIs ti offre il controllo diretto sulle singole risorse. Ciò consente di andare oltre il set di parametri fornito dalla modalità Express per la configurazione e consente di controllare ogni parametro di ogni risorsa. Entrambi i set APIs operano sulla stessa infrastruttura sottostante del tuo account ed entrambi eseguiranno le operazioni richieste. AWS ha la responsabilità di eseguire l'operazione richiesta. Express Mode non verifica se le modifiche alle risorse che utilizzano direct APIs entrino in conflitto con la configurazione della modalità Express, né impedisce le modifiche che potrebbero interrompere la funzionalità del servizio. Ciò significa anche che avete la possibilità di apportare modifiche e tali modifiche persisteranno. Express Mode non sovrascriverà le modifiche a meno che non vengano richieste come parte di un aggiornamento della modalità Express. Sei responsabile della comprensione del modo in cui le modifiche che utilizzano Direct APIs

Interagiscono con la configurazione di Express Mode e della risoluzione di eventuali conflitti o problemi di servizio che ne derivano.

Per ulteriori dettagli sulle impostazioni predefinite impostate da Express Mode e sulle risorse che orchestra, consulta Risorse [create dai servizi in modalità Amazon ECS](#) Express.

Devo fare qualcosa prima di poter sfruttare le funzionalità di Amazon ECS nel mio servizio Express Mode?

No, non esiste un percorso di graduazione o un vetro rotto per la modalità Express. L'intero set di funzionalità Amazon ECS è sempre disponibile per il servizio Express Mode. Se desideri assicurarti che non possano essere eseguiti ulteriori aggiornamenti della modalità Express sul tuo servizio, sono disponibili diverse opzioni. Rimuovendo il Managed Tag dalle tue risorse, Amazon ECS non sarà più in grado di identificarlo e utilizzarlo come servizio Express Mode. In alternativa, puoi anche utilizzare IAM per limitare le autorizzazioni degli utenti alla modalità Express. APIs

Esempi di aggiornamenti al di fuori della modalità Express

Aggiungere un sidecar a un servizio Express Mode

Puoi aggiungere qualsiasi contenitore sidecar al tuo servizio. Questo esempio mostra come aggiungere un sidecar di registrazione Fluent Bit.

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione scegliere Cluster.
3. Seleziona il cluster in cui si trova il servizio Express Mode e nel pannello Servizi, selezionalo.
4. Vai alla scheda Risorse e seleziona la definizione dell'attività.
5. Seleziona la scheda JSON, quindi seleziona la freccia rivolta verso il basso nel pulsante in alto denominato Crea nuova revisione. Qui troverai un'azione chiamata Crea nuova revisione con JSON.
6. Aggiungi il contenitore sidecar prima o dopo la definizione del contenitore principale. Per ulteriori informazioni, consulta [Esempio di definizione di attività Amazon ECS: Route logs](#) to. FireLens
7. Se disponi di un componente secondario per la registrazione, come in questo esempio, ti consigliamo di modificarlo logDriver nel contenitore principale:

```
"logConfiguration": {
```

```
"logDriver": "awsfirelens"  
}
```

8. Supponendo le impostazioni predefinite per un servizio Express Mode, imposta il contenitore principale MemoryReservation su 1998 e il contenitore Fluent Bit MemoryReservation su 50. Ciò garantisce che la riserva totale di memoria tra i due contenitori non superi la quantità di memoria in esecuzione sull'attività Fargate (2048 per un servizio Express Mode predefinito).
9. Crea la tua nuova revisione.
10. Fai clic sul menu a discesa che dice Deploy e seleziona Update Service.
11. Seleziona il cluster e il servizio che stai aggiornando, quindi seleziona Forza la nuova distribuzione. La revisione della definizione dell'attività appena creata dovrebbe essere selezionata automaticamente. Selezionare Update (Aggiorna).
12. Guarda come avviene la tua implementazione. Le attività di questo servizio avranno ora due contenitori. Gli aggiornamenti aggiuntivi verranno eseguiti tramite Express Mode Console oppure non APIs sovrascriveranno le modifiche alla definizione dell'attività, a meno che non siano in conflitto diretto. In questo caso, ad esempio, se si assegna a Express Mode un nuovo nome per il gruppo di log e il flusso di log, LogDriver verrà nuovamente aggiornato a. awslogs

Aggiungere un dominio personalizzato al servizio

Prerequisiti: questa guida presuppone che il dominio sia una zona ospitata gestita da Route 53 con un certificato ACM. Se il tuo dominio è ospitato altrove, avrai bisogno di un CNAME che punti al record DNS di Application Load Balancer.

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nel pannello di navigazione scegliere Cluster.
3. Seleziona il cluster in cui si trova il servizio Express Mode e nel pannello Servizi, selezionalo.
4. Vai alla scheda Risorse e seleziona la regola del listener.
5. Seleziona Azione e Modifica regola.
6. Copia il valore dell'intestazione Host corrente (l'URL dell'applicazione), quindi Rimuovi la regola poiché può essercene solo una per tipo.
7. Aggiungi una condizione di tipo Host header, incolla nuovamente l'URL dell'applicazione come valore della condizione dell'intestazione Host.
8. Fai clic su Aggiungi valore della condizione OR.

9. Inserisci qui il tuo dominio personalizzato.
10. Fai clic su Avanti e salva le modifiche.
11. Ora selezionate la scheda Certificati del listener Application Load Balancer.
12. Fate clic su Aggiungi certificato. Qui puoi importare il certificato nel listener.
13. Nella tua zona ospitata da Route 53, fai clic su Crea record per creare un record di routing verso il tuo servizio.
14. Seleziona un record di routing semplice.
15. Aggiungi un nome di record.
16. Seleziona un tipo di record.
17. Indirizza il traffico verso un alias verso Application and Classic Load Balancer.
18. Scegli la tua Regione.
19. Seleziona il sistema di bilanciamento del carico utilizzato dal servizio Express Mode.

Dopo alcuni minuti di propagazione, il servizio inizierà a ricevere traffico sul dominio personalizzato. Se stai aggiungendo molti certificati al tuo Application Load Balancer o eseguendo la manutenzione di molti servizi in modalità Express dall'Application Load Balancer, potresti prendere in considerazione l'innalzamento del limite di servizio per i certificati sull'Application Load Balancer.

Definizioni dei processi di Amazon ECS

Una definizione di attività è un blueprint 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:

- La capacità 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 container su cui viene eseguito il processo
- 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 di attività Amazon ECS per 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 delle definizioni di attività Amazon ECS](#)
- [Progetta la tua applicazione per Amazon ECS](#)
- [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#)
- [Utilizzo di Amazon Q Developer per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS](#)
- [Aggiornamento di una definizione di attività Amazon ECS attraverso la console](#)
- [Annullamento della registrazione di una revisione di definizione dell'attività con la console](#)
- [Eliminazione di una revisione di definizione di attività Amazon ECS attraverso la console](#)
- [Casi d'uso della definizione dell'attività di Amazon ECS](#)
- [Parametri di definizione dell'attività di Amazon ECS per istanze gestite da Amazon ECS](#)
- [Parametri di definizione di attività Amazon ECS per Fargate](#)
- [Parametri di definizione delle attività di Amazon ECS per Amazon EC2](#)
- [Modello di definizione di attività di Amazon ECS](#)
- [Esempio di definizione di attività di Amazon ECS](#)

Stati delle definizioni di attività Amazon ECS

La definizione di un'attività cambia stato quando viene creata, eliminata o quando si annulla la registrazione. È possibile visualizzare lo stato di definizione di processo nella console oppure 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`. È possibile richiedere l'eliminazione di una definizione di attività in qualsiasi momento senza che ciò abbia un impatto 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 lo stato della revisione delle definizioni di attività è `DELETE_IN_PROGRESS`, il nome della definizione di 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 si utilizza AWS Config per gestire le definizioni delle attività, viene AWS Config addebitato 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

La richiesta di eliminazione di una definizione di attività non verrà completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione di attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività autonome di Amazon ECS: la definizione di attività è necessaria affinché quest'ultima rimanga integra.
- Attività di servizio di Amazon ECS: la definizione di attività è necessaria affinché quest'ultima rimanga integra.

- Implementazioni e set di attività di servizio di Amazon ECS: la definizione di attività è necessaria quando si avvia un evento di dimensionamento per un'implementazione o un set di attività di Amazon ECS.

Se la definizione delle `DELETE_IN_PROGRESS` attività rimane invariata, è possibile utilizzare la console oppure AWS CLI identificare e quindi interrompere le risorse che bloccano l'eliminazione delle definizioni delle 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.
- Implementazioni e set di attività di servizio di Amazon ECS: dopo l'eliminazione dell'implementazione o del set di attività, l'eliminazione della definizione di attività può richiedere fino a 24 ore.

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:

- La capacità da utilizzare, che determina l'infrastruttura in cui vengono ospitate le attività.

Quando si utilizza il provider EC2 di capacità, si sceglie anche il tipo di istanza. Quando utilizzi il provider di capacità su istanze gestite da Amazon ECS, puoi fornire i requisiti di istanza per Amazon ECS per gestire la capacità di calcolo. Per alcuni tipi di istanze, come la GPU, è necessario impostare parametri specifici. Per ulteriori informazioni, consulta [Casi d'uso della definizione dell'attività di Amazon ECS](#).

- L'immagine del container, che contiene il codice dell'applicazione e tutte le dipendenze necessarie per l'esecuzione di tale codice.
- La modalità di rete da utilizzare per i container nella tua attività.

La modalità di rete determina il modo in cui l'attività comunica sulla rete.

Per le attività eseguite su EC2 istanze e istanze gestite di Amazon ECS, sono disponibili diverse opzioni, ma consigliamo di utilizzare la `awsvpc` modalità di rete. La modalità `awsvpc` di rete semplifica la rete di container offrendoti un maggiore controllo sul modo in cui le applicazioni comunicano tra loro e con gli altri servizi al tuo interno. VPCs

Per le attività eseguite su Fargate, è necessario utilizzare la modalità di rete `awsvpc`.

- La configurazione di registrazione da utilizzare per i tuoi processi.
- Tutti i volumi di dati utilizzati con il container nell'attività.

Per un elenco completo dei parametri per la definizione di attività, consulta [Parametri di definizione di attività Amazon ECS per 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 scalare tali container in modo indipendente. Ad esempio, un sito Web e un'API richiedono in genere modelli di scalabilità diversi. Con l'aumento del traffico, potrebbe essere necessario un numero diverso di container Web rispetto a quelli 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 di attività, ogni revisione della definizione di attività rappresenta un'istantanea delle impostazioni per una particolare immagine del contenitore. È simile al modo in cui il container può essere considerato uno snapshot di tutti i componenti necessari per eseguire una particolare versione del codice dell'applicazione.

Crea 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 implementare la nuova revisione della definizione di attività.

- Utilizza ruoli IAM diversi per ogni famiglia di definizioni delle attività.

Definisci ogni definizione delle attività con il proprio ruolo IAM. Implementa questa pratica fornendo 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 dimensioni delle attività di Amazon ECS

Le dimensioni dei container 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 mebibyte. 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 dichiarato. L'utilizzo di un importo superiore a quello 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 container o l'attività può utilizzare. Qualsiasi tentativo di utilizzare una quantità di CPU superiore a questo limite comporta una limitazione (della larghezza di banda della rete). Qualsiasi tentativo di utilizzare più memoria comporta l'interruzione del container.

Scegliere questi valori può essere complesso. Questo perché i valori più adatti all'applicazione dipendono in larga misura dai requisiti di risorse della stessa. 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 senza stato

Per le applicazioni senza stato 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, puoi utilizzare strumenti tradizionali come `ps` o `top` o soluzioni di monitoraggio come Container Insights. CloudWatch

Quando si stabilisce una prenotazione della CPU, bisogna tenere in considerazione come 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 aumentare 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 aumentare e ridurre orizzontalmente più in fretta 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 worker singleton o server di database, la capacità e i costi disponibili rappresentano gli aspetti più importanti da considerare. È consigliabile scegliere la quantità di memoria e CPU in base a ciò che i test di carico indicano necessario per servire il traffico in modo da raggiungere l'obiettivo del livello di servizio. Amazon ECS garantisce che l'applicazione sia collocata su un host con una capacità adeguata.

Rete di attività di Amazon ECS per istanze gestite Amazon ECS

Il comportamento della rete dei processi Amazon ECS eseguiti sulle istanze gestite da Amazon ECS è determinato dalla modalità di rete specificata nella definizione di processo. È necessario specificare la modalità di rete nella definizione dell'attività. Non sarai in grado di eseguire attività sulle istanze gestite da Amazon ECS utilizzando una definizione di attività che non specifica una modalità di rete. Amazon ECS Managed Instances supporta le seguenti modalità di rete, garantendo la compatibilità con le versioni precedenti per la migrazione dei carichi di lavoro da Fargate o Amazon ECS su Amazon: EC2

Modalità di rete	Description	
awsvpc	Ogni attività riceve la propria elastic network interface (ENI) e un IPv4 indirizzo privato. Ciò fornisce le	

Modalità di rete	Description
	stesse proprietà di rete delle EC2 istanze Amazon ed è compatibile con le attività tradizionali di Fargate. Utilizza il trunking ENI per un'elevata densità di attività.
host	Le attività condividono direttamente il namespace di rete dell'host. La rete di container è legata all'istanza host sottostante.

Utilizzo di un VPC in IPv6 modalità -only

In una configurazione IPv6 solo, le attività di Amazon ECS comunicano esclusivamente tramite IPv6. Per configurare VPCs sottoreti per una configurazione IPv6 solo, è necessario aggiungere un blocco IPv6 CIDR al VPC e creare sottoreti che includano solo un blocco CIDR. Per ulteriori informazioni, consulta [Aggiungi IPv6 supporto per il tuo VPC](#) e [crea una sottorete](#) nella Amazon VPC User Guide. È inoltre necessario aggiornare le tabelle di routing con le IPv6 destinazioni e configurare i gruppi di sicurezza con regole. Per ulteriori informazioni, consultare [Configure route tables](#) e [Configure security group rules](#) nella Guida per l'utente di Amazon VPC.

Tieni presente le seguenti considerazioni:

- Puoi aggiornare un servizio Amazon ECS IPv4 -only o dualstack a IPv6 una configurazione -only aggiornando il servizio direttamente per IPv6 utilizzare sottoreti -only o creando un servizio IPv6 solo parallelo e utilizzando le distribuzioni blu-verdi di Amazon ECS per spostare il traffico verso il nuovo servizio. Per ulteriori informazioni sulle implementazioni blu/verdi di Amazon ECS, consultare [Implementazioni Amazon ECS blue/green](#).
- Un servizio IPv6 solo Amazon ECS deve utilizzare sistemi di bilanciamento del carico dualstack con gruppi target. IPv6 Se si sta migrando un servizio di Amazon ECS esistente basato su un Application Load Balancer o un Network Load Balancer, è possibile creare un nuovo sistema di bilanciamento del carico dualstack e spostare il traffico dal vecchio bilanciamento del carico, o aggiornare il tipo di indirizzo IP del bilanciamento del carico esistente.

Per ulteriori informazioni sui Network Load Balancer, consultare [Create a Network Load Balancer](#) e [Update the IP address types for your Network Load Balancer](#) nella Guida per l'utente di Network Load Balancer. Per ulteriori informazioni su Application Load Balancer, consultare [Create an Application Load Balancer](#) e [Update the IP address types for your Application Load Balancer](#) nella Guida per l'utente di Application Load Balancer.

- Per le attività di Amazon ECS in una configurazione IPv6 -only per comunicare IPv4 solo con endpoint, puoi configurare DNS64 e tradurre gli indirizzi NAT64 di rete da a. IPv6 IPv4 Per ulteriori informazioni, consulta [DNS64 e NAT64](#) consulta la Amazon VPC User Guide.
- I carichi di lavoro Amazon ECS in una configurazione IPv6 solo devono utilizzare gli endpoint URI dell'immagine dualstack di Amazon ECR quando estraggono immagini da Amazon ECR. Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

Note

Amazon ECR non supporta gli endpoint VPC con interfaccia dualstack utilizzabili dalle attività in una sola configurazione. IPv6 Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

- Amazon ECS Exec non è supportato in una configurazione di IPv6 sola configurazione.

Assegna un'interfaccia di rete per le attività sulle istanze gestite da Amazon ECS

L'uso della modalità di awsvpc rete in Amazon ECS Managed Instances semplifica la rete di container perché hai un maggiore controllo sul modo in cui le tue applicazioni comunicano tra loro e con gli altri servizi all'interno del tuo. VPCs La modalità di rete awsvpc fornisce inoltre una maggiore

sicurezza per i container in quanto permette di utilizzare i gruppi di sicurezza e gli strumenti di monitoraggio di rete a un livello più granulare all'interno dei processi.

Per impostazione predefinita, ogni istanza di Istanze gestite da Amazon ECS ha un trunk interfaccia di rete elastica (ENI) collegato durante il lancio come ENI principale quando il tipo di istanza supporta il trunking. Per ulteriori informazioni sui tipi di istanza che supportano il trunking ENI, consulta [Supported instances for increased Amazon ECS container network interfaces](#).

 Note

Quando il tipo di istanza scelto non supporta il trunk ENIs, l'istanza verrà avviata con un normale ENI.

Ogni attività eseguita sull'istanza riceve il proprio ENI collegato al trunk ENI, con un indirizzo IP privato principale. Se il tuo VPC è configurato per la modalità dual-stack e utilizzi una sottorete con un blocco IPv6 CIDR, anche l'ENI riceve un indirizzo IPv6. Quando utilizzi una sottorete pubblica, puoi facoltativamente assegnare un indirizzo IP pubblico all'ENI primario di Amazon ECS Managed Instance abilitando l'indirizzamento IPv4 pubblico per la sottorete. Per ulteriori informazioni, consultare [Modify the IP addressing attributes of your subnet](#) nella Guida per l'utente di Amazon VPC. 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 sottoreti VPCs e sulle sottoreti, consulta Come [funziona Amazon VPC nella Amazon VPC User Guide](#)

Le seguenti operazioni utilizzano l'ENI primario collegato all'istanza:

- Download delle immagini: le immagini dei container vengono scaricate da Amazon ECR tramite l'ENI principale.
- Recupero dei segreti: i segreti e le altre credenziali di Secrets Manager vengono recuperati tramite l'ENI primario.
- Caricamenti dei log: i log vengono caricati tramite l'ENI principale. CloudWatch
- Download dei file di ambiente: i file di ambiente vengono scaricati tramite l'ENI principale.

Il traffico delle applicazioni fluisce attraverso l'attività ENI.

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, consultare [Amazon ECS interface VPC endpoints \(AWS PrivateLink\)](#).

La modalità di `awsvpc` rete consente inoltre di sfruttare Amazon VPC Traffic Mirroring per la sicurezza e il monitoraggio del traffico di rete quando si utilizzano tipi di istanze senza trunk collegato. ENIs Per ulteriori informazioni, consultare [What is Traffic Mirroring?](#) nella Guida di Mirroring del traffico Amazon VPC.

Considerazioni per la modalità **awsvpc**

- Le attività richiedono il ruolo collegato ai servizi Amazon ECS per la gestione ENI. Questo ruolo viene creato automaticamente quando crei un cluster o un servizio.
- ENIs Le attività sono gestite da Amazon ECS e non possono essere scollegate o modificate manualmente.
- L'assegnazione di un indirizzo IP pubblico all'attività ENI utilizzando `assignPublicIp` durante l'esecuzione di un'attività autonoma (`RunTask`) o la creazione o l'aggiornamento di un servizio (`CreateService/UpdateService`) non è supportata.
- Quando configuri la rete `awsvpc` a livello di attività, devi utilizzare lo stesso VPC specificato come parte del modello di lancio del provider di capacità di Istanze gestite da Amazon ECS. Puoi utilizzare sottoreti e gruppi di sicurezza diversi da quelli specificati nel modello di lancio.
- Per le attività nella modalità di rete `awsvpc`, utilizzare il tipo di destinazione `ip` durante la configurazione dei gruppi di destinazione del bilanciatore del carico. Amazon ECS gestisce automaticamente la registrazione del gruppo target per le modalità di rete supportate.

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. Pertanto è necessario configurare il routing e la sicurezza nel VPC separatamente IPv4 per 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 di Istanze gestite da Amazon ECS che ospita l'attività sta usando la versione 1.45.0 o successiva dell'agente del container. 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.

Modalità di rete host

Nella modalità host, le attività condividono direttamente il namespace di rete dell'host. La configurazione di rete del container è legata all'istanza host di Istanze gestite da Amazon ECS sottostante che si specifica utilizzando il parametro `networkConfiguration` quando si crea un provider di capacità istanze gestite da Amazon ECS.

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.

Utilizza la modalità `host` solo quando è necessario l'accesso diretto alla rete `host` o durante la migrazione di applicazioni che richiedono l'accesso alla rete a livello di `host`.

Opzioni di task networking di Amazon ECS per EC2

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	Description
<code>awsvpc</code>	Sì	Sì	All'attività viene assegnata una propria elastic network interface (ENI) e un IPv6 indirizzo privato IPv4 o 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'EC2 istanza Amazon che ospita l'attività. Le mappature dinamiche delle porte non possono

Modalità di rete	Contenitori Linux su EC2	Contenitori Windows attivi EC2	Description
			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, consultare [Networking overview](#) nella documentazione di Docker.

Per ulteriori informazioni sulla rete Docker su Windows, consultare [Windows container networking](#) nella documentazione di Microsoft Containers su Windows.

Utilizzo di un VPC in IPv6 modalità -only

In una configurazione IPv6 solo, le attività di Amazon ECS comunicano esclusivamente tramite IPv6. Per configurare VPCs e creare sottoreti per una configurazione IPv6 solo, è necessario aggiungere un blocco IPv6 CIDR al VPC e creare nuove sottoreti che includano solo un blocco CIDR. IPv6 Per ulteriori informazioni, consulta [Aggiungi IPv6 supporto per il tuo VPC](#) e [crea una sottorete](#) nella Amazon VPC User Guide.

È inoltre necessario aggiornare le tabelle di routing con le IPv6 destinazioni e configurare i gruppi di sicurezza con regole. IPv6 Per ulteriori informazioni, consultare [Configure route tables](#) e [Configure security group rules](#) nella Guida per l'utente di Amazon VPC.

Tieni presente le seguenti considerazioni:

- Puoi aggiornare un servizio Amazon ECS IPv4 -only o dualstack a IPv6 una configurazione -only aggiornando il servizio direttamente per IPv6 utilizzare sottoreti -only o creando un servizio IPv6 solo parallelo e utilizzando le distribuzioni blu-verdi di Amazon ECS per spostare il traffico verso il nuovo servizio. Per ulteriori informazioni sulle implementazioni blu/verdi di Amazon ECS, consultare [Implementazioni Amazon ECS blue/green](#).
- Un servizio IPv6 solo Amazon ECS deve utilizzare sistemi di bilanciamento del carico dualstack con gruppi target. IPv6 Se si sta migrando un servizio di Amazon ECS esistente basato su un Application Load Balancer o un Network Load Balancer, è possibile creare un nuovo sistema di bilanciamento del carico dualstack e spostare il traffico dal vecchio bilanciamento del carico, o aggiornare il tipo di indirizzo IP del bilanciamento del carico esistente.

Per ulteriori informazioni sui Network Load Balancer, consultare [Create a Network Load Balancer](#) e [Update the IP address types for your Network Load Balancer](#) nella Guida per l'utente di Network Load Balancer. Per ulteriori informazioni su Application Load Balancer, consultare [Create an Application Load Balancer](#) e [Update the IP address types for your Application Load Balancer](#) nella Guida per l'utente di Application Load Balancer.

- IPv6La configurazione -only non è supportata su. Windows È necessario utilizzare Linux ottimizzato per Amazon ECS AMIs per eseguire attività in una configurazione solo IPv6. Per ulteriori informazioni su Linux ottimizzato per Amazon ECS AMIs, consulta. [Linux ottimizzato per Amazon ECS AMIs](#)
- Quando avvii un'istanza di contenitore per eseguire attività in una configurazione IPv6 -only, devi impostare un IPv6 indirizzo primario per l'istanza utilizzando il parametro. `--enable-primary-ipv6` EC2

 Note

Senza un IPv6 indirizzo primario, le attività in esecuzione sull'istanza del contenitore nelle modalità di rete host o bridge non verranno registrate con i sistemi di bilanciamento del carico o con. AWS Cloud Map

Per ulteriori informazioni sull'esecuzione delle `--enable-primary-ipv6` istanze Amazon, consulta EC2 [run-instances](#) nel Command Reference.AWS CLI

Per ulteriori informazioni sul lancio di istanze di container utilizzando il, consulta. Console di gestione AWS [Avvio di un'istanza di container Linux di Amazon ECS](#)

- Per impostazione predefinita, l'agente container Amazon ECS cercherà di rilevare la compatibilità dell'istanza del contenitore per una configurazione IPv6 solo esaminando le impostazioni predefinite IPv4 e IPv6 le rotte dell'istanza. Per ignorare questo comportamento, puoi impostare il parametro `ECS_INSTANCE_IP_COMPATIBILITY` su `ipv4` o `ipv6` nel file dell'istanza `/etc/ecs/ecs.config`.
- Le attività devono utilizzare la versione 1.99.1 o successiva dell'agente del container.. Per informazioni sul controllo della versione dell'agente utilizzata dall'istanza e sull'aggiornamento, se necessario, consultare [Aggiornamento dell'agente del container Amazon ECS](#).
- Per le attività di Amazon ECS in una configurazione IPv6 -only per comunicare IPv4 solo con endpoint, puoi configurare DNS64 e tradurre gli indirizzi NAT64 di rete da a. IPv6 IPv4 Per ulteriori informazioni, consulta [DNS64 e NAT64](#) consulta la Amazon VPC User Guide.
- I carichi di lavoro Amazon ECS in una configurazione IPv6 solo devono utilizzare gli endpoint URI dell'immagine dualstack di Amazon ECR quando estraggono immagini da Amazon ECR. Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

Note

Amazon ECR non supporta gli endpoint VPC con interfaccia dualstack utilizzabili dalle attività in una sola configurazione. IPv6 Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

- Amazon ECS Exec non è supportato in una configurazione di IPv6 sola configurazione.

Regioni AWS che supporta la modalità IPv6 solo per Amazon ECS

Puoi eseguire attività in una configurazione IPv6 solo nelle seguenti AWS regioni in cui Amazon ECS è disponibile:

- Stati Uniti orientali (Ohio)

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacific (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Osaka)
- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- Cina (Ningxia)
- Europa (Francoforte)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)

- AWS GovCloud (Stati Uniti occidentali)

Assegna un'interfaccia di rete per un'attività su 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 di rete `awsvpc` fornisce inoltre una maggiore sicurezza per i container in quanto permette di utilizzare i gruppi di sicurezza e gli strumenti di monitoraggio di rete a un livello più granulare all'interno dei processi. 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`.

L'interfaccia di rete elastica (ENI) dell'attività 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.

Quando è necessaria una maggiore densità ENI, utilizzare le impostazioni dell'account `awsvpcTrunking`. Inoltre, Amazon ECS crea e collega un'interfaccia di rete "trunk" per l'istanza di container. 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 sulle impostazioni dell'account `awsvpcTrunking`, consultare [Prerequisiti](#).

Specificare `awsvpc` nel parametro `networkMode` della definizione dell'attività. Per ulteriori informazioni, consulta [Modalità di rete](#).

Pertanto, quando si esegue un'attività o si crea un servizio, utilizzare un parametro `networkConfiguration` che include una o più sottoreti in cui inserire le attività e uno o più gruppi

di sicurezza da collegare all'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 si utilizza un'istanza p5.48xlarge in modalità aws-vpc, non è possibile 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 Console di gestione AWS. 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'`awsVpcTrunkingAccount` 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 delle istanze di container Linux di Amazon ECS](#).
- 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 rete `awsvpc` 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 rete `awsvpc`, le attività che utilizzano la modalità di rete `awsvpc` non vengono avviate sulle istanze di Windows 2016 Server. Piuttosto, vengono avviati su istanze che supportano la 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 rete `awsvpc`.
- Le attività e i servizi che utilizzano la modalità di rete `awsvpc` 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 Console di gestione AWS. 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 “Bootstrap delle istanze di 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 ENIs 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 ENIs 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 ENIs 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 ENIs ciò che è collegato alla tua istanza Amazon EC2 Windows. 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 di Amazon EC2 Windows devono essere registrate con la `ecs.capability.task-eni` capacità di essere prese in considerazione per il posizionamento delle attività in modalità `awsvpc` rete.
- Non puoi modificare o scollegare ENIs 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 `awsvpc` rete 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. IPv6 e IPv4 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.

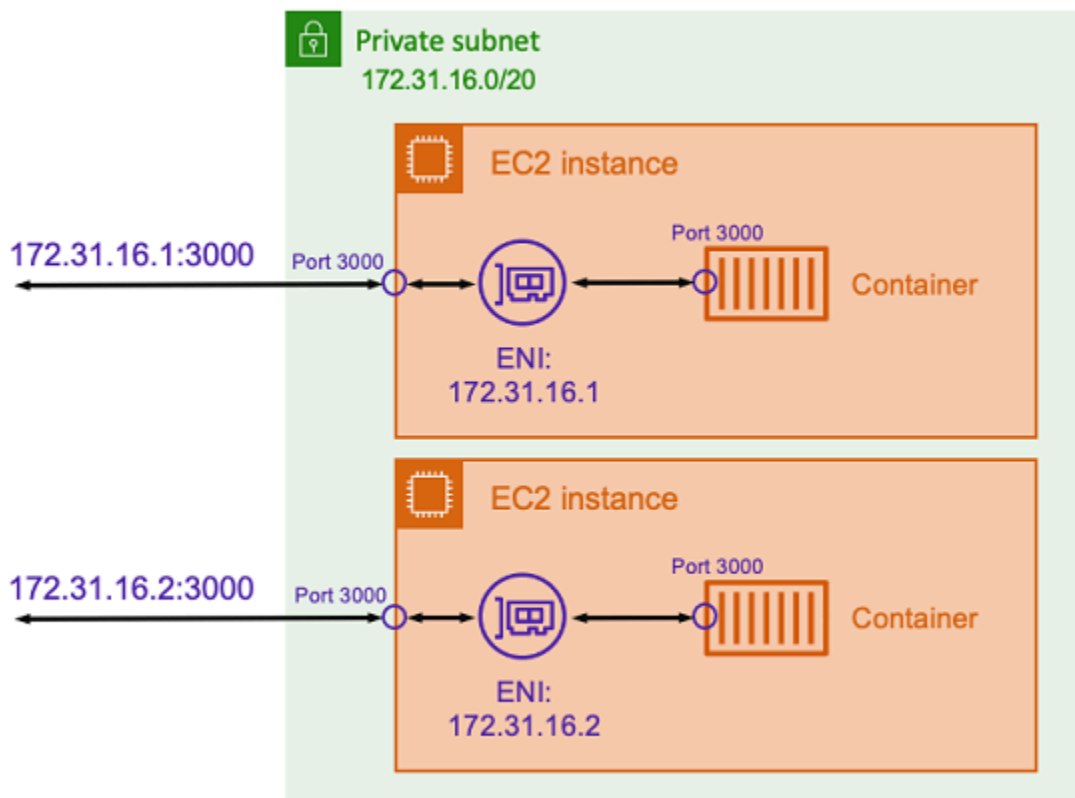
Alle attività di Amazon ECS viene assegnato un IPv6 indirizzo se vengono soddisfatte le seguenti condizioni:

- 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' EC2istanza

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.

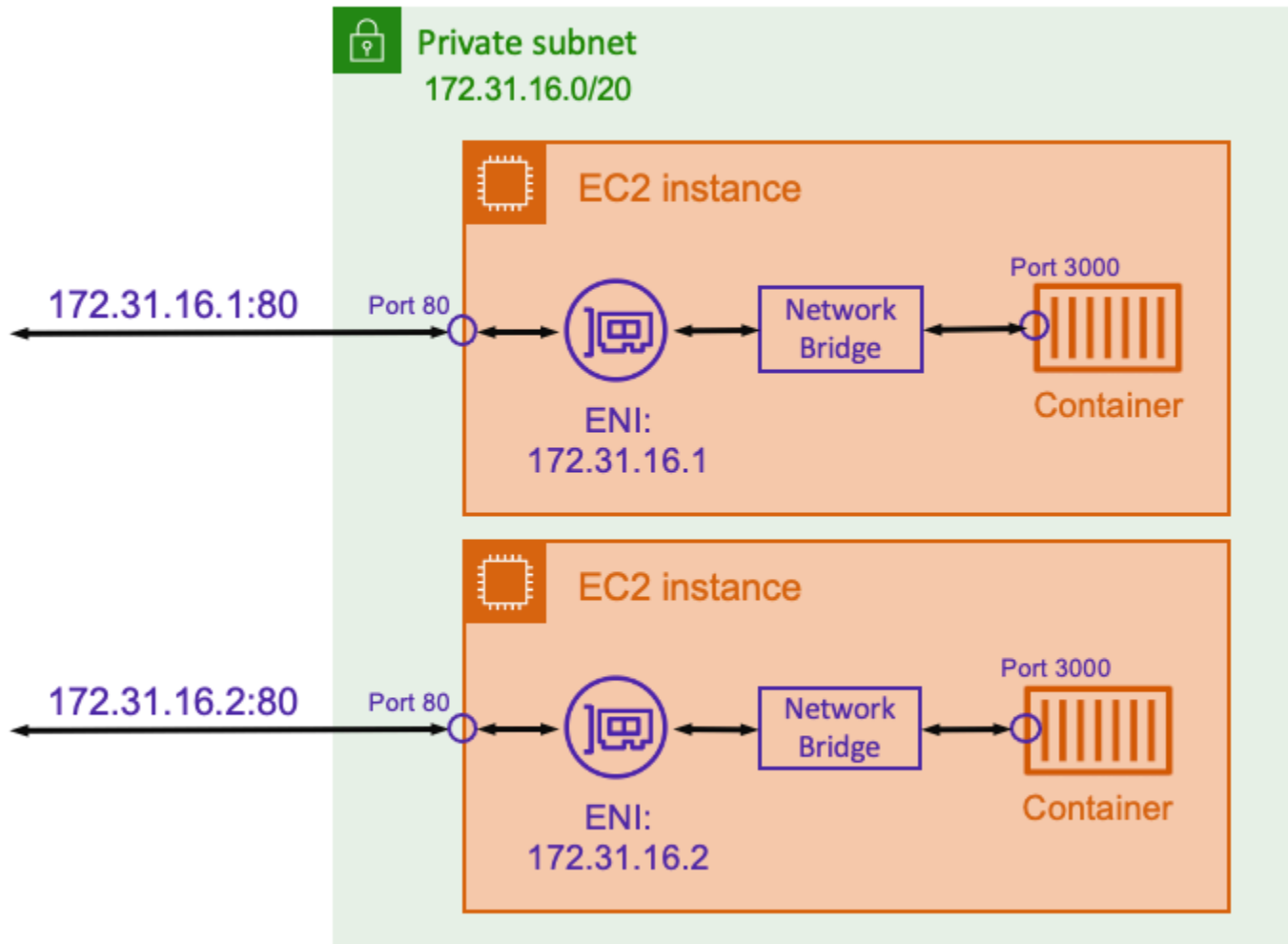
L'utilizzo di questa modalità di rete presenta notevoli svantaggi. È possibile creare solo una singola istanza di un'attività su ciascun host, Questo perché solo la prima attività può collegarsi alla porta richiesta sull' EC2 istanza Amazon. Inoltre, non è possibile rimappare una porta del container 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.

Utilizzare 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.

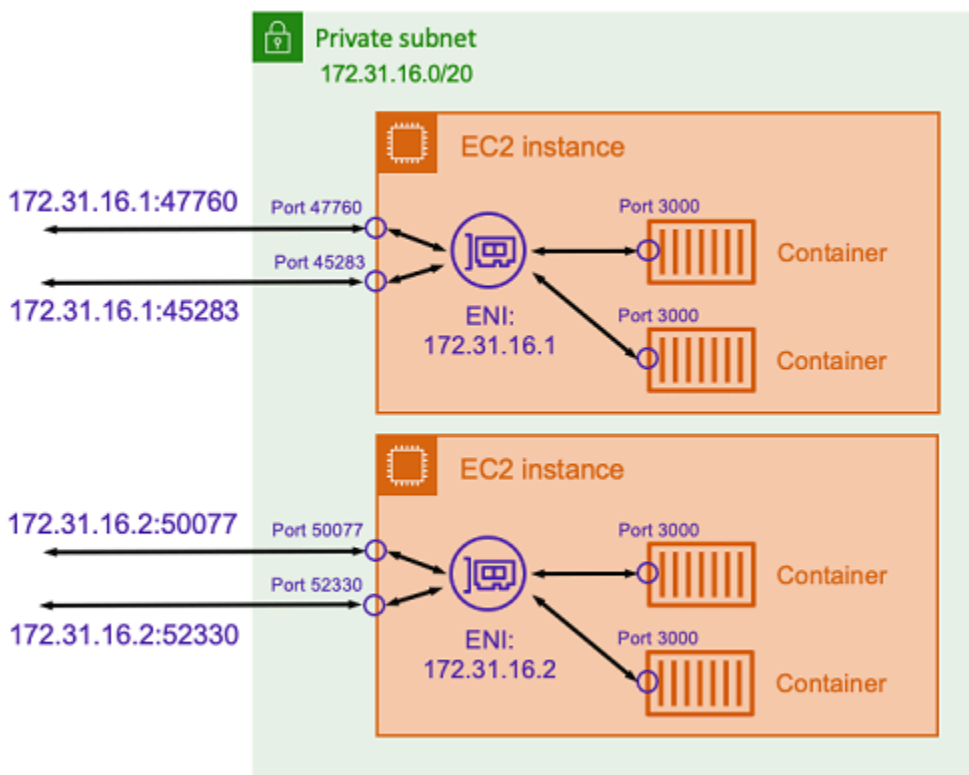
Con la modalità `bridge`, stai utilizzando un bridge di rete virtuale per creare un livello tra l'host e la rete del container. In questo modo, puoi creare mappature delle porte che rimappano una porta host su una porta container. Le mappature possono essere statiche o dinamiche.



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 si specifica 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.

Configurazione della modalità di rete bridge per IPv6 carichi di lavoro solo

Per configurare la bridge modalità di comunicazione IPv6, è necessario aggiornare Docker le impostazioni del demone. Aggiornare `/etc/docker/daemon.json` come segue:

```
{
  "ipv6": true,
  "fixed-cidr-v6": "2001:db8:1::/64",
  "ip6tables": true,
  "experimental": true
}
```

Dopo aver aggiornato le impostazioni del daemon Docker, sarà necessario riavviarlo.

Note

Quando aggiorni e riavvii il daemon, Docker abilita l'IPv6 in tutto sull'istanza, il che può comportare la perdita delle route predefinite sulle istanze che utilizzano un'AMI Amazon Linux 2. Per evitare ciò, usa il seguente comando per aggiungere una route predefinita tramite il gateway della sottorete. IPv6

```
ip route add default via FE80:EC2::1 dev eth0 metric 100
```

Opzioni di rete di attività di Amazon ECS per 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, consultare [Amazon ECS interface VPC endpoints \(AWS PrivateLink\)](#).

Per esempi di come utilizzare la `NetworkConfiguration` risorsa con CloudFormation, consulta [the section called “Modelli di esempio 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
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 Console di gestione AWS. 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 per Windows. Le attività avviate in sottoreti dual-stack ricevono un indirizzo e un IPv4 indirizzo. IPv6 Le attività avviate nelle sottoreti IPv6 -only ricevono solo un indirizzo. IPv6
- Per le attività che utilizzano la versione della piattaforma 1.4.0 o successiva per Linux o 1.0.0 Windows, l'attività ENIs supporta 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.

- I servizi con attività che utilizzano 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. 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 account `dualStackIPv6` di Amazon ECS è attivata (`enabled`) per l'IAM principale che avvia le attività nella regione in cui le stai avviando. Questa impostazione può essere modificata solo utilizzando l'API o AWS CLI. Esiste la possibilità di attivare questa impostazione per un IAM principale specifico sul proprio account o per l'intero account attivando le impostazioni predefinite. 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.

Per le attività di Amazon ECS su Fargate eseguite in un VPC in modalità dual-stack, per comunicare con i servizi di dipendenza utilizzati nel processo di avvio delle attività come ECR, SSM, SecretManager e, la tabella di routing della sottorete pubblica richiede il routing (0.0.0.0/0) verso un gateway Internet e la tabella di routing della sottorete privata IPv4 richiede il routing (0.0.0.0/0) verso

un gateway NAT. IPv4 Per ulteriori informazioni, consulta i [gateway Internet e i gateway NAT nella Amazon VPC User Guide](#).

Per esempi su come configurare un VPC dual-stack, consulta [Esempio](#) di configurazione VPC dual-stack.

Utilizzo di un VPC in IPv6 modalità -only

In una configurazione IPv6 solo, le attività di Amazon ECS comunicano esclusivamente tramite IPv6. Per configurare VPCs sottoreti per una configurazione IPv6 solo, è necessario aggiungere un blocco IPv6 CIDR al VPC e creare sottoreti che includano solo un blocco CIDR. Per ulteriori informazioni, consulta [Aggiungi IPv6 supporto per il tuo VPC](#) e [crea una sottorete](#) nella Amazon VPC User Guide. È inoltre necessario aggiornare le tabelle di routing con le IPv6 destinazioni e configurare i gruppi di sicurezza con regole. Per ulteriori informazioni, consultare [Configure route tables](#) e [Configure security group rules](#) nella Guida per l'utente di Amazon VPC.

Tieni presente le seguenti considerazioni:

- Puoi aggiornare un servizio Amazon ECS IPv4 -only o dualstack a IPv6 una configurazione -only aggiornando il servizio direttamente per IPv6 utilizzare sottoreti -only o creando un servizio IPv6 solo parallelo e utilizzando le distribuzioni blu-verdi di Amazon ECS per spostare il traffico verso il nuovo servizio. Per ulteriori informazioni sulle implementazioni blu/verdi di Amazon ECS, consultare [Implementazioni Amazon ECS blue/green](#).
- Un servizio IPv6 solo Amazon ECS deve utilizzare sistemi di bilanciamento del carico dualstack con gruppi target. IPv6 Se si sta migrando un servizio di Amazon ECS esistente basato su un Application Load Balancer o un Network Load Balancer, è possibile creare un nuovo sistema di bilanciamento del carico dualstack e spostare il traffico dal vecchio bilanciamento del carico, o aggiornare il tipo di indirizzo IP del bilanciamento del carico esistente.

Per ulteriori informazioni sui Network Load Balancer, consultare [Create a Network Load Balancer](#) e [Update the IP address types for your Network Load Balancer](#) nella Guida per l'utente di Network Load Balancer. Per ulteriori informazioni su Application Load Balancer, consultare [Create an Application Load Balancer](#) e [Update the IP address types for your Application Load Balancer](#) nella Guida per l'utente di Application Load Balancer.

- IPv6La configurazione -only non è supportata su Windows
- Per le attività di Amazon ECS in una configurazione IPv6 -only per comunicare IPv4 solo con endpoint, puoi configurare DNS64 e tradurre gli indirizzi NAT64 di rete da a. IPv6 IPv4 Per ulteriori informazioni, consulta [DNS64 e NAT64](#) consulta la Amazon VPC User Guide.

- IPv6-la configurazione è supportata solo sulla 1.4.0 versione della piattaforma Fargate o successiva.
- I carichi di lavoro Amazon ECS in una configurazione IPv6 solo devono utilizzare gli endpoint URI dell'immagine dualstack di Amazon ECR quando estraggono immagini da Amazon ECR. Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

Note

Amazon ECR non supporta gli endpoint VPC con interfaccia dualstack utilizzabili dalle attività in una sola configurazione. IPv6 Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.

- Amazon ECS Exec non è supportato in una configurazione di IPv6 sola configurazione.
- Amazon CloudWatch non supporta un endpoint FIPS dualstack che può essere utilizzato per monitorare le attività di Amazon ECS in IPv6 una sola configurazione che utilizza la conformità FIPS-140. Per ulteriori informazioni sul FIPS-140, consultare [AWS Fargate Standard federale per l'elaborazione delle informazioni \(FIPS-140\)](#).

Regioni AWS che supporta la modalità IPv6 solo per Amazon ECS

Puoi eseguire attività in una configurazione IPv6 solo nelle seguenti configurazioni in Regioni AWS cui Amazon ECS è disponibile:

- Stati Uniti orientali (Ohio)
- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Africa (Città del Capo)
- Asia Pacifico (Hong Kong)
- Asia Pacifico (Hyderabad)
- Asia Pacifico (Giacarta)
- Asia Pacifico (Melbourne)
- Asia Pacifico (Mumbai)
- Asia Pacifico (Osaka)

- Asia Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Canada occidentale (Calgary)
- Cina (Pechino)
- Cina (Ningxia)
- Europa (Francoforte)
- Europa (Londra)
- Europa (Milano)
- Europa (Parigi)
- Europa (Spagna)
- Israele (Tel Aviv)
- Medio Oriente (Bahrein)
- Medio Oriente (Emirati Arabi Uniti)
- Sud America (San Paolo)
- AWS GovCloud (Stati Uniti orientali)
- AWS GovCloud (Stati Uniti occidentali)

Opzioni di archiviazione 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 container:

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
Amazon Elastic Block Store (Amazon EBS)	Istanze gestite Fargate, EC2 Amazon, Amazon ECS	Linux, Windows (EC2 solo su Amazon)	Può essere mantenuto se collegato a un'attività autonoma.	I volumi Amazon EBS forniscono un'archiviazione a blocchi conveniente,

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
			Effimero se associato a un'attività gestita da un servizio.	durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. I casi d'uso più comuni includono carichi di lavoro transazionali come database, desktop virtuali e volumi root, nonché carichi di lavoro ad alta intensità di throughput come l'elaborazione dei log e i carichi di lavoro ETL. Per ulteriori informazioni, consulta Usare i volumi Amazon EBS con Amazon ECS .

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
Amazon Elastic File System (Amazon EFS)	Istanze gestite Fargate, EC2 Amazon, Amazon ECS	Linux	Persistente	I volumi di Amazon EFS offrono uno spazio di archiviazione di file condiviso, semplice, scalabile e persistente da utilizzare con i processi Amazon ECS che aumentano e si riducono automaticamente con l'aggiunta e la rimozione di 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-

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
				after-write I casi d'uso più comuni includono carichi di lavoro come analisi dei dati, elaborazione multimediale, gestione dei contenuti e server Web. Per ulteriori informazioni, consulta Usare i volumi Amazon EFS con Amazon ECS .

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di 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	Capacità supportata	Sistemi operativi supportati	Persistenza di 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	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
Amazon FSx per NetApp ONTAP	Amazon EC2	Linux	Persistente	I volumi Amazon FSx for NetApp ONTAP forniscono file system NetApp ONTAP completamente gestiti che puoi utilizzare e per eseguire il provisioning di attività Linux che richiedono o uno storage di file condiviso persistente, ad alte prestazioni e ricco di funzionalità. Amazon FSx for NetApp ONTAP supporta i protocolli NFS e SMB e fornisce funzionalità di livello aziendale come istantanee, clonazione e deduplicazione dei dati. I casi d'uso più comuni includono carichi di lavoro di elaborazione ad

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
				alte prestazioni, repository di contenuti e applicazioni che richiedono uno storage condiviso conforme a POSIX. Per ulteriori informazioni, consulta Montaggio dei file system Amazon FSx for NetApp ONTAP dai contenitori Amazon ECS .

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
Volumi Docker	Amazon EC2	Windows, Linux	Persistente	I volumi Docker sono una funzionalità del runtime del container Docker che consente ai container di mantenere i dati in modo persistente montando una directory dal file system dell'host . I driver del volume docker (detti anche plugin) vengono utilizzati per integrare i volumi dei container ai sistemi di archiviazione esterni. I volumi Docker possono essere gestiti da driver di terzi o dal driver integrato local. I casi d'uso comuni dei volumi Docker includono la fornitura di

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
				volumi di dati persistenti o la condivisione di volumi in posizioni diverse su container diversi sulla stessa istanza di container. Per ulteriori informazioni, consulta Utilizzo dei volumi Docker con Amazon ECS .

Volume di dati	Capacità supportata	Sistemi operativi supportati	Persistenza di archiviazione	Casi d'uso
Montaggi vincolati	Istanze gestite Fargate, EC2 Amazon, Amazon ECS	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 montaggi vincolati includono la condivisione di un volume da un container di origine con altri container nella stessa attività, o il montaggio di un volume host o di un volume vuoto in uno o più container. Per ulteriori informazioni, consulta Utilizzo di montaggio vincolato con Amazon ECS .

Usare i volumi Amazon EBS con Amazon ECS

I volumi Amazon Elastic Block Store (Amazon EBS) forniscono un'archiviazione 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. Per ulteriori informazioni sui volumi Amazon EBS, consultare [Amazon EBS volumes](#) nella Guida per l'utente di Amazon EBS.

I volumi Amazon EBS collegati alle attività di Amazon ECS sono gestiti da Amazon ECS per tuo conto. Durante l'avvio di un'attività autonoma, è possibile 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. È possibile configurare nuovi volumi vuoti per gli allegati oppure utilizzare snapshot per caricare dati da volumi esistenti.

Note

Quando si utilizzano gli snapshot per configurare i volumi, è possibile specificare un intervallo `volumeInitializationRate`, in MiB/s, in corrispondenza del quale i dati vengono recuperati dallo snapshot per creare volumi completamente inizializzati in un periodo di tempo prevedibile. Per ulteriori informazioni sull'inizializzazione dei volumi, consultare [Initialize Amazon EBS volumes](#) nella Guida per l'utente di Amazon EBS. Per ulteriori informazioni sulla configurazione dei volumi Amazon EBS, consultare [Rimanda la configurazione del volume all'ora di avvio nella definizione di un'attività di Amazon ECS](#) e [Specificare la configurazione del volume Amazon EBS nell'implementazione di Amazon ECS](#).

La configurazione del volume viene posticipata al momento dell'avvio utilizzando il parametro `configuredAtLaunch` nella definizione dell'attività. Fornendo la configurazione del volume al momento dell'avvio anziché nella definizione dell'attività, è possibile creare 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 maggiore throughput durante l'implementazione per i carichi di lavoro di produzione rispetto agli ambienti pre-produzione.

I volumi Amazon EBS collegati alle attività possono essere crittografati con AWS Key Management Service (AWS KMS) chiavi per proteggere i dati. Per ulteriori informazioni, consultare [Crittografia dei dati archiviati nei volumi Amazon EBS allegati alle attività Amazon ECS](#).

Per monitorare le prestazioni del volume, puoi anche utilizzare i CloudWatch parametri di Amazon. Per ulteriori informazioni sulle metriche di Amazon ECS per i volumi Amazon EBS, consultare [Metriche di Amazon ECS CloudWatch](#) e [Amazon ECS Container Insights metrics](#).

Il collegamento di un volume Amazon EBS a un'attività è supportato in tutte le aziende commerciali e nelle [Regioni AWS](#) in Cina, che supportano Amazon ECS.

Sistemi operativi e capacità supportati

La tabella seguente fornisce il sistema operativo e le configurazioni di capacità supportate.

Capacity	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 supportata
EC2	I volumi Amazon EBS sono supportati per le attività ospitate su istanze Nitro basate con Amazon Machine Images () ottimizzato per Amazon ECS. AMIs Per ulteriori informazioni sui tipi di istanze, consulta Tipi di istanza nella Amazon EC2 User Guide. I volumi Amazon EBS sono supportati su AMI 20231219 ottimizzate per ECS o versioni successive. Per ulteriori informazioni, consulta	Attività ospitate su istanze Nitro basate con 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 AMI 20241017 ottimizzate per ECS o versioni successive. Per ulteriori informazioni, consultare Retrieving Amazon ECS-Optimized Windows AMI metadata.

Capacity	Linux	Windows
	Retrieving Amazon ECS-Optimized AMI metadata.	
Istanze gestite da Amazon ECS	I volumi Amazon EBS sono supportati per le attività ospitate su Istanze gestite da Amazon ECS su Linux.	Non supportata

Considerazioni

Quando si usano i volumi Amazon EBS, si tenga presente quanto segue:

- Non è possibile configurare i volumi Amazon EBS per il collegamento alle attività Fargate Amazon ECS nella zona di disponibilità use1 - az3.
- 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 di Amazon ECS è necessario quando si crea un servizio o un'attività autonoma che consiste nella configurazione di un volume al momento dell'implementazione. È possibile allegare la policy IAM AmazonECSInfrastructureRolePolicyForVolumes gestita da AWS al ruolo, oppure è possibile utilizzare la policy gestita come guida per creare e allegare la policy con autorizzazioni che soddisfino le esigenze specifiche. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- È possibile allegare al massimo un volume Amazon EBS a ciascuna attività Amazon ECS e deve trattarsi di un nuovo volume. Non è possibile collegare un volume Amazon EBS esistente a un'attività. Tuttavia, è possibile configurare un nuovo volume Amazon EBS al momento dell'implementazione utilizzando lo snapshot di un volume esistente.
- Per utilizzare i volumi Amazon EBS con i servizi Amazon ECS, il comando dell'implementazione deve essere ECS. Quando si utilizza questo controller di blue/green distribuzione sono supportate sia le strategie di rolling che quelle di distribuzione.
- Affinché un contenitore della tua attività possa scrivere sul volume Amazon EBS montato, deve disporre delle autorizzazioni di file system appropriate. Quando specifichi un utente non root nella definizione del contenitore, Amazon ECS configura automaticamente il volume con autorizzazioni

basate sul gruppo che consentono all'utente specificato di leggere e scrivere sul volume. Se non viene specificato alcun utente, il contenitore funziona come root e ha pieno accesso al volume.

- Amazon ECS allega automaticamente i tag riservati AmazonECSCreated e AmazonECSManaged a un volume Amazon EBS. Se si rimuovono questi tag dal volume, Amazon ECS non sarà in grado di gestire il volume per conto dell'utente. Per ulteriori informazioni sulla menzione di volumi Amazon EBS, consultare [Tagging Amazon EBS volumes](#). Per ulteriori informazioni sull'etichettare le risorse di Amazon ECS, consultare [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 della stessa.
- Non è possibile configurare i volumi Amazon EBS per il collegamento alle attività di Amazon ECS che vengono eseguite su AWS Outposts.

Comportamento degli utenti non root

Quando specifichi un utente non root nella definizione del contenitore, Amazon ECS configura automaticamente il volume Amazon EBS con autorizzazioni basate sul gruppo che consentono all'utente specificato di leggere e scrivere sul volume. Il volume è montato con le seguenti caratteristiche:

- Il volume è di proprietà dell'utente root e del gruppo root.
- Le autorizzazioni di gruppo sono impostate per consentire l'accesso in lettura e scrittura.
- L'utente non root viene aggiunto al gruppo appropriato per accedere al volume.

Segui queste best practice quando usi volumi Amazon EBS con contenitori non root:

- Usa user IDs (UIDs) e group IDs (GIDs) coerenti tra le immagini del contenitore per garantire autorizzazioni coerenti.
- Precrea le directory dei punti di montaggio nell'immagine del contenitore e imposta la proprietà e le autorizzazioni appropriate.
- Testa i tuoi contenitori con volumi Amazon EBS in un ambiente di sviluppo per confermare che le autorizzazioni del file system funzionino come previsto.
- Se più contenitori per la stessa attività condividono un volume, assicurati che utilizzino un volume compatibile UIDs/GIDs o installino il volume con aspettative di accesso coerenti.

Rimanda la configurazione del volume all'ora di avvio nella definizione di un'attività di Amazon ECS

Per configurare un volume Amazon EBS per gli allegati dell'attività, è necessario specificare il punto di montaggio nella definizione di attività e dare un nome al volume. È inoltre necessario impostare `configuredAtLaunch` su `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 l'implementazione.

Per registrare la definizione dell'attività utilizzando AWS Command Line Interface (AWS CLI), salvate il modello come file JSON, quindi passate il file come input per il [register-task-definition](#) comando.

Per creare e registrare una definizione di attività utilizzando il Console di gestione AWS, vedere [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#).

La definizione di attività seguente illustra la sintassi degli oggetti `mountPoints` e `volumes` nella definizione di attività. Per ulteriori informazioni sui parametri per la definizione di attività, consultare [Parametri di definizione di attività Amazon ECS per Fargate](#). Per utilizzare questo comando, sostituire *user input placeholders* con le proprie 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

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai `Volumes` nella creazione container dell'API Docker e l'opzione `--volume` per `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

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, lascia il valore predefinito di `false`.

`name`

─Tipo: stringa

Obbligatorio: no

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (-) e trattini bassi (_). Nel parametro `sourceVolume` dell'oggetto `mountPoints` della definizione del container viene fatto riferimento a questo nome.

`configuredAtLaunch`

Tipo: Booleano

Obbligatorio: sì, quando vuoi collegare un volume EBS direttamente a un'attività.

Specificare 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 di 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.

Crittografia dei dati archiviati nei volumi Amazon EBS allegati alle attività 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 quando inattivi sul volume
- I/O del disco
- Snapshot creati dal volume
- Nuovi volumi creati da snapshot crittografati

I volumi Amazon EBS collegati alle attività possono essere crittografati utilizzando un Chiave gestita da AWS predefinito con un alias `alias/aws/ebs` 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, seguire le fasi descritte in [Creazione delle chiavi KMS per la crittografia simmetrica](#) nella Guida per gli sviluppatori di 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 di Amazon EBS e la crittografia per impostazione predefinita, consultare [Amazon EBS encryption](#) nella Guida per l'utente di Amazon EBS.

Comportamento di Istanze gestite da Amazon ECS

È possibile crittografare i volumi Amazon EBS abilitando la crittografia, utilizzando la crittografia per impostazione predefinita o abilitando la crittografia al momento della creazione di un volume che si desidera crittografare. Per informazioni su come abilitare la crittografia per impostazione predefinita (a livello di account), consultare [Encryption by default](#) nella Guida per l'utente di Amazon EBS.

È possibile configurare qualsiasi combinazione di questi tasti. 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 account.
2. La chiave KMS specificata a livello di account. 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 account 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 si imposta `encrypted` a `false` nella configurazione del volume, non si specifica alcuna chiave KMS a livello di account e si abilita la crittografia Amazon EBS per impostazione predefinita, il volume verrà comunque crittografato con la chiave specificata per la crittografia Amazon EBS per impostazione predefinita.

Comportamento di Istanze non gestite da Amazon ECS

È possibile anche configurare la crittografia a livello di cluster di Amazon ECS per l'archiviazione gestita da Amazon ECS quando si crea o aggiorna un cluster. La crittografia a livello di cluster ha effetto a livello di attività e può essere utilizzata per crittografare i volumi di Amazon EBS collegati a ciascuna attività in esecuzione in un cluster specifico utilizzando la chiave KMS specificata. Per ulteriori informazioni sulla configurazione della crittografia a livello di cluster per ogni attività, consulta il riferimento [ManagedStorageConfiguration](#) alle API di Amazon ECS.

È possibile configurare qualsiasi combinazione di questi tasti. 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 si imposta `encrypted` a `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.

Policy delle chiavi KMS gestite dal cliente

Per crittografare un volume EBS collegato all'attività utilizzando una chiave gestita dal cliente, è necessario configurare la policy della chiave KMS per garantire che il ruolo IAM utilizzato per la configurazione del volume disponga delle autorizzazioni necessarie per utilizzare la chiave. La policy della chiave deve includere le autorizzazioni `kms:CreateGrant` e `kms:GenerateDataKey*`. Le autorizzazioni `kms:ReEncryptTo` e `kms:ReEncryptFrom` sono necessarie per crittografare i volumi creati utilizzando gli snapshot. Se desideri configurare e crittografare solo nuovi volumi vuoti da allegare, puoi escludere le autorizzazioni `kms:ReEncryptTo` e `kms:ReEncryptFrom`.

Il seguente frammento di codice JSON mostra le dichiarazioni della policy della chiave che puoi allegare alla policy della chiave KMS. L'utilizzo di queste istruzioni consentirà ad Amazon ECS di utilizzare la chiave per crittografare il volume EBS. Per usare gli esempi di dichiarazioni della policy, sostituire *user input placeholders* con le proprie informazioni. Come sempre, configurare solo le autorizzazioni necessarie.

```
{
```

```

    "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 maggiori informazioni sulle autorizzazioni e le policy della chiave, consultare [Key policies in AWS KMS](#) e [permissions AWS KMS](#) nella Guida per sviluppatori di AWS KMS . Per la risoluzione dei problemi degli allegati dei volumi EBS relativi alle autorizzazioni chiave, consultare [Risoluzione dei problemi relativi ai volumi di Amazon EBS collegati alle attività di Amazon ECS](#) .

Specificare la configurazione del volume Amazon EBS nell'implementazione di Amazon ECS

Dopo aver registrato una definizione di attività con il parametro `configuredAtLaunch` impostato su `true`, puoi configurare un volume Amazon EBS durante l'implementazione quando esegui un'attività autonoma o quando crei o aggiorni un servizio. Per ulteriori informazioni su come posticipare la configurazione del volume all'ora di avvio utilizzando il parametro `configuredAtLaunch`, consultare [Rimanda la configurazione del volume all'ora di avvio nella definizione di un'attività di Amazon ECS](#).

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 container specifica. Questo comando non è applicabile alle attività 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, deve disporre delle autorizzazioni di file system appropriate. Quando specifichi un utente non root nella definizione del contenitore, Amazon ECS configura automaticamente il volume con autorizzazioni basate sul gruppo che consentono all'utente specificato di leggere e scrivere sul volume. Se non viene specificato alcun utente, il contenitore funziona come root e ha pieno accesso al volume.

È possibile anche configurare un volume Amazon EBS utilizzando Console di gestione AWS. Per ulteriori informazioni, consultare [Esecuzione di un'applicazione come attività Amazon ECS](#), [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#) e [Aggiornamento di un servizio Amazon ECS](#).

Il seguente frammento di codice JSON mostra tutti i parametri di un volume Amazon EBS che può essere configurato al momento dell'implementazione. Per utilizzare questi parametri per la configurazione del volume, sostituire *user input placeholders* con le proprie informazioni. Per ulteriori informazioni sui parametri di configurazione, consultare [Volume configurations](#).

```
"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

Assicurarsi che `volumeName`, specificato nella configurazione, sia uguale a `volumeName`, specificato nella definizione dell'attività.

Per informazioni sulla verifica dello stato del volume dell'allegato, consultare [Risoluzione dei problemi relativi ai volumi di Amazon EBS collegati 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 frammenti di esempio 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 `--cli-input-json file://filename` comandi. AWS CLI Sostituisci *user input placeholders* con le informazioni appropriate.

Configurare un volume per un'attività autonoma

Il seguente frammento mostra la sintassi per configurare i volumi di Amazon EBS per l'allegato a un'attività autonoma. Il seguente frammento di codice JSON mostra la sintassi per la configurazione delle impostazioni `volumeType`, `sizeInGiB`, `encrypted` e `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:11112223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
      }
    }
  ]
}
```

```
}
```

Configurare un volume al momento della creazione del servizio

Il seguente frammento mostra la sintassi per configurare i volumi di Amazon EBS per l'allegato alle attività gestite da un servizio. I volumi provengono dallo snapshot specificato utilizzando il parametro `snapshotId` a una velocità di 200 MiB/s. La configurazione specificata nel file JSON viene utilizzata per creare e allegare un volume EBS a ogni attività gestita da un 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 al momento dell'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 dell'attività con `configuredAtLaunch` impostato su `true`. Il seguente frammento di codice JSON mostra la sintassi per la configurazione delle impostazioni `volumeType`, `sizeInGiB`, `throughput`, `iops` e `filesystemType`. Questa configurazione viene utilizzata per creare e allegare un volume EBS a ogni attività gestita da un 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"
    }
  }
]
}

```

Configurare 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` impostato su `false` o una definizione di attività senza il parametro `configuredAtLaunch`. È inoltre necessario fornire un oggetto `volumeConfigurations` vuoto.

```

{
  "cluster": "mycluster",
  "taskDefinition": "mytaskdef",
  "service": "mysvc",
  "desiredCount": 2,
  "volumeConfigurations": []
}

```

Policy di terminazione per i volumi di Amazon EBS

Quando si termina un'attività Amazon ECS, questo utilizza il valore `deleteOnTermination` 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, è possibile modificare questa impostazione per preservare 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 della stessa.

Tag volumi Amazon EBS

È possibile etichettare i volumi Amazon EBS utilizzando l'oggetto `tagSpecifications`. Utilizzando l'oggetto, è possibile 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 collegare a un volume è pari a 50.

 Important

Amazon ECS allega automaticamente i tag `AmazonECSManaged` e `AmazonECSCreated` riservati a un volume Amazon EBS. Ciò significa che è possibile 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 si desidera aggiungere tag gestiti da Amazon ECS al proprio volume, è necessario impostare `enableECSManagedTags` su `true` nella propria chiamata `UpdateService`, `CreateService`, `RunTask` o `StartTask`. Se si attivano i tag gestiti da Amazon ECS, quest'ultimo etichetterà automaticamente il volume con informazioni su cluster e servizi (`aws:ecs:clusterName` e `aws:ecs:serviceName`). Per ulteriori informazioni sull'etichettare le risorse di Amazon ECS, consultare [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 la creazione di un servizio, sostituisci *user input placeholders* con le tue informazioni.

```
{
  "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:11112223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
]
}

```

Important

È necessario specificare un tipo di risorsa volume 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 si richiede 0,25, 0,5 o 1 unità CPU virtuale (vCPU) per l'attività Fargate, consigliamo di configurare un volume SSD per uso generico (gp2 o gp3) o un volume Unità disco rigido (HDD) (st1 o sc1). Se si richiede più di 1 vCPU per l'attività Fargate, i seguenti limiti di prestazioni di base si applicano a un volume Amazon EBS collegato all'attività. È possibile ottenere temporaneamente prestazioni EBS superiori ai seguenti limiti. Consigliamo, tuttavia, di pianificare il carico di lavoro in base a questi limiti.

Unità CPU richieste (in v) CPUs	IOPS di base di Amazon EBS (16 KiB I/O)	Throughput di base di Amazon EBS (in, MiBps 128 KB di I/O)	Larghezza di banda della rete 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 si configura 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 l'archivio temporaneo dell'attività e il volume allegato.

Prestazioni dei volumi Amazon EBS per le attività EC2

Amazon EBS fornisce i tipi di volume, che presentano caratteristiche di prestazioni e prezzi diversi, consentendo di definire le prestazioni e i costi di archiviazione in base alle esigenze imposte dalle proprie applicazioni. Per informazioni sulle prestazioni, inclusi IOPS per volume e throughput per volume, consultare [Amazon EBS volume types](#) nella Guida per l'utente di Amazon Elastic Block Store.

Prestazioni dei volumi Amazon EBS per le attività di istanze gestite da Amazon ECS

Amazon EBS fornisce i tipi di volume, che presentano caratteristiche di prestazioni e prezzi diversi, consentendo di definire le prestazioni e i costi di archiviazione in base alle esigenze imposte dalle proprie applicazioni. Per informazioni sulle prestazioni, inclusi IOPS per volume e throughput per volume, consultare [Amazon EBS volume types](#) nella Guida per l'utente di Amazon Elastic Block Store.

Risoluzione dei problemi relativi ai volumi di Amazon EBS collegati alle attività di Amazon ECS

Potrebbe essere necessario risolvere i problemi o verificare il collegamento dei volumi Amazon EBS alle attività di Amazon ECS.

Verificare lo stato di collegamento dei volumi

Puoi utilizzare il Console di gestione AWS per visualizzare lo stato dell'allegato di un volume Amazon EBS a un'attività Amazon ECS. Se l'attività inizia e l'allegato fallisce, verrà visualizzato anche un motivo dello stato che si può utilizzare per risolvere il problema. Il volume creato verrà eliminato e l'operazione verrà interrotta. Per ulteriori informazioni sui motivi di stato, consultare [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 versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Cluster, scegliere il cluster in cui è in esecuzione l'attività. Viene visualizzata la pagina dei dettagli del cluster.
3. Nella pagina dettagli del cluster, scegliere la scheda Attività.
4. Scegliere l'attività per la quale si desidera visualizzare lo stato degli allegati del volume. Potrebbe essere necessario utilizzare Filtra lo stato desiderato e scegliere Arrestato se l'attività che si desidera esaminare è stata interrotta.
5. Nella pagina dettagli dell'attività, scegliere la scheda Volumi. Sarà possibile visualizzare lo stato degli allegati del volume Amazon EBS in Stato degli allegati. Se il volume non riesce a collegarsi all'attività, è possibile scegliere lo stato in Stato degli allegati 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'[DescribeTasksAPI](#).

Errori di servizio e attività

È possibile che si verifichino errori di servizio o attività non specifici dei volumi Amazon EBS che possono influire sull'allegato del volume. Per ulteriori informazioni, consultare la pagina

- [Messaggi di evento relativi ai servizi](#)
- [Codici di errore dell'attività arrestata](#)
- [Motivi di errore dell'API](#)

Il contenitore non può scrivere sul volume Amazon EBS

Utente non root senza autorizzazioni adeguate

Quando specifichi un utente non root nella definizione del contenitore, Amazon ECS configura automaticamente il volume con autorizzazioni basate sul gruppo per consentire l'accesso in scrittura. Tuttavia, se continui a riscontrare problemi di autorizzazione:

- Verifica che il `user` parametro sia specificato correttamente nella definizione del contenitore utilizzando il formato `uid:gid` (ad esempio, `1001:1001`).
- Assicurati che l'immagine del contenitore non prevalga sulle autorizzazioni dell'utente dopo il montaggio del volume.
- Verifica che l'applicazione sia in esecuzione con l'ID utente previsto esaminando i log del contenitore o utilizzando Amazon ECS Exec per ispezionare il contenitore in esecuzione.

Utente root con problemi di autorizzazione

Se non è specificato alcun utente nella definizione del contenitore, il contenitore funziona come root e dovrebbe avere pieno accesso al volume. Se riscontri problemi:

- Verifica che il volume sia montato correttamente controllando i punti di montaggio all'interno del contenitore.
- Assicurati che il volume non sia configurato come di sola lettura nella configurazione del punto di montaggio.

Attività multi-contenitore con utenti diversi

Nelle attività con più contenitori eseguiti come utenti diversi, Amazon ECS gestisce automaticamente le autorizzazioni di gruppo per consentire a tutti gli utenti specificati di scrivere sul volume. Se i contenitori non sono in grado di scrivere:

- Verifica che tutti i contenitori che richiedono l'accesso in scrittura abbiano il `user` parametro configurato correttamente.
- Verificate che il volume sia montato in tutti i contenitori che devono accedervi.

Per ulteriori informazioni sulla configurazione degli utenti nelle definizioni dei container, consulta [Parametri di definizione delle attività di Amazon ECS per Fargate](#).

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 Console di gestione AWS durante la configurazione dei volumi Amazon EBS da allegare alle

attività di Amazon ECS. Per ulteriori informazioni sulla localizzazione di questi motivi di stato nella console, consultare [Verificare lo stato di collegamento dei volumi](#).

ECS non è stato in grado di assumere il ruolo dell'infrastruttura ECS configurato 'arn:aws:iam: ::role/ '. **111122223333 ecsInfrastructureRole** Verificare che il ruolo passato abbia un rapporto di fiducia adeguato con Amazon ECS

Questo motivo dello stato viene visualizzato nei seguenti scenari.

- Fornire un ruolo IAM senza la necessaria policy di attendibilità allegata. Amazon ECS non può accedere al ruolo IAM dell'infrastruttura Amazon ECS che è stato fornito se il ruolo non dispone della policy di attendibilità necessaria. L'attività può rimanere bloccata nello stato DEPROVISIONING. Per ulteriori informazioni sulla policy di attendibilità necessaria, consultare [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- L'utente IAM non è autorizzato a trasferire il ruolo dell'infrastruttura Amazon ECS ad Amazon ECS. L'attività può rimanere bloccata nello stato DEPROVISIONING. Per evitare questo problema, è possibile allegare l'autorizzazione PassRole al proprio utente. Per ulteriori informazioni, consultare [Ruolo IAM dell'infrastruttura Amazon ECS](#).
- Il ruolo IAM non dispone delle autorizzazioni necessarie per gli allegati del volume Amazon EBS. L'attività può rimanere bloccata nello stato DEPROVISIONING. Per ulteriori informazioni sulle autorizzazioni specifiche necessarie per collegare i volumi Amazon EBS alle attività, consultare [Ruolo IAM dell'infrastruttura Amazon ECS](#).

 Note

È possibile anche visualizzare questo messaggio di errore a causa di un ritardo nella propagazione dei ruoli. Se il problema non si risolve riprovando a utilizzare il ruolo dopo aver atteso qualche minuto, è possibile che tu abbia configurato erroneamente la policy di attendibilità per il ruolo.

ECS non è riuscito a configurare il volume EBS. Incontro IdempotentParameterMismatch «; «Il token client che hai fornito è associato a una risorsa che è già stata eliminata. Utilizzare un token client diverso.»

I seguenti scenari AWS KMS chiave possono portare alla IdempotentParameterMismatch visualizzazione di un messaggio:

- Specificare 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 non riesce perché AWS autentica la chiave KMS in modo asincrono. Per ulteriori informazioni, consulta la [crittografia Amazon EBS](#) nella Amazon EC2 User Guide.
- Fornire 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. È quindi possibile utilizzare questi eventi per identificare lo specifico problema relativo alla chiave gestita dal cliente che ha interessato l'allegato del volume. Per ulteriori informazioni, consultare la pagina

- [Come posso creare un gruppo di CloudWatch log da utilizzare come destinazione per una EventBridge regola?](#) su AWS Re:post.
- [Task state change events](#).
- [EventBridge Eventi Amazon per Amazon EBS nella Guida](#) per l'utente di Amazon EBS.

ECS è scaduto durante la configurazione dell'allegato del volume EBS all'attività.

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](#).
- Configurare 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'attività. Per i volumi creati da una snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione della snapshot.

Puoi utilizzare i log degli agenti container di Amazon ECS per risolvere i problemi relativi a questo messaggio relativo alle attività. EC2 Per ulteriori informazioni, consultare [Amazon ECS log file locations](#) e [Amazon ECS log collector](#).

Usare 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 utilizzando la console](#).

Considerazioni

Quando usi i volumi Amazon EFS, tieni presente quanto segue:

- Per le attività eseguite su EC2, 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 versione della piattaforma 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 container supervisor utilizza una piccola quantità di memoria e di 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 del contenitore. `aws-fargate-supervisor` Per ulteriori informazioni sull'utilizzo di EC2, consulta [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#). Per ulteriori informazioni durante l'utilizzo di Fargate, consultare [Endpoint metadati delle attività Amazon ECS versione 4 per le attività su Fargate](#).
- L'utilizzo di volumi Amazon EFS o la specifica di una `EFSVolumeConfiguration` non sono supportati su istanze esterne.
- L'utilizzo dei volumi Amazon EFS è supportato per le attività che utilizzano istanze gestite da Amazon ECS.
- 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](#).

Utilizzo dei 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 di volumi Amazon EFS con Amazon ECS

Prendere nota dei seguenti consigli di best practice quando si utilizza Amazon EFS con Amazon ECS.

Sicurezza e controlli di accesso per i volumi Amazon EFS

Amazon EFS offre funzionalità di controllo degli accessi che è possibile utilizzare per garantire che i dati archiviati in un file system Amazon EFS siano sicuri e accessibili solo dalle applicazioni che ne hanno bisogno. È possibile proteggere i dati abilitando la crittografia dei dati memorizzati su disco e in transito. Per ulteriori informazioni, consulta [Crittografia dati in Amazon EFS](#) nella Amazon Elastic File System User Guide.

Oltre alla crittografia dei dati, è possibile 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, è possibile configurare un gruppo di sicurezza utilizzato per consentire e negare il traffico di rete. È possibile configurare il gruppo di sicurezza collegato ad Amazon EFS per consentire il traffico NFS (porta 2049) dal gruppo di sicurezza collegato alle istanze Amazon ECS o, quando si utilizza la modalità di rete `awsvpc`, l'attività di Amazon ECS.
- **IAM:** è possibile 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, consultare [Using IAM to control file system data access](#) nella Guida per l'utente di Amazon Elastic File System.

Le policy IAM possono inoltre applicare condizioni predefinite, ad esempio richiedere a un client di utilizzare TLS durante la connessione a un file system Amazon EFS. Per ulteriori informazioni, consultare [Amazon EFS condition keys for clients](#), nella Guida per l'utente di Amazon Elastic File System.

- **Punti di accesso Amazon EFS:** i punti di accesso di Amazon EFS sono punti di accesso specifici dell'applicazione in un file system Amazon EFS. È possibile utilizzare i punti di accesso per 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.

Policy IAM

È possibile 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	Description
<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.
<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 policy possono essere definite nei seguenti modi:

- **Basata su client:** collega la policy al ruolo dell'attività

Impostare l'opzione Autorizzazione IAM quando si crea la definizione dell'attività.

- Basata sulle risorse: collega la policy al file system Amazon EFS

Se la policy basata sulle risorse non esiste, per impostazione predefinita al momento della creazione del file system l'accesso è concesso a tutti i principali (*).

Quando si imposta l'opzione Autorizzazione IAM, uniamo la policy associata al ruolo dell'attività e quella basata sulle risorse Amazon EFS. L'opzione 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 di container o all'attività Amazon ECS. Inoltre, è possibile 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, è possibile utilizzare i punti di accesso di 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 di volume Amazon EFS

Amazon EFS offre due modalità di prestazioni: i I/O. General Purpose is suitable for latency-sensitive applications such as content management systems and CI/CD tools. In contrast, Max I/O file system General Purpose e Max 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, consultare [Amazon EFS performance modes](#) nella Guida per l'utente di Amazon Elastic File System.

Alcuni carichi di lavoro sensibili alla latenza richiedono sia i I/O livelli più alti forniti dalla modalità Max I/O Performance sia la latenza più bassa fornita dalla modalità a prestazioni 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 dell'applicazione su tutti questi file system, purché il carico di lavoro e le applicazioni possano supportarlo.

Throughput 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 Throughput assegnato o dalla quantità di dati archiviati nella classe di archiviazione EFS Standard o a zona unica per i file system che utilizzano Throughput di bursting. Per ulteriori informazioni, consultare [Understanding metered throughput](#) nella Guida per l'utente di Amazon Elastic File System.

La modalità di throughput predefinita per i file system Amazon EFS è la modalità bursting. Con la modalità bursting, il throughput disponibile per un file system aumenta o diminuisce man mano che quest'ultimo cresce. Poiché i carichi di lavoro basati su file in genere registrano picchi, richiedendo livelli di throughput elevati per certi periodi di tempo e livelli di throughput inferiori per il resto, Amazon EFS è progettato per consentire livelli di throughput elevati per certi periodi di tempo. Inoltre, poiché molti carichi di lavoro richiedono un elevato livello di lettura, le operazioni di lettura vengono misurate con un rapporto 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 per ogni TB di storage Amazon EFS Standard o Amazon EFS One Zone. Tutti i file system (indipendentemente dalle dimensioni) possono arrivare a 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 velocità di lettura. Man mano che aggiungi dati al file system, il throughput massimo disponibile per il file system scala in modo lineare e automatico con l'archivio nella classe di archiviazione Amazon EFS Standard. Se è necessaria una quantità

maggiore di throughput rispetto a quella che si può ottenere con la quantità di dati archiviati, è possibile configurare il Throughput assegnato sulla quantità specifica richiesta dal 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 in grado di raggiungere il 100% MB/s del throughput può generare 100 MB/s da una singola EC2 istanza Amazon, ciascuna con una velocità effettiva di 10 MB/s. Per ulteriori informazioni, consulta [Amazon EFS performance](#) nella Guida per l'utente di Amazon Elastic File System.

Ottimizzazione dei costi per i volumi di Amazon EFS

Amazon EFS semplifica la scalabilità dell'archiviazione. I file system di Amazon EFS crescono automaticamente man mano che aggiungi più dati. Specialmente con la modalità Throughput di bursting di Amazon EFS, il throughput su Amazon EFS scala mentre le dimensioni del file system nella classe di archiviazione Standard aumentano. Per migliorare il throughput senza pagare costi aggiuntivi per il throughput assegnato su un file system EFS, è possibile condividere un file system Amazon EFS con più applicazioni. Utilizzando i punti di accesso Amazon EFS, puoi implementare l'isolamento dell'archiviazione nei file system Amazon EFS condivisi. In questo modo, anche se le applicazioni condividono ancora lo stesso file system, possono accedere ai dati solo con l'autorizzazione.

Man mano che i dati crescono, Amazon EFS ti aiuta a spostare automaticamente i file a cui si accede raramente in una classe di archiviazione inferiore. La classe di archiviazione Accesso Infrequente Amazon EFS Standard (IA) riduce i costi di archiviazione per i file ai quali non viene effettuato l'accesso ogni giorno. Ciò avviene senza sacrificare disponibilità elevata, durabilità elevata, elasticità e accesso al file system POSIX disponibili tramite Amazon EFS. Per ulteriori informazioni, consultare [EFS storage classes](#) nella Guida per l'utente di Amazon Elastic File System.

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 nell'archivio di Amazon EFS IA. Per ulteriori informazioni, consulta [Amazon EFS lifecycle management](#) nella Guida per l'utente di Amazon Elastic File System.

Quando si crea un file system Amazon EFS, è possibile scegliere se Amazon EFS replica i dati su più zone di disponibilità (Standard) o se li archivia in modo ridondante all'interno di un'unica zona di disponibilità. La classe di archiviazione Amazon EFS a zona unica può ridurre i costi di archiviazione di un margine significativo rispetto alle classi di archiviazione Amazon EFS Standard. Prendere in considerazione l'utilizzo della classe di archiviazione Amazon EFS a zona unica per carichi di lavoro che non richiedono resilienza Multi-AZ. È possibile ridurre ulteriormente il costo dell'archiviazione

Amazon EFS a zona unica spostando i file a cui si accede raramente su Accesso infrequente a zona unica Amazon EFS. 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 utilizzando classi di archiviazione standard. Se si selezionano le classi di archiviazione Amazon EFS a zona unica, i dati vengono archiviati in modo ridondante all'interno di una singola zona di disponibilità. Inoltre, Amazon EFS è progettato per fornire il 99,999999999% (11 volte 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 archiviazione Amazon EFS a zona unica 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, consultare [Backing up EFS file systems](#) della Guida per l'utente di Amazon Elastic File System.

Specificare di un file system Amazon EFS nella definizione di un'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

Obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

fileSystemId

Tipo: stringa

Obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

rootDirectory

─Tipo: stringa

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: String

Valori validi: ENABLED | DISABLED

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 [Encrypting Data in Transit](#) nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: Integer

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

Obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

■Tipo: stringa

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: String

Valori validi: ENABLED | DISABLED

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 utilizzando la console

Informazioni su come utilizzare 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 <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 `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 EC2istanze 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 dà per scontato che il file system Amazon EFS, il cluster Amazon ECS, le istanze di container e i processi si trovano nello stesso VPC. Per ulteriori informazioni sul montaggio di un file system da un VPC diverso, consultare [Walkthrough: Mount a file system from a different VPC](#) nella Guida per l'utente di Amazon EFS.

4. Nella pagina Impostazioni del file system, configura le impostazioni facoltative, quindi in Impostazioni delle prestazioni, scegli Ottimizzazione come modalità di throughput 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

Il ruolo IAM di esecuzione delle attività di Amazon ECS non richiede autorizzazioni specifiche relative ad Amazon EFS per montare un file system di Amazon EFS. Per impostazione predefinita, se non esiste una policy basata sulle risorse di Amazon EFS, l'accesso è concesso a tutti i principali (*) al momento della creazione del file system. Il ruolo dell'attività di Amazon ECS è richiesto solo se "Autorizzazione EFS IAM" è abilitata nella definizione dell'attività di Amazon ECS. Se abilitata, l'identità del ruolo dell'attività deve avere accesso al file system di Amazon EFS nella policy basata sulle risorse di Amazon EFS e l'accesso anonimo deve essere disabilitato.

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 <https://console.aws.amazon.com/ecs/v2>.
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	- Nella pagina Cluster, seleziona il cluster in cui creare il servizio. - Dalla scheda Processi, scegli Esegui nuovo processo.	
Tipo di lancio	- Nella pagina Task (Attività), scegli la definizione di attività. - Se è presente più di una revisione, selezionane una. - 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 lancio	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. Invece, è possibile [modificare i container 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 Gestione dei segreti AWS 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 non è supportato su Amazon ECS Managed Instances.
- 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 riuscirà a eseguire l'avvio dell'attività e sarà restituito il seguente messaggio di errore: `unable to mount file system fs-id`.
- FSx for Windows File Server è integrato con AWS Identity and Access Management (IAM) per controllare le azioni che gli utenti e i gruppi IAM possono eseguire 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. Si tratta della crittografia dei dati in transito e di quella dei dati a riposo. La crittografia dei dati in transito è supportata sulle condivisioni di file mappate su un'istanza di container che supporta il protocollo SMB 3.0 o successivo. 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 le credenziali salvate in Secrets Manager. Il nome della risorsa Amazon (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 le credenziali salvate 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

Per utilizzarli FSx per i volumi del file system Windows File Server per i contenitori, specifica 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

Obbligatorio: no

Questo parametro viene specificato quando si utilizza FSx il file system [Windows File Server](#) per l'archiviazione delle attività.

`fileSystemId`

Tipo: stringa

Obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

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

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

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.

- Gestione dei segreti AWS segreto

Questa credenziale può essere creata nella Gestione dei segreti AWS console utilizzando la categoria segreta Altro tipo. Si aggiunge una riga per ogni key/value coppia username/admin e una *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 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. Per informazioni, consulta [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. 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 Console di gestione AWS.

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 SSMManaged InstanceCore`
 - `Amazon SSMDirectory ServiceAccess`

Per allegare politiche AWS gestite.

- a. Aprire la [console IAM](#).

- b. Nel pannello di navigazione, selezionare Ruoli.
 - c. Scegli un ruolo gestito da AWS .
 - d. Scegli Autorizzazioni, Collega policy.
 - e. Per limitare le policy disponibili da collegare, utilizza Filtro.
 - f. Seleziona la policy appropriata, quindi scegli Collega policy.
3. 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)

1. 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. Annotare inoltre il nome completo della directory DNS per le operazioni successive. È possibile completare la fase 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. Fare clic su Avanti.
- h. Fare clic su Avanti. 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-securitygroup</i>

Tipo	Protocollo	Intervallo porte	Origine
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 del 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 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 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/versions/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 [Retrieving Amazon ECS-Optimized AMI metadata](#).
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/versions/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 Launch Type, 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 [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 windows-fsx-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.

Utilizzo dei 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, consultare [Volumes](#) nella documentazione di Docker.

Alcuni casi d'uso 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 dell'attività, consultare [Specifica un volume Docker in una definizione dell'attività Amazon ECS](#).

Specifica un volume Docker in una definizione dell'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: stringa

Obbligatorio: no

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (-) e trattini bassi (_). Nel parametro `sourceVolume` dell'oggetto `mountPoints` della definizione del container viene fatto riferimento a questo nome.

dockerVolumeConfiguration

Tipo: oggetto [DockerVolumeConfiguration](#)

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 container Windows supportano solo l'uso del driver `local`. Per utilizzare i montaggi vincolati, specifica invece un `host`.

scope

Tipo: String

Valori validi: `task` | `shared`

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`

Obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo è utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omesso.

`driver`

─Tipo: stringa

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 è utilizzato per il posizionamento dell'attività. Se il driver è stato installato utilizzando la CLI del plugin Docker, utilizzare `docker plugin ls` per richiamare il nome del driver dall'istanza di container. Se il driver è stato installato utilizzando un altro metodo, utilizzare il rilevamento del plugin Docker per richiamare il nome del driver.

`driverOpts`

─Tipo: stringa

Obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da inviare. Questo parametro fa riferimento a `DriverOpts` nella sezione [Crea un volume di Docker](#).

`labels`

─Tipo: stringa

Obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`mountPoints`

Tipo: array di oggetti

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai Volumes nella creazione container dell'API Docker e l'opzione `--volume` per `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

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, lascia 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 contenitori e come fornire storage persistente NFS per un contenitore.

Fornire un archivio temporaneo per un container 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 un archivio persistente per più container 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"  
      }  
    ]  
  }  
]
```

Utilizzo di montaggio vincolato 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 montaggi vincolati sono legati al ciclo di vita del container 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, consultare [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. Puoi aumentare la quantità totale di archiviazione temporanea fino a un massimo di 200 GiB, specificando il parametro `ephemeralStorage` nella definizione di attività.
- 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 montaggio vincolato in una definizione dell'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 `permountPoints`, 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 a `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: stringa

Obbligatorio: no

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (-) e trattini bassi (_). Nel parametro `sourceVolume` dell'oggetto `mountPoints` della definizione del container viene fatto riferimento a questo nome.

host

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: stringa

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

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai `Volumes` nella creazione container dell'API Docker e l'opzione `--volume` per `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

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, lascia il valore predefinito di `false`

`ephemeralStorage`

Tipo: oggetto

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 vincolato per Amazon ECS

Gli esempi seguenti riguardano i casi d'uso comuni per l'utilizzo di un montaggio vincolato per i container.

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 [Usare 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",  
    "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'immagine `my-repo/httpd_dockerfile_volume` di esempio utilizzata nelle seguenti fasi 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 convertito nell'opzione `--memory-swap` in `docker run` dove il valore sarebbe la somma della memoria del container più il valore maxSwap.

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 omissso, 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'opzione `--memory-swappiness` su `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 EC2 specificano. 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 container.
- Se utilizzi le attività su Amazon Linux 2023, il parametro `swappiness` non è supportato.

Differenze nella definizione dell'attività di Amazon ECS per istanze gestite da Amazon ECS

Per utilizzare Istanze gestite da Amazon ECS, bisogna configurare la definizione dell'attività per utilizzare il tipo di avvio di istanze gestite da Amazon ECS. Sono possibili ulteriori considerazioni quando si utilizza Istanze gestite da Amazon ECS.

Parametri di definizione di attività

I processi che utilizzano Istanze gestite da Amazon ECS supportano la maggior parte dei parametri di definizione delle attività Amazon ECS disponibili. Tuttavia, alcuni parametri presentano comportamenti o limitazioni specifici se utilizzati con attività di Istanze gestite da Amazon ECS.

I parametri di definizione delle attività seguenti non sono validi per le attività di Istanze gestite da Amazon ECS:

- `disableNetworking`
- `dnsSearchDomains`
- `dnsServers`
- `dockerLabels`
- `dockerSecurityOptions`
- `dockerVolumeConfiguration`
- `ephemeralStorage`
- `extraHosts`
- `fsxWindowsFileServerVolumeConfiguration`
- `hostname`
- `inferenceAccelerator`

- `ipcMode`
- `links`
- `maxSwap`
- `proxyConfiguration`
- `sharedMemorySize`
- `Volumi sourcepath`
- `swappiness`
- `tmpfs`

I seguenti parametri di definizione delle attività sono validi in Istanze gestite da Amazon ECS, ma hanno delle limitazioni da considerare:

- `networkConfiguration`: le attività di Istanze gestite da Amazon ECS utilizzano la modalità di rete `awsvpc` o `host`.
- `placementConstraints`: sono supportati gli attributi di vincolo seguenti.
 - `ecs.subnet-id`
 - `ecs.availability-zone`
 - `ecs.instance-type`
 - `ecs.cpu-architecture`
- `requiresCompatibilities`: è necessario includere `MANAGED_INSTANCES` per garantire che la definizione dell'attività sia compatibile con le istanze gestite da Amazon ECS.
- `resourceRequirement`: `InferenceAccelerator` non è supportato.
- `operatingSystemFamily`: Istanze gestite da Amazon ECS utilizza `LINUX`.

Per assicurarsi che la definizione di attività sia valida per l'utilizzo con Istanze gestite da Amazon ECS, è possibile specificare quanto riportato di seguito quando si registra la definizione dell'attività:

- Nel campo `Richiede compatibilità Console di gestione AWS`, specificare. `MANAGED_INSTANCES`
- Nel `AWS CLI`, specificare l'opzione. `--requires-compatibilities`
- Nell'API Amazon ECS, specifica il flag `requiresCompatibilities`.

Differenze nelle definizioni di attività Amazon ECS per Fargate

Per utilizzare Fargate, è necessario configurare la definizione dell'attività in modo da utilizzare il tipo di avvio Fargate. Ci sono ulteriori considerazioni quando si utilizza Fargate.

Parametri di definizione di attività

I processi che utilizzano 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 container Windows su AWS Fargate, il valore non può essere inferiore a 1 vCPU.
- `networkConfiguration`: le attività Fargate utilizzano sempre la modalità di rete `awsvpc`.

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 Console di gestione AWS, per il campo Richiede compatibilità, 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 attività e container per AWS Fargate, è necessario specificare il sistema operativo eseguito dal container. Sono supportati i seguenti sistemi operativi per AWS Fargate:

- Amazon Linux 2

Note

I container Linux utilizzano solo il kernel e la relativa configurazione 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 container Windows su AWS Fargate, è necessario disporre di un'architettura 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 di attività per carichi di lavoro ARM a 64 bit”](#).

CPU e memoria del processo

Le definizioni di attività di Amazon ECS per AWS Fargate richiedono che la CPU e la memoria vengano specificate a livello di processo. 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 pari a 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 (v). CPUs CPUs 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
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

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
ma 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 ma Linux 1.4.0 o successiva.		

Reti di attività

I processi di Amazon ECS per AWS Fargate richiedono la modalità di rete `awsvpc`, la quale fornisce un'interfaccia di rete elastica a ciascun processo. 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 sezione `networkConfiguration` per un servizio Fargate:

```
"networkConfiguration": {
  "awsvpcConfiguration": {
```

```
"assignPublicIp": "ENABLED",  
"securityGroups": [ "sg-12345678" ],  
"subnets": [ "subnet-12345678" ]  
}  
}
```

Limiti delle risorse dei processi

Le definizioni di attività di Amazon ECS per container Linux su AWS Fargate supportano il parametro `ulimits` per definire i limiti di risorse da impostare per un container.

Le definizioni di attività di Amazon ECS per Windows su AWS Fargate non supportano il parametro `ulimits` per definire i limiti di risorse da impostare per un container.

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 dei log

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 Console di gestione AWS 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

Le definizioni di attività di Amazon ECS per AWS Fargate supportano i driver di log `awslogs`, `splunk` e `awsfirelens` per la configurazione dei log.

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 [Driver di log splunk](#).

Archiviazione dei processi

Per i processi Amazon ECS ospitati su Fargate, sono supportati i seguenti tipi di archiviazione:

- I volumi Amazon EBS forniscono un'archiviazione a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Per ulteriori informazioni, consulta [Usare i volumi Amazon EBS con Amazon ECS](#).
- Volumi Amazon EFS per l'archiviazione persistente. Per ulteriori informazioni, consulta [Usare i volumi Amazon EFS con Amazon ECS](#).
- Montaggi vincolati per l'archiviazione temporanea. Per ulteriori informazioni, consulta [Utilizzo di montaggio vincolato con Amazon ECS](#).

Caricamento lento delle immagini di container utilizzando Seekable OCI (SOCl)

Le attività di Amazon ECS su Fargate che utilizzano la versione della piattaforma Linux 1.4.0 possono utilizzare Seekable OCI (SOCl) per avviare le attività più velocemente. Con SOCl, 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 SOCl per un'immagine nell'operazione e avvia il container senza attendere che l'intera immagine venga scaricata.

Per i container che funzionano senza indici SOCl, 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.

Seekable OCI (SOCl) è una tecnologia open source sviluppata da Seekable OCI AWS che può avviare i container più velocemente caricando pigramente l'immagine del contenitore. SOCl funziona creando un indice (indice SOCl) 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 SOCl deve essere archiviato come artefatto nello stesso repository dell'immagine all'interno del registro dei container. Devi utilizzare solo indici SOCl 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](#).

I clienti che desiderano utilizzare SOCl potranno usare solo il manifesto dell'indice SOCl v2. I clienti esistenti che hanno già utilizzato SOCl su Fargate possono continuare a usare il manifesto dell'indice SOCl v1, tuttavia consigliamo vivamente di migrare al manifesto dell'indice SOCl v2. Il manifesto

dell'indice SOCI v2 crea una relazione esplicita tra le immagini dei container e i relativi indici SOCI per garantire implementazioni coerenti.

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.
- Sono supportate le attività eseguite sull'architettura della CPU X86_64 o ARM64.
- 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.
- Sono supportate solo le immagini di container che utilizzano la compressione gzip o non sono compresse. Le immagini di container che utilizzano la compressione zstd non sono supportate.
- Per il manifesto dell'indice SOCI v2, la generazione di un manifesto dell'indice SOCI modifica il manifesto dell'immagine del container in quanto viene aggiunta un'annotazione per tale indice. Ciò si traduce in un nuovo digest di immagini del container. Il contenuto dei livelli del filesystem delle immagini del container non cambia.
- Per il manifesto dell'indice SOCI v2, quando l'immagine del container è già stata archiviata nel repository di immagini del container, dopo aver generato un indice SOCI, è necessario ripristinare l'immagine del container. Il reinserimento dell'immagine del container non aumenterà i costi di archiviazione duplicando i livelli del filesystem, ma caricherà solo un nuovo file manifesto.
- Consigliamo di provare il caricamento lento con immagini di container con dimensioni della compressione superiori a 250 MiB. 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 si desidera evitare che un'immagine di container venga caricata lentamente, sarà necessario eseguire di nuovo il push dell'immagine di container senza un indice SOCI allegato.

Creazione di un indice Seekable OCI

Affinché un'immagine del container venga caricata lentamente, è necessario un indice SOCI (un file di metadati) creato e archiviato nel repository di immagini del container insieme all'immagine del container. 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 container 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 nell'archivio di immagini containerd sul computer che esegue `soci-snapshotter`. Se l'immagine è nell'archivio di immagini Docker, 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. Per verificare che un manifesto dell'indice SOCI v2 sia allegato a un'immagine del container, è possibile recuperare l'indice delle immagini da Amazon ECR utilizzando AWS CLI.

```
IMAGE_REPOSITORY=r
IMAGE_TAG=latest

aws ecr batch-get-image \
  --repository-name=$IMAGE_REPOSITORY \
  --image-ids imageTag=$IMAGE_TAG \
  --query 'images[0].imageManifest' --output text | jq -r '.manifests[] |
select(.artifactType=="application/vnd.amazon.soci.index.v2+json")'
```

Per verificare che un manifesto dell'indice SOCI v1 sia allegato a un'immagine del container, puoi utilizzare l'API OCI Referrers.

```
ACCOUNT_ID=111222333444
AWS_REGION=us-east-1
IMAGE_REPOSITORY=nginx-demo
IMAGE_TAG=latest
IMAGE_DIGEST=$(aws ecr describe-images --repository-name $IMAGE_REPOSITORY --image-ids
  imageTag=$IMAGE_TAG --query 'imageDetails[0].imageDigest' --output text)
ECR_PASSWORD=$(aws ecr get-login-password)

curl \
  --silent \
  --user AWS:$ECR_PASSWORD \
  https://$ACCOUNT_ID.dkr.ecr.$AWS_REGION.amazonaws.com/v2/$IMAGE_REPOSITORY/
  referrers/$IMAGE_DIGEST?artifactType=application%2Fvnd.amazon.soci.index.v1%2Bjson | jq
  -r '.'
```

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 supportati, mentre altri si comportano diversamente.

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 di Amazon ECS attraverso la nuova console

Creare una definizione di attività in modo da poter definire l'applicazione che si esegue 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 parametro `requireCapabilities` su `EXTERNAL`.

È possibile creare una definizione di attività utilizzando l'esperienza della console o specificando un file JSON. È possibile fare in modo che Amazon Q fornisca consigli quando si usa l'editor JSON. Per ulteriori informazioni, consulta [Utilizzo di Amazon Q Developer per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS](#)

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`.
- C'è almeno una voce sotto `containerDefinitions`

CloudFormation pile

Il comportamento seguente si applica alle definizioni di attività che sono state 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, consultare [Deleting a stack](#) nella Guida per l'utente di 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 Creare nuova definizione di attività, scegliere Crea nuova definizione 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 (che è serverless). Amazon ECS utilizza questo valore per eseguire la convalida così da garantire che i parametri di definizione di 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 ARM64Linux/. 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
8192 (8 vCPU)	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux

 **Note**
 Questa opzione richiede la piattafor

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
ma 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.		

Per le attività che utilizzano EC2 istanze o istanze esterne, i valori CPU delle attività supportate sono compresi tra 128 unità CPU (0,125 vCPUs) e 196608 unità CPU (192 vCPUs).

Per specificare il valore di memoria in GB, immetti GB dopo il valore. Ad esempio, per impostare il valore di memoria su 3 GB, immetti 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 si sceglie bridge, in Mappature delle porte, per Porta host, specificare il numero di porta nell'istanza di container per prenotare per il container.

- (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 serve il ruolo, scegliere Nessuno.

10. (Facoltativo) Espandere la sezione Posizionamento delle attività per aggiungere vincoli di posizionamento. I vincoli di posizionamento dell'attività consentono di filtrare le istanze di container utilizzate per il posizionamento delle attività tramite attributi integrati o personalizzati.
11. (Facoltativo) Espandere la sezione iniezione di guasti per abilitarla. L'iniezione di guasti consente di testare la risposta dell'applicazione 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 Galleria pubblica di Amazon ECR possono essere specificate utilizzando solo il nome del registro pubblico di Amazon ECR. Ad esempio, se è specificato `public.ecr.aws/ecs/amazon-ecs-agent:latest`, viene utilizzato il container Amazon Linux ospitato su Galleria pubblica di Amazon ECR. Per tutti gli altri repository, specificare il repository utilizzando i formati `repository-url/image:tag` 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. In Container essenziale, se la definizione del processo ha due o più container definiti, è possibile specificare se il container deve essere considerato essenziale. Quando un container è contrassegnato come Essenziale, in caso di arresto, il processo viene interrotto. 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 dei limiti di CPU, GPU e memoria a livello di container diversi dai valori a livello di attività, in Limiti di allocazione delle risorse, procedere come segue:
 - Per CPU, inserire il numero di unità di CPU che l'agente di container di Amazon ECS deve riservare per il container.
 - 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 di attività per carichi di lavoro GPU”](#).

- Per Limite di memoria rigido, inserire la quantità di memoria, in GB, da prenotare per il container. Se il container tenta di superare il limite rigido, si arresta.
- Il Docker 20.10.0 o daemon successivo prenota un minimo di 6 mebibytes (MiB) di memoria per un container, perciò non specificarne di meno per i container.

Il Docker 19.03.13-ce o daemon precedente prenota un minimo di 4 MiB di memoria per un container, perciò non specificare meno di 4 MiB di memoria per i container.

- 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 conservare la memoria del container 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 per i container Windows.

- h. (Facoltativo) Espandi la sezione Variabili di ambiente per specificare le variabili di ambiente da inserire nel container. È possibile specificare variabili di ambiente singolarmente utilizzando coppie chiave-valore o in blocco specificando un file variabile di ambiente ospitato in un bucket Amazon S3. Per informazioni su come formattare un file di variabile di ambiente, consultare [Passare una singola variabile di ambiente a un container Amazon ECS](#).

Quando si specifica una variabile di ambiente per l'archiviazione segreta, immettere il nome segreto in Chiave. Quindi ValueFrom, inserisci l'ARN completo del segreto di Systems Manager Parameter Store o del segreto di Secrets Manager

- i. (Facoltativo) Seleziona l'opzione 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, 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.
- Esporta i log in Splunk: configura l'attività per inviare i log del container al driver Splunk che a sua volta li invia a un servizio remoto. È necessario immettere l'URL al servizio Web Splunk. Il token Splunk è specificato come opzione segreta perché può essere trattato come dati sensibili.
- Esporta i log in Amazon Data Firehose: configura l'attività per inviare i log del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
- Esporta i log in Flusso di dati Amazon Kinesis: configura l'attività per inviare i log del container a Flusso di dati Kinesis. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna 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.
 - Esporta i log in Amazon S3: configura l'attività per inviare i log del container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del 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 policy di riavvio per riavviare un container all'uscita.	Espandere Policy di riavvio, quindi configurare i seguenti elementi: • Per abilitare una politica di riavvio per il container , attiva la Abilita politica di riavvio. • In Codici di uscita ignorati, specificare un elenco per i codici di uscita interi del container separati da virgole. Se il container 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. • In Periodo di tentativo di ripristino, specificare un periodo di tempo intero, in secondi, per il quale il container deve essere eseguito prima di poter tentare il riavvio in caso di uscita. Amazon	

Per configurare questa opzione	Esegui questa operazione	
	ECS può tentare di riavviare un container solo una volta ogni secondo del Periodo di tentativo di ripristino. Se non viene specificato nulla, il container 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 container è integro. Per ulteriori informazioni, consulta Determina lo stato delle attività di Amazon ECS utilizzando i controlli dell'integrità dei container.	HealthCheckEspandi e configura i seguenti elementi: • In Comando, inserire 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 Intervallo, inserire il numero di secondi tra ogni controllo dell'integrità. I valori validi sono compresi tra 5 e 30. • In Timeout, specificare il periodo di tempo in secondi per cui attendere che un controllo dell'integrità venga superato prima	

Per configurare questa opzione	Esegui questa operazione	
	che sia considerato un errore. I valori validi sono compresi tra 2 e 60. • In Periodo di inizio, inserire 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 Tentativi, inserire 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	
Ordine delle dipendenze di avvio Questa opzione definisce le dipendenze per l'avvio e l'arresto del container. Un contenitore 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 a risolvere le dipendenze per un container, in Timeout iniziale, inserire il numero di secondi. • Per configurare il tempo di attesa prima dell'arresto del container se questa interruzione non avviene normalmente, in Timeout di arresto, inserire 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	
Configurazione Docker Questa operazione sovrascrive i valori in Dockerfile.	Espandere Configurazione Docker, quindi configurare gli elementi seguenti: In Comando, inserire un comando eseguibile per un container. Questo parametro è mappato a Cmd nella sezione Create a container di Docker Remote API e l'opzione COMMAND a <code>docker run</code>. Questo parametro sostituisce l'istruzione CMD in Dockerfile. In Punto di ingresso, inserire l'ENTRYPOINT Docker che viene trasmesso al container. Questo parametro è mappato a Entrypoint nella sezione Create a container di Docker Remote API e l'opzione <code>--entrypoint</code> a <code>docker run</code>. Questo parametro sostituisce l'istruz	

Per configurare questa opzione	Esegui questa operazione	
	one ENTRYPOINT in 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 mappato a WorkingDir nella sezione Create a container di Docker Remote API e l'opzione <code>--workdir</code> a <code>docker run</code>. Questo parametro sostituisce l'istruzione WORKDIR in Dockerfile.	

Per configurare questa opzione	Esegui questa operazione	
Limiti di risorse (Ulimits) Questi valori sovrascrivono l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è mappato a <code>Ulimits</code> nella sezione Create a container di Docker Remote API e l'opzione <code>--ulimit</code> a docker run.	Espandere Limiti di risorse (ulimits), quindi scegliere Aggiungi ulimit. In Nome limite, selezionare il limite. Quindi, in Limite flessibile e Limite rigido, inserire i relativi valori. Per aggiungerne altri ulimits, scegliere Aggiungi ulimit.	
Etichette Docker Questa opzione aggiunge metadati al container. Questo parametro è mappato a <code>Labels</code> nella sezione Create a container di Docker Remote API e l'opzione <code>--label</code> a docker run.	Espandere Etichette Docker, scegliere Aggiungi coppia chiave-valore, quindi inserire i valori Chiave e Valore. Per aggiungere altre etichette Docker, scegliere Aggiungi coppia chiave-valore.	

- 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 archiviazione temporanea per i processi ospitati 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 attività Fargate, in Quantità, immetti un valore massimo di 200 GiB.

14. (Facoltativo) Per aggiungere una configurazione del volume di dati per la definizione di attività, scegliere Aggiungi volume, quindi seguire questi passaggi.
 - a. Per 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, selezionare se si desidera configurare il volume durante la creazione della definizione dell'attività o durante l'implementazione.

 Note

I volumi che possono essere configurati durante la creazione di una definizione di attività includono Bind mountDocker, Amazon EFS e Amazon FSx for Windows File Server. Volumi che possono essere configurati al momento dell'implementazione durante l'esecuzione di un'attività o durante la creazione o l'aggiornamento di un servizio includono Amazon EBS.

- c. Per Tipo di volume, selezionare un tipo di volume compatibile con il tipo di configurazione selezionato, quindi configurarlo.

Tipo di volume	Fasi	
Bind mount	a. Scegli Aggiungi punto di montaggio, quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Volume sorgente, scegli il volume di dati da montare sul container. • Per 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 Aggiungi punto di montaggio.	

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 Volume sorgente, scegli il volume di dati da montare sul container. • Per 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. f. Per aggiungere altri punti di montaggio,	

Tipo di volume	Fasi	
	seleziona Aggiungi punto di montaggio.	

Tipo di volume	Fasi	
Docker	a. Per Driver, inserisci la configurazione del volume Docker. I container Windows supportano solo l'uso del driver locale. Per utilizzar e 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 Volume sorgente, scegli il volume di dati da montare sul container. • Per 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 Aggiungi punto di montaggio.	

Tipo di volume	Fasi	
FSx per Windows File Server	a. Per ID del file system, scegli FSx l'ID del file system Windows File Server. b. 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. c. Per il parametro Credenziali, scegli il metodo di archiviazione delle credenziali. • Per utilizzarlo Gestione dei segreti AWS, 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. - d.	

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 Aggiungi punto di montaggio, quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Volume sorgente, scegli il volume di dati da montare sul container. • Per 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. f.	

Tipo di volume	Fasi	
	Per aggiungere altri punti di montaggio, seleziona Aggiungi punto di montaggio.	

Tipo di volume	Fasi	
Amazon EBS	a. Scegli Aggiungi punto di montaggio, quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Volume sorgente, scegli il volume di dati da montare sul container. • Per 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 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 l' AWS Distro for OpenTelemetry integrazione, espandi Monitoraggio, quindi seleziona Usa la raccolta dei parametri 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 Endpoint di scrittura remota dell'area di lavoro, specificare l'URL dell'endpoint di scrittura remota per l'area di lavoro Prometheus. Per Destinazione di scraping, specificare l'host e la porta che il raccoglitore AWS Distro for OpenTelemetry può utilizzare per lo scraping dei 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 Endpoint di scrittura remota dell'area di lavoro, specificare l'URL dell'endpoint di scrittura remota per l'area di lavoro Prometheus. 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. <https://console.aws.amazon.com/ecs/Apri> la console nella versione 2.

2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Nel menu Crea nuova definizione di attività, scegliere Crea nuova definizione 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).

Utilizzo di Amazon Q Developer per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS

Quando utilizzi l'editor JSON nella console Amazon ECS per creare una definizione di attività, puoi utilizzare Amazon Q Developer per fornire suggerimenti di codice generati dall'intelligenza artificiale per le definizioni delle attività.

Puoi utilizzare la funzionalità di chat in linea per chiedere ad Amazon Q Developer di generare, spiegare o rifattorizzare la definizione delle attività in formato JSON con un'interfaccia conversazionale. È possibile inserire in qualsiasi momento suggerimenti generati nella definizione dell'attività e accettare o rifiutare le modifiche proposte. Inoltre, Amazon ECS ha migliorato la funzionalità di suggerimenti in linea esistente per utilizzare Amazon Q Developer.

Quando si crea una definizione di attività utilizzando l'editor JSON, è possibile fare in modo che Amazon Q Developer fornisca consigli per creare più in fretta una definizione di attività. È possibile avere suggerimenti in linea basati sulle proprietà o utilizzare i suggerimenti di Amazon Q Developer per completare automaticamente interi blocchi di codice di esempio.

È possibile utilizzare questa funzionalità nelle regioni in cui è supportato Amazon Q Developer. Per ulteriori informazioni, consultare [Servizi AWS per regione](#).

Prerequisiti

Di seguito sono indicati i prerequisiti:

- Oltre alle autorizzazioni della console, l'utente che crea la definizione dell'attività nella console deve disporre dell'autorizzazione `codewhisperer:GenerateRecommendations` per i consigli e `q:SendMessage` per utilizzare la chat in linea. Per ulteriori informazioni, consulta [Autorizzazioni necessarie per usare Amazon Q Developer per fornire consigli nella console](#).

Procedura

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. Nel menu Crea nuova definizione di attività, scegliere Crea nuova definizione attività con JSON.

Si apre la pagina Crea nuova definizione di attività.

La console offre il seguente modello predefinito.

```
{
  "requiresCompatibilities": [
    "FARGATE"
  ],
  "family": "",
  "containerDefinitions": [
    {
      "name": "",
      "image": "",
      "essential": true
    }
  ],
  "volumes": [],
  "networkMode": "awsvpc",
  "memory": "3 GB",
  "cpu": "1 vCPU",
  "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
}
```

4. Nel pop-up dei suggerimenti in linea di Amazon Q, scegli Consenti.

Se si chiude il pop-up, è possibile abilitare Amazon Q sotto l'icona a forma di ingranaggio.

5. Nella casella dell'editor JSON, modifica il documento JSON.

Per fare in modo che Amazon Q crei e compili i parametri, inserire un commento con ciò che si desidera aggiungere. Nell'esempio seguente, il commento fa sì che Amazon Q generi le righe in grassetto.

```
{
  "requiresCompatibilities": [
    "FARGATE"
```

```

    ],
    "family": "",
    "containerDefinitions": [
      {
        "name": "",
        "image": "",
        "essential": true
      },
      // add an nginx container using an image from Public ECR, with port 80
      // open, and send logs to CloudWatch log group "myproxy"
      {
        "name": "nginx",
        "image": "public.ecr.aws/nginx/nginx:latest",
        "essential": true,
        "portMappings": [
          {
            "containerPort": 80,
            "hostPort": 80,
            "protocol": "tcp"
          }
        ],
        "logConfiguration": {
          "logDriver": "awslogs",
          "options": {
            "awslogs-group": "myproxy",
            "awslogs-region": "us-east-1",
            "awslogs-stream-prefix": "nginx"
          }
        }
      }
    ],
    "volumes": [],
    "networkMode": "awsvpc",
    "memory": "3 GB",
    "cpu": "1 vCPU",
    "executionRoleArn": "arn:aws:iam::123456789012:role/ec2TaskExecutionRole"
  }
}

```

6. Per utilizzare la funzionalità di chat in linea, è possibile evidenziare le linee e quindi scegliere l'icona a forma di stella.

Viene visualizzata la finestra di chat di Amazon Q Developer.

Inserire la richiesta.

Amazon Q Developer genera e quindi aggiorna il codice JSON.

Per accettare le modifiche, scegliere Accetta tutto

7. Scegliere Crea.

Aggiornamento di una definizione di attività Amazon ECS attraverso 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
- Requisiti di infrastruttura
- Dimensioni processo
- Dimensioni del contenitore
- Ruolo del processo
- Ruolo di esecuzione di attività
- Volumi e punti di montaggio dei container
- Registro privato

È possibile fare in modo che Amazon Q fornisca consigli quando si usa l'editor JSON. Per ulteriori informazioni, consulta [Utilizzo di Amazon Q Developer per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS](#)

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. Puoi aggiornare la compatibilità delle definizioni delle attività con una delle istanze AWS Fargate, Managed Instances, Amazon. EC2
7. Verifica le informazioni e scegli Aggiorna.
8. Se la definizione dell'attività viene utilizzata in un servizio, aggiorna il servizio con la definizione dell'attività aggiornata. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).

Amazon ECS console JSON editor

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 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 di definizione dell'attività con la console

È possibile annullare la registrazione di una definizione di attività in modo che non venga più visualizzata nelle chiamate API `ListTaskDefinition` o nella console quando si esegue un'attività o si aggiorna 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.

CloudFormation pile

Il comportamento seguente si applica alle definizioni di attività che sono state 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, consultare [Deleting a stack](#) nella Guida per l'utente di 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 2https://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 di definizione di attività Amazon ECS attraverso la console

Quando non è più necessaria una revisione specifica della definizione dell'attività in Amazon ECS, è possibile eliminarne la revisione.

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 lo stato della revisione delle definizioni di attività è `DELETE_IN_PROGRESS`, il nome della definizione di 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

La richiesta di eliminazione di una definizione di attività non verrà completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione di attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività autonome di Amazon ECS: la definizione di attività è necessaria affinché quest'ultima rimanga integra.
- Attività di servizio di Amazon ECS: la definizione di attività è necessaria affinché quest'ultima rimanga integra.
- Implementazioni e set di attività di servizio di Amazon ECS: la definizione di attività è necessaria quando si avvia un evento di dimensionamento per un'implementazione o un set di attività di Amazon ECS.

Se la definizione dell'`DELETE_IN_PROGRESS`attività rimane invariata, è possibile utilizzare la console o AWS CLI identificare e quindi interrompere 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.

- Implementazioni e set di attività di servizio di Amazon ECS: dopo l'eliminazione dell'implementazione o del set di attività, l'eliminazione della definizione di attività può richiedere fino a 24 ore.

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 <https://console.aws.amazon.com/ecs/versione 2>.
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 definizione di attività, selezionare le revisioni da eliminare, quindi selezionare Operazioni, Elimina.

Se Elimina non è disponibile, è necessario annullare la registrazione della definizione dell'attività.
6. Verificare le informazioni riportate nella casella di conferma Elimina, quindi selezionare Elimina per terminare.

Casi d'uso della definizione dell'attività di Amazon ECS

Scopri di più su come scrivere le definizioni delle attività per vari AWS servizi e funzionalità.

A seconda del carico di lavoro, è necessario impostare determinati parametri di definizione delle attività. Inoltre EC2, devi scegliere istanze specifiche progettate per il carico di lavoro.

Argomenti

- [Definizioni di attività Amazon ECS per carichi di lavoro GPU](#)
- [Definizioni di attività 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 di attività 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 AWSAWS Partner](#)
- [Utilizzo di immagini non AWS containerizzate in Amazon ECS](#)
- [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#)
- [Trasferimento di dati sensibili a un container 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. GPUs 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. GPUs 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 GPUs in considerazione a livello di contenitore. Amazon ECS pianifica le istanze di container disponibili che supportano GPUs e collegano contenitori fisici GPUs 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

Tipo di istanza	GPUs	Memoria GPU (GiB)	v CPUs	Memoria (GiB)
p4d.24xlarge	8	320	96	1152
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

Tipo di istanza	GPUs	Memoria GPU (GiB)	v CPUs	Memoria (GiB)
g6.xlarge	1	24	4	16
g6.2xlarge	1	24	8	32
g6.4xlarge	1	24	16	64
g6.8xlarge	1	24	32	128
g6.16.x grande	1	24	64	256
g6.12xlarge	4	96	48	192
g6.24xlarge	4	96	96	384
g6.48xlarge	8	192	192	768
g6.metal	8	192	192	768
gr6.4xlarge	1	24	16	128
g6e.xlarge	1	48	4	32
g6e.2xlarge	1	48	8	64
g6e.4xlarge	1	48	16	128
g6e.8xlarge	1	48	32	256
g6e16.xlarge	1	48	64	512
g6e12.xlarge	4	192	48	384
g6e24.xlarge	4	192	96	768
g6e48.xlarge	8	384	192	1536
gr6.8xlarge	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
```

Utilizzo GPU con Amazon ECS Managed Instances

Amazon ECS Managed Instances supporta l'elaborazione accelerata da GPU per carichi di lavoro come apprendimento automatico, elaborazione ad alte prestazioni ed elaborazione video tramite i seguenti tipi di istanze Amazon. EC2 Per ulteriori informazioni sui tipi di istanza supportati da Istanze gestite da Amazon ECS, consultare [Tipi di Istanze gestite da Amazon ECS](#).

Di seguito è riportato un sottoinsieme di tipi di istanze basate su GPU supportato su istanze gestite da Amazon ECS:

- g4dn: con tecnologia NVIDIA T4 GPUs, adatto per inferenza di machine learning, visione artificiale e applicazioni a uso intensivo di grafica.
- g5: con tecnologia NVIDIA A10G GPUs, offre prestazioni più elevate per applicazioni a uso intensivo di grafica e carichi di lavoro di machine learning.
- p3: con tecnologia NVIDIA V100 GPUs, progettato per l'elaborazione ad alte prestazioni e l'apprendimento deep learning.
- p4d: con tecnologia NVIDIA A100 GPUs, offre le massime prestazioni per l'apprendimento di machine learning e l'elaborazione ad alte prestazioni.

Quando utilizzi tipi di istanze abilitati per GPU con Istanze gestite da Amazon ECS, i driver NVIDIA e il toolkit CUDA sono preinstallati sull'istanza, semplificando l'esecuzione di carichi di lavoro accelerati da GPU.

Selezione di istanze abilitate per GPU

Per selezionare i tipi di istanza abilitati alla GPU per i carichi di lavoro di Istanze gestite da Amazon ECS, utilizzare l'oggetto `instanceRequirements` nel modello di lancio del provider di capacità. Il

seguente frammento mostra gli attributi che possono essere utilizzati per selezionare istanze abilitate per GPU.

```
{
  "instanceRequirements": {
    "acceleratorTypes": "gpu",
    "acceleratorCount": 1,
    "acceleratorManufacturers": ["nvidia"]
  }
}
```

Il seguente frammento mostra gli attributi che possono essere utilizzati per specificare i tipi di istanza abilitati per GPU nel modello di avvio.

```
{
  "instanceRequirements": {
    "allowedInstanceTypes": ["g4dn.xlarge", "p4de.24xlarge"]
  }
}
```

Immagini di container abilitati per GPU

Per utilizzarli GPUs nei contenitori, è necessario utilizzare immagini di container che contengano le librerie e gli strumenti GPU necessari. NVIDIA fornisce diverse immagini di container predefinite che puoi utilizzare come base per i carichi di lavoro della GPU, tra cui:

- `nvidia:cuda`: immagini di base con il toolkit CUDA per l'elaborazione tramite GPU.
- `tensorflow/tensorflow:latest-gpu`: TensorFlow con supporto GPU.
- `pytorch/pytorch:latest-cuda`: PyTorch con supporto GPU.

Per un esempio di definizione di attività per Amazon ECS su Amazon ECS Managed Instances che prevede l'uso di GPUs, consulta [Specificazione GPUs in una definizione di attività Amazon ECS](#)

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 immediati dei NVIDIA/CUDA driver su entrambi questi tipi di famiglie di istanze causeranno potenziali guasti del carico di lavoro della GPU.

Ti consigliamo di considerare quanto segue prima di iniziare a lavorare GPU 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 dei 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, consultare [Specialized Configurations with Docker](#). Amazon ECS imposta il valore della variabile di `NVIDIA_VISIBLE_DEVICES` ambiente in modo che sia un elenco dei dispositivi GPU assegnati da IDs Amazon ECS 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 create un'immagine del contenitore che non utilizza le immagini di NVIDIA/CUDA base, dovete impostare la variabile di runtime del NVIDIA_DRIVER_CAPABILITIES contenitore 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.

- GPU non sono supportati nei contenitori Windows.

Avvio di un'istanza di container GPU per Amazon ECS

Per utilizzare un'istanza GPU su Amazon ECS su Amazon EC2, 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 avvio.

- Creare un modello di avvio 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.

Utilizzare l'ID AMI ottenuto nel 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.

Aggiungere 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. Eseguire questo comando per ottenere l'ID AMI 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. Eseguire 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 Docker GPU, assicurati di indicare il numero di componenti richiesti dal contenitore nella definizione dell' GPUs 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 di Amazon ECS attraverso la nuova 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"  
}
```


La seguente definizione di attività di esempio mostra un TensorFlow contenitore che stampa il numero di elementi disponibili GPUs. L'attività viene eseguita su Istanze gestite da Amazon ECS, richiede una GPU e utilizza un'istanza `g4dn.xlarge`.

```
{
  "family": "tensorflow-gpu",
  "networkMode": "awsvpc",
  "executionRoleArn": "arn:aws:iam::account-id:role/ecsTaskExecutionRole",
  "containerDefinitions": [
    {
      "name": "tensorflow",
      "image": "tensorflow/tensorflow:latest-gpu",
      "essential": true,
      "command": [
        "python",
        "-c",
        "import tensorflow as tf; print('Num GPUs Available: ',
len(tf.config.list_physical_devices('GPU')))"
      ],
      "resourceRequirements": [
        {
          "type": "GPU",
          "value": "1"
        }
      ],
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/tensorflow-gpu",
          "awslogs-region": "region",
          "awslogs-stream-prefix": "ecs"
        }
      }
    }
  ],
  "requiresCompatibilities": [
    "MANAGED_INSTANCES"
  ],
  "cpu": "4096",
  "memory": "8192",
}
```

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 nfile=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. Impostare la variabile di ambiente NVIDIA_VISIBLE_DEVICES sul container. È possibile eseguire questa operazione specificando la variabile di ambiente nella definizione dell'attività. Per informazioni sui valori validi, consultare [GPU Enumeration](#) sul 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.pkrvars.hcl` o `overrides.auto.pkrvars.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 di attività 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 dei 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-xffmpeg-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/ renderD12 8 ,/dev/ dri/ renderD12 9
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.`

Specifica della transcodifica video in una definizione 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 dei 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](#).

Utilizzare 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 Pacifico (Osaka)
- Asia Pacifico (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- Europe (Paris)
- Europa (Stoccolma)
- Sud America (San Paolo)

Utilizza 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 Pacifico (Osaka)
- Asia Pacifico (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europa (Londra)
- 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

Specifica del machine learning 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 dei 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 [che sono 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](#).

Specifica del 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 di attività Amazon ECS per carichi di lavoro ARM a 64 bit

Amazon ECS supporta l'utilizzo di applicazioni ARM a 64 bit. È possibile eseguire le applicazioni sulla piattaforma con tecnologia [Processori AWS Graviton](#), 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 Fargate o EC2s
- 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 essere utilizzate Fluent Bit o CloudWatch per il monitoraggio.
- Per 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 la EC2, 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.

Specifica 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 si utilizza Fargate per le attività, è possibile visualizzare i log provenienti dai container. Se lo utilizzi EC2, puoi visualizzare diversi registri dei 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

Fargate

Se si utilizza Fargate per le tue attività, per attivare il driver di log logConfiguration è necessario aggiungere i parametri awslogs richiesti alla definizione di attività. Per ulteriori informazioni, consulta [Esempio di definizione dell'attività Amazon ECS: indirizzare i log a CloudWatch](#).

Per il container Windows su Fargate, eseguire una delle seguenti opzioni quando uno dei parametri di definizione dell'attività contiene caratteri speciali come & \ < > ^ |:

- Aggiungere un carattere escape (\) con virgolette doppie attorno all'intera stringa del parametro

Esempio

```
"awslogs-multiline-pattern": "\"^[|DEBUG|INFO|WARNING|ERROR\\\"\",
```

- Aggiungere un carattere escape (^) attorno a ogni carattere speciale

Esempio

```
"awslogs-multiline-pattern": "\"^[^|DEBUG^|INFO^|WARNING^|ERROR\",
```

EC2

Se utilizzi EC2 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 una personalizzata con almeno la versione 1.9.0-1 del pacchetto ecs-init. 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, consultare [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 policy IAM gestita per le istanze di container, consultare [Ruolo IAM delle istanze di container Amazon ECS](#).

Esempio di definizione dell'attività Amazon ECS: indirizzare i log a 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 log, consultare [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"
}

```

Fasi successive

- 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'implementazione di aggiornamenti continui di Amazon ECS](#).

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 l'immagine AWS for Fluent Bit oppure puoi utilizzare la tua immagine Fluentd o Fluent Bit.

Per impostazione predefinita, Amazon ECS configura la dipendenza del container in modo che il container FireLens si avvii prima di qualsiasi container che lo utilizza. Inoltre, il container FireLens si ferma dopo che tutti i container che lo utilizzano si fermano.

Per utilizzare questa funzionalità, devi creare un ruolo IAM per le tue attività che fornisca le autorizzazioni necessarie per utilizzare tutti AWS i servizi richiesti dalle attività. Ad esempio, se un container esegue il routing dei log a Firehose, allora l'attività 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.

Il processo potrebbe richiedere anche il ruolo di esecuzione del processo 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 Gestione dei segreti AWS a dati sensibili dalla tua configurazione di registro, devi includere il ruolo IAM di esecuzione dell'attività.
- Quando si utilizza un file di configurazione personalizzato ospitato in Amazon S3, il ruolo IAM per l'esecuzione dell'attività deve includere l'autorizzazione `s3:GetObject`.

Considera quanto segue quando utilizzi FireLens Amazon ECS:

- Consigliamo di aggiungere `my_service_` al log del nome del container in modo da poter distinguere facilmente i nomi dei container nella console.
- Per impostazione predefinita, Amazon ECS aggiunge una dipendenza dall'ordine iniziale del container tra i container dell'applicazione e il container FireLens. Quando si specifica un ordine del container tra i container dell'applicazione e il container FireLens, l'ordine di avvio predefinito del container viene sovrascritto.

- FireLens per Amazon ECS è supportato per le attività ospitate sia su Linux che AWS Fargate su Amazon EC2 su Linux. I container di 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](#).

- Puoi utilizzare CloudFormation modelli FireLens per configurare 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 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.
- Per le attività che utilizzano la modalità di rete `bridge`, 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 del cluster e della definizione dell'attività e l'Amazon Resource Name (ARN) del cluster come chiavi di metadati ai `stdout/stderr` log del contenitore. 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"
  }
}
```

Puoi configurare il FireLens contenitore in modo che venga eseguito come utente non root. Considera i seguenti aspetti:

- Per configurare il FireLens contenitore in modo che venga eseguito come utente non root, è necessario specificare l'utente in uno dei seguenti formati:
 - uid
 - uid:gid
 - uid:group

Per ulteriori informazioni sulla specificazione di un utente in una definizione di contenitore, consulta [ContainerDefinition](#) Amazon Elastic Container Service API Reference.

Il FireLens contenitore riceve i log delle applicazioni tramite un UNIX socket. L'agente Amazon ECS utilizza il uid per assegnare la proprietà della directory socket al FireLens contenitore.

- La configurazione del FireLens contenitore per l'esecuzione come utente non root è supportata nelle versioni di Amazon ECS Agent 1.96.0 e successive e nelle versioni AMI ottimizzate per Amazon ECS e successive. v20250716
- Quando si specifica un utente per il FireLens contenitore, questo uid deve essere univoco e non deve essere utilizzato per altri processi appartenenti ad altri contenitori nell'attività o nell'istanza del contenitore.

Per informazioni su come utilizzare più file di configurazione con Amazon ECS, inclusi i file ospitati o i file in Amazon S3, consultare [Init process for Fluent Bit on ECS, multi-config support](#).

Per informazioni sulle configurazioni di esempio, vedere [Esempio di definizione dell'attività Amazon ECS: indirizzare i log a FireLens](#).

Per ulteriori informazioni sulla configurazione dei log per un throughput elevato, vedere.

[Configurazione dei log di Amazon ECS per un throughput elevato](#)

Configurazione dei log di Amazon ECS per un throughput elevato

Quando si crea una definizione di attività, è possibile specificare il numero righe di log che sono stati salvati nella memoria specificando il valore nel `log-driver-buffer-limit`. Per ulteriori informazioni, consulta la pagina [Driver di registro di Fluentd](#) nella documentazione Docker.

Utilizzare questa opzione in caso di throughput elevato, perché Docker potrebbe esaurire la memoria buffer e scartare i messaggi 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 su qualsiasi EC2 tipo 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 di default è 1048576 righe di log.
- Il limite del buffer deve essere maggiore o uguale a 0 e inferiore a 536870912 righe di log.
- La quantità massima di memoria utilizzata per questo buffer è il prodotto della dimensione di ogni riga di log e della dimensione del buffer. Ad esempio, se le righe di log dell'applicazione sono in media di 2 KiB, un limite di buffer di 4096 utilizzerebbe al massimo 8 MiB. La quantità totale di memoria allocata a livello di attività deve essere superiore a quella allocata per tutti i container oltre al driver di log del buffer di memoria.

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 FireLens contenitore.

Note

Se utilizzi la modalità EC2 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 container dell'applicazione all'indirizzo IP FLUENT_HOST può iniziare a fallire dopo la modifica dell'indirizzo. Per ulteriori informazioni sul riavvio di singoli container, consultare [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

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, consulta [fluent-logger-golang](#) on GitHub.

Di seguito viene illustrata la sintassi per specificare il `log-driver-buffer-limit`. Sostituire `my_service_` con il nome del 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": [],
      "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 gli archivi di Fluent Bit immagini per Amazon ECS

AWS fornisce un'Fluent Bit immagine con i plugin sia per CloudWatch Logs che per Firehose. Si consiglia di utilizzare Fluent Bit come router di log perché dispone di 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 per l'alta disponibilità.

Galleria pubblica di Amazon ECR

L'Fluent Bit immagine AWS per è disponibile nella Amazon ECR Public Gallery. Questa è la posizione consigliata per scaricare l'Fluent Bit immagine AWS for 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

L'Fluent Bitimmagine AWS for nella galleria pubblica di Amazon ECR supporta il sistema operativo Amazon Linux con x86-64 architettura ARM64 or.

Puoi estrarre l'Fluent Bitimmagine AWS for 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 estrarre l'ultima versione stabile AWS per Fluent Bit l'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
```

AWS per 3.0.0 Fluent Bit

Oltre alle Fluent Bit versioni AWS for esistenti 2.x, AWS for Fluent Bit supporta una nuova versione principale. 3.0.0 La nuova versione principale include l'aggiornamento delle immagini da Amazon Linux 2 ad Amazon Linux 2023 e dalla Fluent Bit versione 1.9.10 a 4.1.1 Per ulteriori informazioni, consulta il [Fluent Bitrepository AWS for](#) su GitHub

Gli esempi seguenti mostrano i tag aggiornati AWS per Fluent Bit 3.0.0 le immagini:

È possibile utilizzare tag specifici dell'architettura per l'immagine for. AWS Fluent Bit Ad esempio, puoi estrarre un'immagine ARM64 dell'architettura usando un comando docker che segue questa sintassi.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:arm64-3.0.0
```

È possibile utilizzare tag multiarchitettura per l' AWS immagine for. Fluent Bit Ad esempio, puoi estrarre un'immagine con l'ultima versione di debug e il processo di inizializzazione utilizzando un comando docker che segue questa sintassi.

```
docker pull public.ecr.aws/aws-observability/aws-for-fluent-bit:init-debug-3.0.0
```

Windows

L'Fluent Bitimmagine AWS for 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 estrarre l'Fluent Bitimmagine AWS for 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 estrarre la più recente stable AWS for Fluent Bit image usando questo comando Docker CLI.

```
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. I seguenti comandi possono essere utilizzati per recuperare un'immagine URI e stabilire la disponibilità dell'immagine in una determinata area. Regione AWS

Linux

L'ultimo URI stabile AWS per Fluent Bit image può essere recuperato 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 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
```

Note

Se il comando fallisce o non viene generato alcun output, l'immagine non è disponibile nella cartella Regione AWS in cui viene chiamato il comando.

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 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 dell'attività Amazon ECS: indirizzare i log a FireLens

Per utilizzare il routing di log personalizzato con FireLens, è necessario specificare quanto segue nella definizione di attività:

- Un container router di log contenente una configurazione FireLens. 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 Console di gestione AWS, 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 di Amazon ECS attraverso la nuova 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 è in ascolto sulla porta 24224. Pertanto, per garantire che il FireLens log router non sia raggiungibile al di fuori dell'operazione, 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 `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.

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.

È possibile impostare `enable-ecs-log-metadata` su `false` se non si desiderano i metadati.

Il seguente esempio di definizione di attività definisce un contenitore di log router che utilizza Fluent Bit per indirizzare i log verso CloudWatch Logs. Definisce inoltre un container di applicazioni che utilizza una configurazione di log per instradare i log ad Amazon Data Firehose e imposta la memoria usata per il buffer degli eventi a 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": [],
      "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": "1048576"
            }
        },
        "memoryReservation": 100
    }
]
}

```

Le coppie chiave/valore specificate come opzioni nell'oggetto `logConfiguration` vengono utilizzate per generare la configurazione di output Fluentd o Fluent Bit. Di seguito è riportato un esempio di codice da una definizione di output Fluent Bit.

[OUTPUT]

```

Name    firehose
Match   app-firelens*
region  us-west-2
delivery_stream my-stream

```

Note

FireLens gestisce la `match` configurazione. Non specificare la configurazione `match` nella definizione dell'attività.

Utilizzo di un file di configurazione personalizzato

È possibile utilizzare 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, consultare [Fluentd Config File Syntax](#) e [YAML Configuration](#).

Nel file di configurazione personalizzato, per le attività che utilizzano la modalità di rete `bridge` o `awsvpc`, non è necessario impostare un input di inoltro Fluentd o Fluent Bit su TCP perché FireLens lo aggiunge alla configurazione di input.

La configurazione di FireLens 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 utilizzato da FireLens. 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"
        }
      }
    }
  ]
}
```

 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 Gestione dei segreti AWS 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, consulta [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 Console di gestione AWS, 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:

Sostituire 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 [Configurare istanze di container Amazon ECS per immagini Docker private](#).

Per utilizzare il registro privato

1. La definizione dell'attività deve avere un ruolo per l'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](#).

L'autenticazione del registro privato consente alle attività di Amazon ECS di estrarre immagini di container da registri privati esterni AWS (come Docker Hub, Quay.io o il tuo registro privato) che richiedono credenziali di autenticazione. Questa funzionalità utilizza Secrets Manager per archiviare in modo sicuro le credenziali del tuo registro, a cui viene poi fatto riferimento nella definizione dell'attività utilizzando il parametro `repositoryCredentials`.

Per ulteriori informazioni sulla configurazione dell'autenticazione del registro privato, consulta [Utilizzo di immagini non AWS container in 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 contengono credenziali del tuo registro privato. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- `secretsmanager:GetSecretValue`: necessario per recuperare le credenziali del registro privato da Secrets Manager.
- `kms:Decrypt`: obbligatorio solo se il segreto 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret_name",
        "arn:aws:kms:us-east-1:111122223333:key/key_id"
      ]
    }
  ]
}
```

2. Utilizzalo Gestione dei segreti AWS per creare un segreto per le credenziali del tuo registro privato. Per ulteriori informazioni sulla creazione di un segreto, consultare [Create an Gestione dei segreti AWS secret](#) nella Guida per l'utente di Gestione dei segreti AWS .

Inserire le credenziali del registro privato utilizzando il formato seguente:

```
{
```

```
"username" : "privateRegistryUsername",  
"password" : "privateRegistryPassword"  
}
```

3. Registra una definizione di attività. Per ulteriori informazioni, consulta [the section called “Creazione di una definizione di attività attraverso la nuova console”](#).

Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container

È possibile abilitare una politica di riavvio per ogni container essenziale e non essenziale indicato nella definizione dell'attività per superare più rapidamente i guasti temporanei e mantenere la disponibilità delle attività. Quando si abilita una politica di riavvio per un container, Amazon ECS può riavviare il container se esce, senza dover sostituire l'attività.

Per impostazione predefinita, le policy di riavvio non sono abilitate per i container. Quando abiliti una politica di riavvio per un container, è possibile specificare i codici di uscita in base ai quali il container non verrà riavviato. Questi possono essere codici di uscita che indicano l'esito positivo, come il codice di uscita 0, che non richiede un riavvio. È inoltre possibile specificare per quanto tempo un container 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 [Specifica 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 riavvii di un container. Per ulteriori informazioni sugli endpoint dei metadati dell'attività, consultare [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#) e [Endpoint metadati delle attività Amazon ECS versione 4 per le 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

Considera quanto segue prima di abilitare una politica di riavvio per il container:

- Le politiche di riavvio non sono supportate per i container 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](#).
- Se utilizzi la modalità EC2 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'firelensConfigurationoggetto nella definizione del contenitore). Questo perché FLUENT_HOST è un indirizzo IP dinamico e può cambiare dopo un riavvio. La registrazione diretta dal container dell'applicazione all'indirizzo IP FLUENT_HOST può iniziare a fallire dopo la modifica dell'indirizzo. Per ulteriori informazioni su FLUENT_HOST, consultare [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 viene meno o non è più in esecuzione, il container non verrà riavviato.
- Il periodo di tentativo di riavvio definito nella politica determina il periodo di tempo (in secondi) per cui il container deve essere eseguito prima che Amazon ECS riavvii un container.

Specifica di una politica di riavvio dei container in una definizione di attività Amazon ECS

Per specificare una politica di riavvio per un container in una definizione di attività, specificare l'oggetto `restartPolicy` all'interno della definizione del container. Per ulteriori informazioni sull'oggetto `restartPolicy`, consultare [Politica di riavvio](#).

Di seguito è riportata una definizione di attività usando i container Linux su Fargate per configurare un server Web. La definizione del container include l'oggetto `restartPolicy`, con `enabled` su `true` per abilitare una politica di riavvio per il container. Il container deve funzionare per 180 secondi prima di poter essere riavviato e non verrà riavviato se esce con il codice di uscita 0, che indica l'esito positivo.

```
{
  "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 dell'attività con l'oggetto `restartPolicy` nella definizione di un container, puoi eseguire un'attività o creare un servizio con tale definizione dell'attività. Per ulteriori

informazioni, consultare [Esecuzione di un'applicazione come attività Amazon ECS](#) e [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#).

Trasferimento di dati sensibili a un container 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 Gestione dei segreti AWS 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 Gestione dei segreti AWS nostro AWS Systems Manager Parameter Store per archiviare materiali segreti

È consigliabile archiviare in modo sicuro le chiavi API, le credenziali di database e altri materiali segreti in Secrets Manager o come parametri crittografati 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. Tuttavia, Secrets Manager offre anche la possibilità di ruotare in automatico i segreti, di generare segreti casuali e di condividere i segreti tra 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: interrompere l'attività, quindi avviarla. Per ulteriori informazioni, consultare [Interruzione di un'attività di Amazon ECS](#) e [Esecuzione di un'applicazione come attività Amazon ECS](#).
- Servizio: aggiornare il servizio e utilizzare la nuova opzione di implementazione forzata. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).

Recuperare i dati da un bucket Amazon S3 crittografato

È consigliabile archiviare i segreti in un bucket Amazon S3 crittografato e utilizzare i ruoli di 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 rivelati durante l'esecuzione di `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 fuga di dati a causa delle variabili di ambiente, è consigliabile utilizzare un contenitore secondario che legga i segreti e li scriva su un volume condiviso. Gestione dei segreti AWS 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, consultare 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.

Passare una singola variabile di ambiente a un container Amazon ECS

Important

Ti consigliamo di archiviare i dati sensibili nei parametri Gestione dei segreti AWS segreti o in AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container 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"
        }
      ],
      ...
    }
  ],
  ...
}
```

```
}
```

Passare le variabili di ambiente a un container Amazon ECS

Important

Ti consigliamo di archiviare i dati sensibili nei Gestione dei segreti AWS segreti o nei parametri del AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container 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 parametro `environmentFiles` su container Windows e Windows su Fargate.

È possibile creare un file di variabili di ambiente e archivarlo in Amazon S3 per passare le variabili di ambiente al proprio container.

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 applica un limite di dimensione alle variabili di ambiente, ma un file di variabili di ambiente di grandi dimensioni potrebbe riempire 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 del processo.

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.
- Per utilizzare questa funzionalità con le autorizzazioni aggiuntive per Amazon S3, è necessario il ruolo di esecuzione dell'attività. 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, consultare [Set environment variables \(-e, --env, --env-file\)](#) nella documentazione Docker.

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 a Fargate:
 - Il file viene gestito in maniera 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 container.
- 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"
        }
      ],
      ...
    }
  ],
  ...
}
```

Passare i segreti di Secrets Manager in modo programmatico in Amazon ECS

Invece di eseguire la codifica fissa delle 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 Gestione dei segreti AWS .
- 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 Gestione dei segreti AWS](#) nella Guida per l'utente di Gestione dei segreti AWS .

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, consultare [Retrieve secrets from Gestione dei segreti AWS](#) nella Guida per l'utente di Gestione dei segreti AWS .

Per recuperare i dati sensibili memorizzati in Gestione dei segreti AWS, consulta [Esempi di codice da Gestione dei segreti AWS utilizzare AWS SDKs nella libreria di](#) codici AWS SDK Code Examples.

Passare 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 di Systems Manager Parameter Store verrà aggiornato in futuro, l'applicazione recupererà automaticamente l'ultima versione.

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 EC2, 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.

Passare i segreti di Secrets Manager tramite le variabili di ambiente Amazon ECS

Quando si inserisce un segreto come variabile di ambiente, è possibile specificare il contenuto completo 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, consultare [What's in a Secrets Manager secret?](#) nella Guida per l'utente di Gestione dei segreti AWS .

Quando si utilizza una variabile di ambiente per inserire un segreto di Gestione dei segreti in un container, è opportuno 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 container, i log del container 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 1.3.0 o successiva. Per informazioni, consulta [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, consulta [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.

- Utilizzare gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza e connettersi 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 Gestione dei segreti AWS . Per ulteriori informazioni sull'utilizzo di Secrets Manager e Amazon VPC, consultare [How to connect to Secrets Manager service within a 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 l'esecuzione di un ruolo di attività con autorizzazioni aggiuntive per Secrets Manager. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Creazione del segreto di Gestione dei segreti AWS

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni, consulta [Creare un Gestione dei segreti AWS segreto](#) nella Guida per l'Gestione dei segreti AWS 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.

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 Gestione dei segreti AWS](#) nella Guida per l'utente di Gestione dei segreti AWS .

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 segreti completi

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a più segreti di Secrets Manager.

```

{
  "containerDefinitions": [{
    "secrets": [
      {
        "name": "environment_variable_name1",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-AbCdEf"
      },
      {
        "name": "environment_variable_name2",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-abcdef"
      },
      {
        "name": "environment_variable_name3",
        "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 `$environment_variable_name1`, `$environment_variable_name2` e `$environment_variable_name3`.

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 sulla creazione di una definizione di attività con il segreto specificato in una variabile di ambiente, consultare [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#).

Passare 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.

Tenere in considerazione ciò che segue durante l'utilizzo di una variabile di ambiente per inserire un segreto di Systems Manager in un container.

- 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 1.3.0 o successiva. Per informazioni, consulta [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, consulta [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 l'esecuzione di un ruolo di 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

Nella definizione del container all'interno della definizione dell'attività, 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 sulla creazione di una definizione di attività con il segreto specificato in una variabile di ambiente, consultare [Creazione di una definizione di attività di Amazon ECS attraverso la nuova 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.

Passare i segreti per la configurazione di registrazione di Amazon ECS

È possibile utilizzare il parametro `secretOptions` in `logConfiguration` per passare dati sensibili utilizzati per la registrazione.

È possibile archiviare il messaggio in Secrets Manager o Systems Manager.

Utilizzare 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. Per ulteriori informazioni sulla creazione dei segreti, consultare [Create an Gestione dei segreti AWS](#).

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 sulla creazione di una definizione di attività con il segreto specificato in una variabile di ambiente, consultare [Creazione di una definizione di attività di Amazon ECS attraverso la nuova 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"
      }]
    }]
  }]
}
```

Specifica di dati sensibili utilizzando segreti di Secrets Manager in Amazon ECS

Amazon ECS ti consente di iniettare dati sensibili nei tuoi contenitori archiviando i dati sensibili in modo Gestione dei segreti AWS segreto e quindi facendone riferimento nella definizione del contenitore. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container Amazon ECS](#).

Per scoprire come creare un segreto di Secrets Manager, fare riferimento al segreto in una definizione di attività di Amazon ECS e quindi verificarne il funzionamento eseguendo una query sulla variabile di ambiente all'interno di un container che mostra i contenuti 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](#).
- L'utente dispone delle autorizzazioni IAM necessarie per creare le risorse di 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 creeremmo 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 Gestione dei segreti AWS segreto](#) nella Guida per l'Gestione dei segreti AWS utente.

Le key/value coppie da memorizzare in questo segreto sono il valore della variabile di ambiente nel contenitore 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.

Fase 2: aggiungere le autorizzazioni segrete al ruolo di esecuzione di attività

Per consentire ad Amazon ECS di recuperare i dati sensibili dal segreto di Secrets Manager, è necessario disporre delle autorizzazioni dei segreti per il ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta [Autorizzazioni per 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 alla <https://console.aws.amazon.com/ecs/v2>.
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, inserire il seguente testo JSON della definizione di attività, assicurandosi di specificare l'ARN completo del segreto di Secrets Manager creato nella fase 1 e il ruolo dell'esecuzione del processo aggiornato nella fase 2. Scegli 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 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 EC2, vedere [the section called “Creazione di un cluster per i EC2 carichi di lavoro 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 EC2 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, selezionare Cluster e quindi il cluster che è stato 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 Cluster.
3. Nella pagina Cluster, scegliere 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 `ECSecretsTutorial`. 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 dell'attività di Amazon ECS per istanze gestite da Amazon ECS

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 la capacità. 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 la capacità 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 di attività per Istanze gestite da Amazon ECS.

Family

`family`

Tipo: stringa

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.

Capacity

Quando si registra una definizione di attività, è possibile specificare la capacità che Amazon ECS deve convalidare per tale definizione. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client. Per ulteriori informazioni, consultare [Amazon ECS launch types](#).

Il parametro seguente è permesso in una definizione di attività.

`requiresCompatibilities`

Tipo: array di stringhe

Obbligatorio: no

Valori validi: MANAGED_INSTANCES

La capacità per la 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 per Istanze gestite da Amazon ECS.

Ruolo del processo

`taskRoleArn`

■Tipo: stringa

Obbligatorio: no

Quando registri una definizione di attività, puoi fornire un ruolo di attività per un ruolo IAM che consenta ai contenitori inclusi nell'attività di chiamare per tuo conto i dati specificati nelle politiche associate. AWS APIs Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Ruolo per l'esecuzione del processo

`executionRoleArn`

Tipo: String

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. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Note

Il ruolo IAM di esecuzione di attività è necessario a seconda dei requisiti dell'attività. Il ruolo è necessario per l'estrazione di immagini ECR private e l'utilizzo del driver di registro. `awslogs`

Modalità di rete

`networkMode`

─Tipo: stringa

Obbligatorio: no

Impostazione predefinita: `awsvpc`

La modalità di rete da utilizzare per i contenitori inclusi nell'operazione. Per le attività Amazon ECS ospitate su Istanze gestite Amazon ECS, i valori validi sono `awsvpc` e `host`. Se non viene specificata alcuna modalità di rete, la modalità di rete predefinita è `awsvpc`.

Se la modalità di rete è `awsvpc`, l'operazione ignora l'isolamento della rete e i container utilizzano direttamente lo stack di rete dell'host.

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 [Rete di attività di Amazon ECS per istanze gestite Amazon ECS](#).

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.

Piattaforma di runtime

`operatingSystemFamily`

▪Tipo: stringa

Obbligatorio: no

Di default: LINUX

Quando registri una definizione di attività, devi specificare la famiglia del sistema operativo.

Il valore valido per questo campo è LINUX.

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: String

Obbligatorio: condizionale

Quando registri una definizione di attività, devi specificare l'architettura della CPU. I valori validi sono X86_64 e ARM64.

Se non specifichi un valore, Amazon ECS tenta di inserire attività sull'architettura CPU disponibile in base alla configurazione del provider di capacità. Per garantire che le attività vengano posizionate su un'architettura CPU specifica, specifica un valore per `cpuArchitecture` nella definizione dell'attività.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Per ulteriori informazioni su ARM64, consultare [the section called “Definizioni di 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: String

Obbligatorio: condizionale

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 10 v. CPUs

`memory`

Tipo: String

Obbligatorio: condizionale

Il limite rigido di memoria da presentare al processo. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibytes (MiB) o gigabytes (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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

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 di Amazon ECS attraverso la nuova console](#).

Argomenti

- [Archiviazione temporanea](#)
- [Modalità IPC](#)
- [Modalità PID](#)
- [Configurazione del proxy](#)
- [Tag](#)
- [Acceleratore Elastic Inference \(obsoleto\)](#)
- [Vincoli di posizionamento](#)
- [Volumi](#)

Archiviazione temporanea

`ephemeralStorage`

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: oggetto [EphemeralStorage](#)

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”](#).

Modalità IPC

ipcMode

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

─Tipo: stringa

Obbligatorio: no

Il namespace 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 viene specificato alcun valore, la condivisione dello spazio dei nomi delle risorse IPC dipende dalla configurazione di runtime del contenitore.

Modalità PID

pidMode

─Tipo: stringa

Obbligatorio: no

Il namespace del processo da utilizzare per i container nell'attività. I valori validi sono `host` o `task`. Se specificato, tutti i contenitori che `host` si trovano 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 stata specificata l'`task`, tutti i container all'interno dell'attività specificata condividono lo stesso namespace del processo. Se non è stato specificato alcun valore, l'impostazione di default è un namespace privato.

Se viene utilizzata la modalità `host PID`, esiste un maggiore rischio di esposizione a spazi dei nomi di attività indesiderati.

Configurazione del proxy

`proxyConfiguration`

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: oggetto [ProxyConfiguration](#)

Obbligatorio: no

I dettagli di configurazione del proxy App Mesh.

Tag

I metadati applicati alla definizione di un'attività per aiutare a categorizzarli e organizzarli. Ciascun tag è formato da una chiave e da un valore facoltativo. Li definisci entrambi.

Ai tag si applicano le seguenti limitazioni di base:

- Numero massimo di tag per risorsa: 50
- Per ogni risorsa, ogni chiave di tag deve essere unica e ogni chiave di tag può avere un solo valore.
- Lunghezza massima della chiave: 128 caratteri Unicode in formato UTF-8

- Lunghezza massima del valore: 256 caratteri Unicode in formato UTF-8
- Se lo schema di assegnazione dei tag viene utilizzato in più servizi e risorse , tieni presente che altri servizi potrebbero prevedere limitazioni sui caratteri consentiti. I caratteri generalmente consentiti sono: lettere, numeri e spazi rappresentabili in UTF-8 e i seguenti caratteri: + - = . _ : / @ .
- Per le chiavi e i valori dei tag viene fatta la distinzione tra maiuscole e minuscole.
- Non utilizzare alcuna combinazione maiuscola o minuscola `aws:AWS:` , ad esempio un prefisso per chiavi o valori, poiché è riservata all'uso. AWS Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non vengono conteggiati per il limite del numero di tag per risorsa.

key

─Tipo: stringa

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

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).

Acceleratore Elastic Inference (obsoleto)

Note

Amazon Elastic Inference (EI) non è più disponibile per i clienti.

inferenceAccelerator

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: oggetto [InferenceAccelerator](#)

Obbligatorio: no

Gli accelerator di inferenza elastica da utilizzare per i container nell'attività.

Vincoli di posizionamento

placementConstraints

Tipo: matrice di oggetti [TaskDefinitionPlacementConstraint](#)

Obbligatorio: no

Una matrice di oggetti vincolo di posizionamento da utilizzare per l'attività. 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).

Amazon ECS supporta i vincoli di posizionamento `distinctInstance` e `memberOf` per le attività in esecuzione sulle Istanze gestite da Amazon ECS. I seguenti attributi sono supportati per le attività che utilizzano il vincolo di posizionamento `memberOf`:

- `ecs.subnet-id`
- `ecs.availability-zone`
- `ecs.cpu-architecture`
- `ecs.instance-type`

Per ulteriori informazioni sul posizionamento dei vincoli, consultare [Definisci quali istanze di container utilizza Amazon ECS per le attività](#).

Volumi

Quando registri una definizione di attività, puoi facoltativamente specificare un elenco di volumi per le tue attività. Ciò consente di utilizzare i volumi di dati nelle attività.

Per ulteriori informazioni sui tipi di volume e altri parametri, consultare [the section called “Opzioni di archiviazione per le attività”](#).

name

Tipo: stringa

Obbligatorio: sì

Nome del volume. Può essere costituito da un massimo di 255 lettere (maiuscole e minuscole), numeri e trattini. Nel parametro `sourceVolume` della definizione del container `mountPoints` viene fatto riferimento a questo nome.

host

Tipo: oggetto [HostVolumeProperties](#)

Obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi host di montaggi vincolati. Il contenuto del parametro `host` determina se il volume host di montaggio vincolato viene mantenuto nell'istanza di container host e il punto in cui viene archiviato. Se il `host` parametro è vuoto, il sistema assegna un percorso host per il volume di dati. Tuttavia, non è garantito che i dati siano mantenuti dopo l'interruzione del funzionamento dei container associati.

dockerVolumeConfiguration

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: oggetto [DockerVolumeConfiguration](#)

Obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker.

efsVolumeConfiguration

Tipo: oggetto [EFSVolumedi configurazione](#)

Obbligatorio: no

Questo parametro viene specificato quando si utilizza un file system Amazon EFS per l'archiviazione delle attività.

fsxWindowsFileServerVolumeConfiguration

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: oggetto [FSxWindowsFileServerVolumeConfiguration](#)

Obbligatorio: no

Questo parametro viene specificato quando utilizzi il file system Amazon FSx for Windows File Server per l'archiviazione delle attività.

configuredAtLaunch

Tipo: Booleano

Obbligatorio: no

Specificare se il volume deve essere configurato al momento dell'avvio. Viene utilizzato per creare volumi Amazon EBS per attività autonome o attività create come parte di un servizio. Ogni revisione della definizione di attività può avere un solo volume configurato all'avvio nella configurazione del volume.

Definizioni del container

Quando registri una definizione di attività, devi specificare un elenco di definizioni del container che vengono trasmesse al daemon Docker in un'istanza di container. I seguenti parametri sono consentiti in una definizione del container.

Argomenti

- [Name](#)
- [Immagine](#)
- [Memoria](#)
- [CPU](#)
- [Mappature di porte](#)
- [Credenziali del repository privato](#)
- [Essential](#)
- [Punto di ingresso](#)
- [Comando](#)
- [Directory di lavoro](#)
- [Parametri avanzati di definizione del container](#)
- [Parametri Linux](#)

Name

name

Tipo: stringa

Obbligatorio: sì

Il nome di un container. Sono consentite fino a 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

Obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene trasmessa direttamente al daemon Docker. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. 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 mappato su Image nel comando creazione container e nel parametro IMAGE del docker di esecuzione del comando.

- 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 contenitore Amazon ECS utilizza il latest tag per estrarre l'immagine specificata.
- Gli aggiornamenti successivi a un'immagine del repository non vengono propagati 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*.
- Le immagini in repository ufficiali su Docker Hub utilizzano un singolo nome (ad esempio ubuntu o mongo).
- Le immagini in altri repository su Docker Hub vengono qualificate con un nome di 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: String

Valori validi: enabled|disabled

Obbligatorio: no

Specificare se Amazon ECS risolverà il tag dell'immagine del container fornito nella definizione del container in un digest di immagini. Per impostazione predefinita, questo comportamento è

enabled. Se si imposta il valore per un container come disabled, Amazon ECS non risolverà il tag dell'immagine del container in un digest e utilizzerà l'URI dell'immagine originale specificato nella definizione del container per l'implementazione. Per ulteriori informazioni sulla risoluzione dell'immagine del container, consultare [Risoluzione dell'immagine del container](#).

Memoria

memory

Tipo: Integer

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 a Memory nella creazione container del comando del docker e l'opzione `--memory` per 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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

memoryReservation

Tipo: Integer

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 conservare la memoria del container 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 a `MemoryReservation` nella creazione container del comando del docker e l'opzione `--memory-reservation` per `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.

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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

CPU

cpu

Tipo: Integer

Obbligatorio: no

Il numero di unità cpu prenotate per il container. Questo parametro è mappato a CpuShares nella creazione container del comando del docker e l'opzione `--cpu-shares` per docker run.

Questo campo è facoltativo per le attività che utilizzano fornitori di EC2 capacità e l'unico requisito è che la quantità totale di CPU riservata per tutti i contenitori all'interno di un'attività sia inferiore al valore a livello di attivitàcpu.

 Note

Puoi determinare il numero di unità CPU disponibili per tipo di EC2 istanza moltiplicando la v CPUs elencata per quel tipo di istanza nella pagina dei dettagli di [Amazon EC2 Instances](#) per 1.024.

I container Linux condividono unità CPU non assegnate con altri container nell'istanza di container con lo stesso rapporto della quantità assegnata. Ad esempio, se si esegue un'attività di container singolo in un tipo di istanza single-core con 512 unità di CPU specificate per quel container e questa è l'unica attività in esecuzione nell'istanza di container, tale container potrebbe utilizzare tutte le quote di 1.024 unità di CPU in qualsiasi momento. Tuttavia, se hai 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. Inoltre, ogni container potrebbe passare a un maggiore utilizzo della CPU se l'altro container non la utilizza. Se entrambe le attività fossero sempre attive al 100%, sarebbero limitate a 512 unità CPU.

Sulle istanze di container Linux, il daemon Docker nell'istanza di container utilizza il valore CPU per calcolare i relativi rapporti di quote di CPU per i container in esecuzione. Per ulteriori informazioni, consulta [CPU share constraint](#) nella documentazione Docker. Il valore minimo valido per il valore della quota di CPU ammesso dal kernel Linux è 2. Tuttavia, il parametro CPU non è obbligatorio e puoi usare valori di CPU minori di 2 nelle definizioni del container. Per i valori di CPU inferiori a 2 (incluso un valore nullo), il comportamento varia in base alla versione dell'agente del container Amazon ECS:

- Versioni dell'agente inferiori o uguali a 1.1.0: i valori di CPU null e zero vengono trasmessi a Docker come 0 e successivamente convertiti in 1.024 quote di CPU. I valori di CPU di 1 vengono trasmessi a Docker come 1 e il kernel Linux li converte in due quote di CPU.
- Versioni dell'agente superiori o uguali a 1.2.0: i valori di CPU null, zero e 1 vengono trasmessi a Docker come 2.

Sulle istanze di container Windows, il limite di CPU viene applicato come un limite assoluto o una quota. I container Windows hanno accesso solo alla quantità di CPU descritta nella definizione dell'attività. Un valore di CPU null o zero viene passato a Docker come 0, che Windows interpreta come 1% di una CPU.

Mappature di porte

portMappings

Tipo: array di oggetti

Obbligatorio: no

Le mappature delle porte espongono le porte di rete del container al mondo esterno. Ciò consente ai client di accedere all'applicazione. Viene anche utilizzato per la comunicazione tra container all'interno 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 container viene mappata automaticamente su una casuale con numero alto sull'host.

La maggior parte dei campi di questo parametro (inclusi `containerPort`, `hostPort` e `protocol`) è associata a `PortBindings` nella sezione creazione container del comando e l'opzione `--publish` per `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: stringa

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: numero intero

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 rete awsvpc, utilizzare containerPort per specificare le porte esposte.

containerPortRange

─Tipo: stringa

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 di rete `awsvpc`.
- 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 proxy docker nel file di configurazione del daemon Docker quando disponi di un numero elevato di porte.

[Per ulteriori informazioni, consulta il numero #11185 su GitHub](#)

Per informazioni sulla modalità di disattivazione del proxy docker nel file di configurazione del daemon Docker, consulta [Daemon Docker](#) nella Guida per lo sviluppatore di Amazon ECS.

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: stringa

Obbligatorio: no

L'intervallo di numeri di porta sull'host utilizzato con il collegamento di rete. Questo viene assegnato da Docker e consegnato dall'agente Amazon ECS.

hostPort

Tipo: Integer

Obbligatorio: no

Il numero di porta nell'istanza di container per prenotare per il container.

hostPort può essere lasciato vuoto o deve essere lo stesso valore di containerPort.

L'intervallo di porte temporanee predefinite per Docker versione 1.6.0 e successive è elencato nell'istanza in `/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 prenotate di default sono 22 per SSH, le porte Docker 2375 e 2376 e le porte 51678-51680 dell'agente del container di Amazon ECS. 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: String

Obbligatorio: no, necessario per configurare 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 di 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 visualizzati entrambi i campi obbligatori per Service Connect e VPC Lattice.

```
"portMappings": [  
  {  
    "name": string,  
    "containerPort": integer  
  }  
]
```

protocol

─Tipo: stringa

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](#)

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: String

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 Console di gestione AWS, 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"  
    }  
  }  
]
```

Essential

essential

Tipo: Booleano

Obbligatorio: no

Se il parametro `essential` di un container è contrassegnato come `true` e tale container dà esito negativo o si interrompe per qualsiasi motivo, tutti gli altri container che fanno parte del processo vengono interrotti. 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. Se disponi di un'applicazione composta da più container, raggruppa i container utilizzati per scopi comuni in componenti e separa i diversi componenti in più definizioni di attività. Per ulteriori informazioni, consulta [the section called “Progetta l'applicazione”](#).

Punto di ingresso

entryPoint

Tipo: array di stringhe

Obbligatorio: no

Il punto di ingresso che viene trasmesso al container. Questo parametro è mappato a `Entrypoint` nella creazione container del comando del docker e l'opzione `--entrypoint` per `docker run`.

```
"entryPoint": ["string", ...]
```

Comando

command

Tipo: array di stringhe

Obbligatorio: no

Il comando che viene inviato al container. Questo parametro è mappato a `Cmd` nella creazione container del comando del docker e il parametro `COMMAND` per docker run. In caso di più argomenti, ognuno di questi deve essere una stringa separata nell'array.

```
"command": ["string", ...]
```

Directory di lavoro

workingDirectory

─Tipo: stringa

Obbligatorio: no

La directory di lavoro nel container in cui eseguire i comandi. Questo parametro è mappato a `WorkingDir` nella creazione container del comando del docker e l'opzione `--workdir` per docker run.

Parametri avanzati di definizione del container

I seguenti parametri avanzati di definizione del container forniscono funzionalità estese al comando di esecuzione del docker che viene utilizzato per lanciare container nelle istanze di container di Amazon ECS.

Argomenti

- [Politica di riavvio](#)
- [Controllo dell'integrità](#)
- [Ambiente](#)
- [Sicurezza](#)
- [Impostazioni di rete](#)

- [Archiviazione e registrazione](#)
- [Requisiti per le risorse](#)
- [Timeout container](#)
- [Dipendenze per i container](#)
- [Controlli di sistema](#)
- [Interactive](#)
- [Pseudoterminale](#)

Politica di riavvio

`restartPolicy`

La politica di riavvio del container e i parametri di configurazione associati. Quando configuri una politica di riavvio per un container, Amazon ECS può riavviare il container senza dover sostituire l'attività. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

`enabled`

Tipo: Booleano

Obbligatorio: sì

Specificare se è abilitata una politica di riavvio per il container.

`ignoredExitCodes`

Tipo: array di numeri interi

Obbligatorio: no

Un elenco di codici di uscita che Amazon ECS ignorerà e non tenterà di riavviare. È possibile specificare fino a 50 codici di uscita dei container. Per impostazione predefinita, Amazon ECS non ignora alcun codice di uscita.

`restartAttemptPeriod`

Tipo: Integer

Obbligatorio: no

Un periodo di tempo (in secondi) durante il quale il container deve essere eseguito prima che venga tentato un riavvio. Un container può essere riavviato solo una volta ogni `restartAttemptPeriod` secondi. Se un container non è in grado di funzionare per questo periodo di tempo ed esce in anticipo, non verrà riavviato. È possibile specificare un `restartAttemptPeriod` minimo di 60 secondi e un `restartAttemptPeriod` massimo di 1.800 secondi. Per impostazione predefinita, un container deve funzionare per 300 secondi prima di poter essere riavviato.

Controllo dell'integrità

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 dell'integrità 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 Console di gestione AWS, 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 Console di gestione AWS 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 predefinito 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

Obbligatorio: no

Il numero di unità `cpu` che l'agente del container Amazon ECS riserverà per il container. Su Linux, questo parametro è mappato a `CpuShares` nella sezione [Create a container](#).

Questo campo è facoltativo per le attività eseguite su istanze gestite da Amazon ECS. 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.

I container Linux condividono unità CPU non assegnate con altri container nell'istanza di container con lo stesso rapporto della quantità assegnata. 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.

Sulle istanze di container Linux, il daemon Docker nell'istanza di container utilizza il valore CPU per calcolare i relativi rapporti di quote di CPU per i container in esecuzione. Il valore minimo valido per la quota di CPU ammesso dal kernel Linux è 2, mentre quello massimo valido per la quota di CPU ammesso dal kernel Linux è 262.144. Tuttavia, il parametro CPU non è obbligatorio e puoi usare valori di CPU minori di 2 e superiori a 262.144 nelle definizioni del container. Per i valori di CPU minori di 2 (incluso un valore null) e superiori a 262.144, il comportamento varia in base alla versione dell'agente del container Amazon ECS:

Per ulteriori esempi, consulta [Modalità di gestione delle risorse di CPU e memoria di Amazon ECS](#).

gpu

Tipo: oggetto [ResourceRequirement](#)

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](#).

Elastic Inference accelerator

 Note

Questo parametro non è supportato per i container ospitati su Istanze gestite Amazon ECS.

Tipo: oggetto [ResourceRequirement](#)

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 Elastic Inference \(obsoleto\)”](#).

`essential`

Tipo: Booleano

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 omissso, 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

Obbligatorio: no

Il punto di ingresso che viene trasmesso al container.

```
"entryPoint": ["string", ...]
```

command

Tipo: array di stringhe

Obbligatorio: no

Il comando che viene inviato al container. Questo parametro è mappato a `Cmd` nella creazione container del comando e il parametro `COMMAND` per `docker run`. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

workingDirectory

─Tipo: stringa

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

Obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'opzione `--env-file` al comando esecuzione del docker.

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 [Passare una singola variabile di ambiente a un container Amazon ECS](#).

`value`

Tipo: stringa

Obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

`type`

Tipo: stringa

Obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è s3.

`environment`

Tipo: array di oggetti

Obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è mappato a Env nella creazione container del comando del docker e l'opzione `--env` al comando per docker run.

 Important

Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

`name`

Tipo: String

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

`value`

Tipo: String

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

Obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container Amazon ECS](#).

`name`

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

`valueFrom`

Tipo: stringa

Obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del Gestione dei segreti AWS 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"  
  }  
]
```

Sicurezza

privileged

Tipo: Booleano

Obbligatorio: no

Se il parametro è `true`, al container vengono assegnati privilegi elevati nell'istanza di container host (simile all'utente `root`). Questo parametro è mappato a `Privileged` nella creazione container del comando del docker e l'opzione `--privileged` per docker run.

user

─Tipo: stringa

Obbligatorio: no

L'utente da usare all'interno del container. Questo parametro è mappato a `User` nella creazione container del comando del docker e l'opzione `--user` per docker run.

Important

Quando si eseguono attività utilizzando la modalità di rete `host`, non eseguire container utilizzando l'utente `root` (UID 0). È consigliabile utilizzare un utente non root per maggior sicurezza.

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`

`readonlyRootFilesystem`

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è `true`, al container viene assegnato il file system root di sola lettura. Questo parametro è mappato a `ReadonlyRootfs` nella creazione container del comando del `docker` e l'opzione `--read-only` per `docker run`.

`dockerSecurityOptions`

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: array di stringhe

Obbligatorio: no

Un elenco di stringhe per fornire etichette personalizzate per SELinux sistemi di sicurezza AppArmor multilivello. Questo campo non è valido per container in attività che utilizzano Fargate.

`ulimits`

Tipo: matrice di oggetti [Ulimit](#)

Obbligatorio: no

Un elenco di valori `ulimits` da impostare nel container. Se viene specificato un valore `ulimit` in una definizione di attività, questo sostituisce i valori predefiniti impostati da Docker. Questo parametro è mappato a `Ulimits` nella creazione container del comando del `docker` e l'opzione `--ulimit` per `docker run`. Valori di denominazione validi vengono visualizzati nel tipo di dati [Ulimit](#).

I processi di Amazon ECS ospitati su Fargate utilizzano i valori del limite di risorse di default impostati dal sistema operativo, ad eccezione del parametro del limite di risorse `nofile` che

viene sovrascritto da Fargate. Il limite di risorse `nofile` imposta una restrizione sul numero di file aperti che un container può utilizzare. Il limite flessibile `nofile` di default è 1024 mentre il limite rigido predefinito è 65535.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container. Per controllare la versione Docker Remote API nell'istanza di container, accedi all'istanza di container ed esegui il seguente comando: `sudo docker version --format '{{.Server.APIVersion}}'`

`dockerLabels`

Tipo: mappatura stringa a stringa

Obbligatorio: no

Una key/value mappa di etichette da aggiungere al contenitore. Questo parametro è mappato a `Labels` nella creazione container del comando del docker e l'opzione `--label` per docker run.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

```
"dockerLabels": {"string": "string"
...}
```

Impostazioni di rete

`disableNetworking`

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è true, le reti sono disabilitate all'interno del container.

Il valore predefinito è `false`.

```
"disableNetworking": true|false
```

links

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: array di stringhe

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 costrutto `name:internalName` è analogo a `name:alias` nei collegamenti Docker. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e trattini bassi.

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

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

■Tipo: stringa

Obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato a `Hostname` nella creazione container del docker e l'opzione `--hostname` per docker run.

```
"hostname": "string"
```

dnsServers

Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: array di stringhe

Obbligatorio: no

Un elenco di server DNS presentato al container.

```
"dnsServers": ["string", ...]
```

extraHosts

Note

Questo parametro non è supportato per le attività che utilizzano la modalità di rete `awsvpc`.

Tipo: array di oggetti

Obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato a ExtraHosts nella creazione container del comando del docker e l'opzione `--add-host` per docker run.

```
"extraHosts": [  
  {  
    "hostname": "string",  
    "ipAddress": "string"  
  },  
  ...  
]
```

hostname

Tipo: String

Obbligatorio: sì, quando si utilizzano extraHosts

Il nome host da utilizzare nella voce `/etc/hosts`.

ipAddress

Tipo: String

Obbligatorio: sì, quando si utilizzano extraHosts

L'indirizzo IP da utilizzare nella voce `/etc/hosts`.

Archiviazione e registrazione

readonlyRootFilesystem

Tipo: Booleano

Obbligatorio: no

Se il parametro è `true`, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro è mappato a `ReadonlyRootfs` nella creazione container del comando del docker e l'opzione `--read-only` per docker run.

Il valore predefinito è `false`.

```
"readonlyRootFilesystem": true|false
```

mountPoints

Tipo: array di oggetti

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai `Volumes` nella creazione container dell'API Docker e l'opzione `--volume` per `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

sourceVolume

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

containerPath

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

readOnly

Tipo: Booleano

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, lascia il valore predefinito di `false`.

volumesFrom

Tipo: array di oggetti

Obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato a `VolumesFrom` nella creazione container del comando del docker e l'opzione `--volumes-from` per docker run.

`sourceContainer`

Tipo: String

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

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](#)

Obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta [Esempio di definizione di attività di Amazon ECS](#).

Questo parametro è mappato a `LogConfig` nella creazione container del comando del docker e l'opzione `--log-driver` per docker run. Per impostazione predefinita, i container utilizzano lo stesso driver di log utilizzato dal daemon Docker. Tuttavia, il container può utilizzare un driver di

log diverso da quello del daemon Docker, specificando un driver di log con questo parametro nella definizione del container. 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 log disponibili per il daemon Docker.
- Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

```
"logConfiguration": {
  "logDriver": "awslogs","splunk", "awsfirelens",
  "options": {"string": "string"
    ...},
  "secretOptions": [{
    "name": "string",
    "valueFrom": "string"
  }]
}
```

logDriver

Tipo: String

Valori validi: "awslogs", "splunk", "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, 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 CloudWatch Logs, vedere. [Inviare i log di Amazon ECS a CloudWatch](#)

Per ulteriori informazioni sull'utilizzo del driver di log awsfirelens, consultare [Inviare i log di Amazon ECS a un servizio o AWSAWS Partner](#).

 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

Obbligatorio: no

La key/value mappa delle opzioni di configurazione da inviare al driver di registro.

Le opzioni che è possibile specificare dipendono dal driver di log. Alcune delle opzioni che puoi specificare quando utilizzi il `awslogs` router per indirizzare i log ad Amazon CloudWatch includono:

`awslogs-create-group`

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`

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`

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: 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`

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`

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.

mode

Obbligatorio: no

Valori validi: non-blocking | blocking

Questa opzione definisce la modalità di consegna dei messaggi di log dal container al driver di log `awslogs`. La modalità di distribuzione scelta influisce sulla disponibilità dell'applicazione quando il flusso di log dal container viene interrotto.

Se si utilizza la `blocking` modalità e il flusso dei log verso CloudWatch viene interrotto, le chiamate provenienti dal codice contenitore su cui scrivere verso e gli `stdout` stream verranno bloccate. `stderr` 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 smetta di rispondere quando non è possibile inviare i log a CloudWatch. Ti consigliamo di utilizzare questa modalità se vuoi garantire la disponibilità del servizio nonostante una certa perdita di log. Per ulteriori informazioni, consultare [Preventing log loss with non-blocking mode in the awslogs container log driver](#).

max-buffer-size

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 router di log `splunk`, è necessario specificare `splunk-token` e `splunk-url`.

Quando si utilizza il router di `awsfirelens` log per indirizzare i log verso una AWS Partner Network destinazione Servizio AWS o 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 aiutarti a risolvere potenziali problemi di perdita di log perché un throughput elevato potrebbe comportare l'esaurimento della memoria per il buffer all'interno di Docker. Per ulteriori informazioni, consulta [the section called “Configurazione dei log per un throughput elevato”](#).

Le altre opzioni che è possibile specificare quando si utilizza `awsfirelens` per indirizzare i log 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 si esportano i log in flusso di dati Amazon Kinesis, è possibile specificare Regione AWS con `region` 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 si esportano i log in Amazon S3, è possibile specificare il bucket utilizzando l'opzione `bucket`. È possibile 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

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 [Trasferimento di dati sensibili a un container Amazon ECS](#).

`name`

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

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: oggetto [FirelensConfiguration](#)

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

Obbligatorio: no

La key/value mappa delle opzioni da usare durante la configurazione del log router. 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 dell'attività Amazon ECS: indirizzare i log a FireLens](#).

type

Tipo: stringa

Obbligatorio: sì

Il router di log da utilizzare. I valori validi sono fluentd o fluentbit.

Requisiti per le risorse

resourceRequirements

Tipo: matrice di oggetti [ResourceRequirement](#)

Obbligatorio: no

Il tipo e la quantità di una risorsa da assegnare a un container. L'unica risorsa supportata è la GPU.

type

Tipo: stringa

Obbligatorio: sì

Il tipo di risorsa da assegnare a un container. Il valore supportato è GPU.

value

Tipo: stringa

Obbligatorio: sì

Il valore per il tipo di risorsa specificato.

Se si utilizza il tipo GPU, il valore è il numero di GPUs fisiche che l'agente del container Amazon ECS prenota per il container. Il numero di GPUs quelli riservati a 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.

GPUs non sono disponibili per le attività in esecuzione su Fargate.

Timeout container

`startTimeout`

Tipo: Integer

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 600 secondi (10 minuti).

`stopTimeout`

Tipo: Integer

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 parametro non è specificato, viene utilizzato il valore predefinito di 30 secondi. Il valore massimo è di 86.400 secondi (24 ore).

Dipendenze per i container

dependsOn

Tipo: matrice di oggetti [ContainerDependency](#)

Obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un contenitore 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 successive (Linux) o 1.0.0 (Windows).

```
"dependsOn": [  
  {  
    "containerName": "string",  
    "condition": "string"  
  }  
]
```

containerName

Tipo: stringa

Obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

`condition`

Tipo: stringa

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 è confermata solo all'avvio dell'attività.

Controlli di sistema

`systemControls`

Tipo: oggetto [SystemControl](#)

Obbligatorio: no

Un elenco di parametri kernel associati a un namespace da impostare nel container. Questo parametro è mappato a `Sysctl`s nella creazione container del comando del `docker` e l'opzione `--sysctl` per `docker run`. 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 sono stati impostati `systemControls` per qualsiasi container, si applicano a tutti i container nell'attività. Se sono stati impostati diversi `systemControls` per più container in un'unica attività, il container che viene avviato per ultimo determina quale `systemControls` diventa effettivo.

Se stai impostando un namespace 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` del namespace IPC non sono supportati.
- Per le attività che utilizzano la modalità `IPC task`, i valori di `systemControls` del namespace IPC si applicano a tutti i container all'interno di un'attività.

```
"systemControls": [  
  {  
    "namespace": "string",  
    "value": "string"  
  }  
]
```

namespace

─Tipo: stringa

Obbligatorio: no

Il parametro `kernel` associato a un namespace per il quale impostare un parametro `value`.

Valori del namespace 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 namespace di rete validi: `Sysctls` che iniziano con `"net.*"`. In Fargate, vengono accettati solo i namespace `Sysctls` esistenti all'interno del container.

Tutti questi valori sono supportati da Fargate.

`value`

─Tipo: stringa

Obbligatorio: no

Il valore per il parametro `kernel` associato a un namespace specificato in `namespace`.

Interactive

`interactive`

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è `true`, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di `stdin` o `tty`. Questo parametro è mappato a `OpenStdin` nella creazione container del comando del `docker` e l'opzione `--interactive` per `docker run`.

Il valore predefinito è `false`.

Pseudoterminale

`pseudoTerminal`

Tipo: Booleano

Obbligatorio: no

Quando il parametro è `true`, è allocato un TTY. Questo parametro è mappato a `Tty` nella creazione container del comando del `docker` e l'opzione `--tty` per `docker run`.

Il valore predefinito è `false`.

Parametri Linux

`linuxParameters`

Tipo: oggetto [LinuxParameters](#)

Obbligatorio: no

Le modifiche specifiche di Linux che vengono applicate al container, ad esempio le funzionalità kernel Linux.

capabilities

Tipo: oggetto [KernelCapabilities](#)

Obbligatorio: no

La funzionalità di Linux per il container che vengono aggiunte alla configurazione predefinita fornita da Docker o eliminate da questa.

devices

Tipo: matrice di oggetti [Device](#)

Obbligatorio: no

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato a Devices nella creazione container del comando del docker e l'opzione `--device` per docker run.

initProcessEnabled

Tipo: Booleano

Obbligatorio: no

Esegui un processo `init` nel container che inoltra segnali e raccoglie i processi. Questo parametro è mappato all'opzione `--init` su docker run.

maxSwap

 Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: Integer

Obbligatorio: no

La quantità totale di memoria di swap (in MiB) che un container può utilizzare. Questo parametro viene convertito nell'opzione `--memory-swap` in `docker run` dove il valore sarebbe la somma della memoria del container più il valore `maxSwap`.

`swappiness`

 Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: Integer

Obbligatorio: no

In questo modo è possibile ottimizzare il comportamento `swappiness` di memoria di un container. Un valore `swappiness` delle cause 0 per cui lo swap non avviene a meno che non sia assolutamente necessario. Un valore `swappiness` di 100 produrrà lo swap delle pagine in modo molto aggressivo. I valori validi 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'opzione `--memory-swappiness` su `docker run`.

`sharedMemorySize`

 Note

Questo parametro non è supportato per le attività eseguite su Istanze gestite Amazon ECS.

Tipo: Integer

Obbligatorio: no

Le dimensioni (in MiB) del volume `/dev/shm`. Questo parametro è mappato all'opzione `--shm-size` su `docker run`.

`tmpfs`

Tipo: matrice di oggetti [Tmpfs](#)

Obbligatorio: no

Percorso del container, opzioni di montaggio e dimensioni (in MiB) del montaggio del volume tmpfs. Questo parametro è mappato all'opzione `--tmpfs` su `docker run`.

Parametri di definizione di attività Amazon ECS per 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 i tipi 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 Fargate.

Family

`family`

Tipo: stringa

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.

Capacity

Quando si registra una definizione di attività, è possibile specificare la capacità che Amazon ECS deve convalidare per tale definizione. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client.

Il parametro seguente è permesso in una definizione di attività.

requiresCompatibilities

Tipo: array di stringhe

Obbligatorio: no

Valori validi: FARGATE

La capacità per la 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 di Fargate.

Ruolo del processo

taskRoleArn

─Tipo: stringa

Obbligatorio: no

Quando registri una definizione di attività, puoi fornire un ruolo di attività per un ruolo IAM che consenta ai contenitori inclusi nell'attività di chiamare per tuo conto i dati specificati nelle politiche associate. AWS APIs Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Ruolo per l'esecuzione del processo

executionRoleArn

Tipo: String

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

Obbligatorio: sì

La modalità di rete Docker da utilizzare per i container nel processo. Per i processi Amazon ECS ospitati su Fargate, è necessaria la modalità di rete awsvpc.

Piattaforma di runtime

operatingSystemFamily

Tipo: String

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_2025_FULL, WINDOWS_SERVER_2025_CORE, WINDOWS_SERVER_2022_FULL, WINDOWS_SERVER_2022_CORE, WINDOWS_SERVER_2019_FULL e WINDOWS_SERVER_2019_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: String

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 dispone di attività Linux, è possibile impostare il valore su `ARM64`. Per ulteriori informazioni, consulta [the section called “Definizioni di 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`.

Il parametro seguente è permesso in una definizione di attività:

`cpu`

Tipo: stringa

Obbligatorio: sì

Note

I parametri di CPU e memoria a livello di attività sono obbligatori e utilizzati per determinare il tipo e la dimensione dell'istanza su cui vengono eseguite le attività. Per le attività Windows, questi valori non vengono applicati durante il runtime, poiché Windows non dispone di un meccanismo nativo in grado di imporre facilmente limiti collettivi di risorse su un gruppo di container. Se si desidera applicare i limiti delle risorse, consigliamo di utilizzare le risorse a livello di container per quelli di 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 (`v`). CPUs 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 validi per il parametro `memory`. 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: sì

Note

I parametri di CPU e memoria a livello di attività sono obbligatori e utilizzati per determinare il tipo e la dimensione dell'istanza su cui vengono eseguite le attività. Per le attività Windows, questi valori non vengono applicati durante il runtime, poiché Windows non dispone di un meccanismo nativo in grado di imporre facilmente limiti collettivi di risorse su un gruppo di container. Se si desidera applicare i limiti delle risorse, consigliamo di utilizzare le risorse a livello di container per quelli di Windows.

Il limite rigido di memoria da presentare al processo. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibytes (MiB) o gigabytes (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 validi per il parametro cpu:

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

Valore di memoria (in MiB, con valore equivalente approssimativo in GB)	Valore CPU	Sistemi operativi supportati per Fargate
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), 6144 (6 GB), 7168 (7 GB), 8192 (8 GB)	1024 (1 vCPU)	Linux, Windows
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  Note Questa opzione richiede la piattaforma Linux 1.4.0 o successiva.	8192 (8 vCPU)	Linux

Valore di memoria (in MiB, con valore equivalente approssimativo in GB)	Valore CPU	Sistemi operativi supportati per Fargate
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 registri una definizione di attività, devi specificare un elenco di definizioni del container che vengono trasmesse al daemon Docker in un'istanza di container. 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

- [Name](#)
- [Immagine](#)
- [Memoria](#)
- [Mappature di porte](#)

- [Credenziali del repository privato](#)

Name

name

Tipo: stringa

Obbligatorio: sì

Il nome di un container. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini e trattini bassi. 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

Obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene trasmessa direttamente al daemon Docker. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. 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 mappato su Image nel comando creazione container e nel parametro IMAGE del docker di esecuzione del comando.

- 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 contenitore Amazon ECS utilizza il latest tag per estrarre l'immagine specificata.
- Gli aggiornamenti successivi a un'immagine del repository non vengono propagati 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`.
- Le immagini in repository ufficiali su Docker Hub utilizzano un singolo nome (ad esempio `ubuntu` o `mongo`).
- Le immagini in altri repository su Docker Hub vengono qualificate con un nome di 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: String

Valori validi: `enabled`|`disabled`

Obbligatorio: no

Specificare se Amazon ECS risolverà il tag dell'immagine del container fornito nella definizione del container in un digest di immagini. Per impostazione predefinita, questo comportamento è `enabled`. Se si imposta il valore per un container come `disabled`, Amazon ECS non risolverà il tag dell'immagine del container in un digest e utilizzerà l'URI dell'immagine originale specificato nella definizione del container per l'implementazione. Per ulteriori informazioni sulla risoluzione dell'immagine del container, consultare [Risoluzione dell'immagine del container](#).

Memoria

memory

Tipo: Integer

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 a Memory nella creazione container del comando del docker e l'opzione `--memory` per 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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

memoryReservation

Tipo: Integer

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 conservare la memoria del container 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 a `MemoryReservation` nella creazione container del comando del docker e l'opzione `--memory-reservation` per 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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

Mappature di porte

portMappings

Tipo: array di oggetti

Obbligatorio: no

Le mappature delle porte espongono le porte di rete del container al mondo esterno. Ciò consente ai client di accedere all'applicazione. Viene anche utilizzato per la comunicazione tra container all'interno 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 container viene mappata automaticamente su una 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 (inclusi `containerPort`, `hostPort` e `protocol`) è associata a `PortBindings` nella sezione creazione container del comando e l'opzione `--publish` per `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: stringa

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: numero intero

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 rete `awsvpc`, utilizzare `containerPort` per specificare le porte esposte.

Per i container Windows su Fargate, non è possibile utilizzare la porta 3150 per `containerPort`. Questo perché la porta è riservata.

`containerPortRange`

■Tipo: stringa

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 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.
- 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 proxy docker nel file di configurazione del daemon Docker quando disponi di un numero elevato di porte.

[Per ulteriori informazioni, consulta il numero #11185 su GitHub](#)

Per informazioni sulla modalità di disattivazione del proxy docker nel file di configurazione del daemon Docker, consulta [Daemon Docker](#) nella Guida per lo sviluppatore di Amazon ECS.

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: stringa

Obbligatorio: no

L'intervallo di numeri di porta sull'host utilizzato con il collegamento di rete. Questo viene assegnato da Docker e consegnato dall'agente Amazon ECS.

hostPort

Tipo: Integer

Obbligatorio: no

Il numero di porta nell'istanza di container per prenotare per il container.

hostPort può essere lasciato vuoto o deve essere lo stesso valore di containerPort.

L'intervallo di porte temporanee predefinite per Docker versione 1.6.0 e successive è elencato nell'istanza in `/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 prenotate di default sono 22 per SSH, le porte Docker 2375 e 2376 e le porte 51678-51680 dell'agente del container di Amazon ECS. 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: String

Obbligatorio: no, necessario per configurare 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 di 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 visualizzati entrambi i campi obbligatori per Service Connect e VPC Lattice.

```
"portMappings": [  
  {  
    "name": string,  
    "containerPort": integer  
  }  
]
```

protocol

─Tipo: stringa

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](#)

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: String

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 Console di gestione AWS, 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 del container forniscono funzionalità estese al comando di esecuzione del docker che viene utilizzato per lanciare container nelle istanze di container di Amazon ECS.

Argomenti

- [Politica di riavvio](#)
- [Controllo dell'integrità](#)
- [Ambiente](#)

- [Impostazioni di rete](#)
- [Archiviazione e registrazione](#)
- [Sicurezza](#)
- [Limiti delle risorse](#)
- [Etichette Docker](#)

Politica di riavvio

`restartPolicy`

La politica di riavvio del container e i parametri di configurazione associati. Quando configuri una politica di riavvio per un container, Amazon ECS può riavviare il container senza dover sostituire l'attività. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

`enabled`

Tipo: Booleano

Obbligatorio: sì

Specificare se è abilitata una politica di riavvio per il container.

`ignoredExitCodes`

Tipo: array di numeri interi

Obbligatorio: no

Un elenco di codici di uscita che Amazon ECS ignorerà e non tenterà di riavviare. È possibile specificare fino a 50 codici di uscita dei container. Per impostazione predefinita, Amazon ECS non ignora alcun codice di uscita.

`restartAttemptPeriod`

Tipo: Integer

Obbligatorio: no

Un periodo di tempo (in secondi) durante il quale il container deve essere eseguito prima che venga tentato un riavvio. Un container può essere riavviato solo una volta ogni `restartAttemptPeriod` secondi. Se un container non è in grado di funzionare per

questo periodo di tempo ed esce in anticipo, non verrà riavviato. È possibile specificare un `restartAttemptPeriod` minimo di 60 secondi e un `restartAttemptPeriod` massimo di 1.800 secondi. Per impostazione predefinita, un container deve funzionare per 300 secondi prima di poter essere riavviato.

Controllo dell'integrità

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 dell'integrità 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 Console di gestione AWS, 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 Console di gestione AWS 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 predefinito 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

Obbligatorio: no

Il numero di unità `cpu` che l'agente del container Amazon ECS riserverà per il container. Su Linux, questo parametro è mappato a `CpuShares` nella sezione [Creazione di un container](#).

Questo campo è facoltativo solo per le attività che utilizzano 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.

I container Linux condividono unità CPU non assegnate con altri container nell'istanza di container con lo stesso rapporto della quantità assegnata. 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.

Sulle istanze di container Linux, il daemon Docker nell'istanza di container utilizza il valore CPU per calcolare i relativi rapporti di quote di CPU per i container in esecuzione. Il valore minimo valido per la quota di CPU ammesso dal kernel Linux è 2, mentre quello massimo valido per la quota di CPU ammesso dal kernel Linux è 262.144. Tuttavia, il parametro CPU non è obbligatorio e puoi usare valori di CPU minori di 2 e superiori a 262.144 nelle definizioni del container. Per i valori di CPU minori di 2 (incluso un valore null) e superiori a 262.144, il comportamento varia in base alla versione dell'agente del container Amazon ECS:

Sulle istanze di container Windows, la quota di CPU viene applicata come assoluta. I container Windows hanno accesso solo alla quantità di CPU specificata nella definizione delle attività. Un valore di CPU null o zero viene trasmesso a Docker come 0, che Windows interpreta come l'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 container ospitati su Fargate.

Tipo: oggetto [ResourceRequirement](#)

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](#).

Elastic Inference accelerator

Questo parametro non è supportato per i container ospitati su Fargate.

Tipo: oggetto [ResourceRequirement](#)

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 Elastic Inference \(obsoleto\)”](#).

`essential`

Tipo: Booleano

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

Obbligatorio: no

Il punto di ingresso che viene trasmesso al container.

```
"entryPoint": ["string", ...]
```

command

Tipo: array di stringhe

Obbligatorio: no

Il comando che viene inviato al container. Questo parametro è mappato a `Cmd` nella creazione container del comando e il parametro `COMMAND` per `docker run`. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

workingDirectory

─Tipo: stringa

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

Obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'opzione `--env-file` al comando esecuzione del docker.

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 [Passare una singola variabile di ambiente a un container Amazon ECS](#).

`value`

Tipo: stringa

Obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

`type`

Tipo: stringa

Obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è s3.

`environment`

Tipo: array di oggetti

Obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è mappato a Env nella creazione container del comando del docker e l'opzione `--env` al comando per docker run.

 Important

Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

`name`

Tipo: String

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

value

Tipo: String

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

Obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container Amazon ECS](#).

name

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del Gestione dei segreti AWS 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 eseguite su Fargate.

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è true, le reti sono disabilitate 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 di rete awsvpc.

Tipo: array di stringhe

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 costrutto `name:internalName` è analogo a `name:alias` nei collegamenti Docker. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e trattini bassi.

 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: stringa

Obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato a Hostname nella creazione container del docker e l'opzione `--hostname` per docker run.

 Note

Se utilizzi la modalità di rete `awsvpc`, il parametro `hostname` non è supportato.

```
"hostname": "string"
```

dnsServers

Non è supportato per le attività eseguite su Fargate.

Tipo: array di stringhe

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 di rete `awsvpc`.

Tipo: array di oggetti

Obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato a `ExtraHosts` nella creazione container del comando del docker e l'opzione `--add-host` per docker run.

```
"extraHosts": [  
  {  
    "hostname": "string",  
    "ipAddress": "string"  
  },  
  ...  
]
```

hostname

Tipo: String

Obbligatorio: sì, quando si utilizzano `extraHosts`

Il nome host da utilizzare nella voce `/etc/hosts`.

ipAddress

Tipo: String

Obbligatorio: sì, quando si utilizzano `extraHosts`

L'indirizzo IP da utilizzare nella voce `/etc/hosts`.

Archiviazione e registrazione

readonlyRootFilesystem

Tipo: Booleano

Obbligatorio: no

Se il parametro è true, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro è mappato a `ReadonlyRootFs` nella creazione container del comando del docker e l'opzione `--read-only` per 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

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai `Volumes` nella creazione container dell'API Docker e l'opzione `--volume` per docker run.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

sourceVolume

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

containerPath

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

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

Obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato a `VolumesFrom` nella creazione container del comando del docker e l'opzione `--volumes-from` per docker run.

`sourceContainer`

Tipo: String

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

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](#)

Obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta [Esempio di definizione di attività di Amazon ECS](#).

Questo parametro è mappato a LogConfig nella creazione container del comando del docker e l'opzione `--log-driver` per docker run. Per impostazione predefinita, i container utilizzano lo stesso driver di log utilizzato dal daemon Docker. Tuttavia, il container può utilizzare un driver di log diverso da quello del daemon Docker, specificando un driver di log con questo parametro nella definizione del container. 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 log disponibili per il daemon Docker.
- Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.
- È necessario installare qualsiasi software aggiuntivo al di fuori dell'attività. Ad esempio, gli aggregatori di output Fluentd o un host remoto che esegue Logstash per inviare i log Gelf.

```
"logConfiguration": {
    "logDriver": "awslogs","fluentd","gelf","json-
file","journald","splunk","syslog","awsfirelens",
    "options": {"string": "string"
    ...},
    "secretOptions": [{
        "name": "string",
        "valueFrom": "string"
    }]
}
```

```
}
```

logDriver

Tipo: String

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, 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 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

Obbligatorio: no

La key/value mappa delle opzioni di configurazione da inviare al driver di registro.

Le opzioni che è possibile specificare dipendono dal driver di log. Alcune delle opzioni che puoi specificare quando utilizzi il `awslogs` router per indirizzare i log ad Amazon CloudWatch includono:

`awslogs-create-group`

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`

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`

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: 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`

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

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 log supportati.

`mode`

Obbligatorio: no

Valori validi: `non-blocking` | `blocking`

Questa opzione definisce la modalità di consegna dei messaggi di log dal container al driver di log specificato utilizzando `logDriver`. La modalità di distribuzione scelta influisce sulla disponibilità dell'applicazione quando il flusso di log dal container viene interrotto.

Se si utilizza la modalità `blocking` e il flusso di log viene interrotto, le chiamate provenienti dal codice del container e utilizzate per la scrittura sui flussi `stdout` e `stderr` 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 la mancata risposta dell'applicazione quando non è possibile inviare i log. Ti consigliamo di utilizzare questa modalità se vuoi garantire la disponibilità del servizio nonostante una certa perdita di log. Per ulteriori informazioni, consultare [Preventing log loss with non-blocking mode in the awslogs container log driver](#).

È possibile impostare un valore predefinito `mode` per tutti i container in un Regione AWS specifico utilizzando le impostazioni dell'account `defaultLogDriverMode`. Se non si specifica l'opzione `mode` in `logConfiguration` o non si configura l'impostazione dell'account, Amazon ECS utilizzerà per impostazione predefinita la modalità non-blocking. Per ulteriori informazioni sulle impostazioni dell'account, consultare [Modalità driver log predefinita](#).

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. La quantità totale di memoria allocata a livello di attività deve essere superiore a quella allocata per tutti i container oltre al driver di log del buffer di memoria.

 Note

Il 25 giugno 2025, Amazon ECS ha cambiato la modalità predefinita del driver di log da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto al logging. Per continuare a utilizzare la modalità `blocking` dopo questa modifica, procedere in uno dei seguenti modi:

- Impostare l'opzione `mode` nella definizione del container `logConfiguration` come `blocking`.
- Impostare le impostazioni dell'account `defaultLogDriverMode` su `blocking`.

`max-buffer-size`

Obbligatorio: no

Valore predefinito: 10m

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 router di log `splunk`, è necessario specificare `splunk-token` e `splunk-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'`log-driver-buffer-limit` opzione per limitare il numero di righe di registro memorizzate nel buffer prima di essere inviate al contenitore del log router. Può aiutarti a risolvere potenziali problemi di perdita di log perché un throughput elevato potrebbe comportare l'esaurimento della memoria per il buffer all'interno di Docker. Per ulteriori informazioni, consulta [the section called “Configurazione dei log per un throughput elevato”](#).

Le altre opzioni che è possibile specificare quando si utilizza `awsfirelens` per indirizzare i log 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 si esportano i log in flusso di dati Amazon Kinesis, è possibile specificare Regione AWS con `region` 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 si esportano i log in Amazon S3, è possibile specificare il bucket utilizzando l'opzione `bucket`. È possibile 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

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 [Trasferimento di dati sensibili a un container Amazon ECS](#).

name

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

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: oggetto [FirelensConfiguration](#)

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

Obbligatorio: no

La key/value mappa delle opzioni da usare durante la configurazione del log router. 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 dell'attività Amazon ECS: indirizzare i log a FireLens](#).

type

Tipo: stringa

Obbligatorio: sì

Il router di log da utilizzare. I valori validi sono fluentd o fluentbit.

Sicurezza

Per ulteriori informazioni sulla sicurezza di container, consultare [Best practice per la sicurezza delle attività e dei container di Amazon ECS](#).

credentialSpecs

Tipo: array di stringhe

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.

`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](#).

`user`

─Tipo: stringa

Obbligatorio: no

L'utente da usare all'interno del container. Questo parametro è mappato a `User` nella creazione container del comando del docker e l'opzione `--user` per 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

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 a `Ulimits` nella creazione container del comando del docker e l'opzione `--ulimit` per 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: String

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: numero intero

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` di `ulimit`.

`softLimit`

Tipo: numero intero

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 del `type` di `ulimit`.

Etichette Docker

`dockerLabels`

Tipo: mappatura stringa a stringa

Obbligatorio: no

Una key/value mappa di etichette da aggiungere al contenitore. Questo parametro è mappato a `Labels` nella creazione container del comando del `docker` e l'opzione `--label` per `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 di Amazon ECS attraverso la nuova console](#).

Argomenti

- [Parametri Linux](#)
- [Dipendenze per i container](#)

- [Timeout del container](#)
- [Controlli di sistema](#)
- [Interactive](#)
- [Pseudoterminale](#)

Parametri Linux

linuxParameters

Tipo: oggetto [LinuxParameters](#)

Obbligatorio: no

Linux-opzioni specifiche applicate al contenitore, ad esempio [KernelCapabilities](#).

Note

Questo parametro non è supportato per i container Windows.

```
"linuxParameters": {  
  "capabilities": {  
    "add": ["string", ...],  
    "drop": ["string", ...]  
  }  
}
```

capabilities

Tipo: oggetto [KernelCapabilities](#)

Obbligatorio: no

Le funzionalità di Linux per il container che vengono eliminate dalla configurazione predefinita fornita da Docker. Per ulteriori informazioni su queste funzionalità di Linux, consulta la pagina del manuale Linux [capabilities\(7\)](#).

add

Tipo: array di stringhe

Valori validi: "SYS_PTRACE"

Campo obbligatorio: no

Le funzionalità di Linux per il container da aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato a CapAdd nella creazione container del comando del docker e l'opzione `--cap-add` per 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

Le funzionalità di Linux per il container da eliminare dalla configurazione predefinita fornita da Docker. Questo parametro è mappato a CapDrop nella creazione container del comando del docker e l'opzione `--cap-drop` per docker run.

devices

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato a Devices nella creazione container del comando del docker e l'opzione `--device` per docker run.

Note

Il parametro `devices` non è supportato quando si utilizza il tipo di avvio Fargate.

Tipo: matrice di oggetti [Device](#)

Obbligatorio: no

hostPath

Il percorso per il dispositivo nell'istanza di container dell'host.

Tipo: stringa

Obbligatorio: sì

containerPath

Il percorso nel container in cui esporre il dispositivo dell'host.

■Tipo: stringa

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: array 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'opzione `--init` su `docker run`.

Questo parametro richiede la versione 1.25 o successiva di Docker Remote API sull'istanza di container.

maxSwap

Non è supportato per le attività eseguite su Fargate.

La quantità totale di memoria di swap (in MiB) che un container può utilizzare. Questo parametro viene convertito nell'opzione `--memory-swap` in `docker run` dove il valore sarebbe la somma della memoria del container più il valore `maxSwap`.

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 omissso, 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'opzione `--shm-size` su `docker run`.

Note

Se si utilizzano attività che ricorrono a Fargate, il parametro `sharedMemorySize` non è supportato.

Tipo: numero intero

`tmpfs`

Il percorso del container, le opzioni di montaggio e le dimensioni massime (in MiB) del montaggio `tmpfs`. Questo parametro è mappato all'opzione `--tmpfs` su `docker run`.

Tipo: matrice di oggetti [Tmpfs](#)

Obbligatorio: no

`containerPath`

Percorso assoluto in cui deve essere montato il volume `tmpfs`.

Tipo: stringa

Obbligatorio: sì

`mountOptions`

L'elenco delle opzioni di montaggio del volume `tmpfs`.

Tipo: array di stringhe

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

Le dimensioni massime (in MiB) del volume tmpfs.

Tipo: numero intero

Obbligatorio: sì

Dipendenze per i container

dependsOn

Tipo: matrice di oggetti [ContainerDependency](#)

Obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un contenitore 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 successive (Linux) o 1.0.0 (Windows).

```
"dependsOn": [
  {
    "containerName": "string",
    "condition": "string"
```



```
}  
]
```

`containerName`

Tipo: stringa

Obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

`condition`

Tipo: stringa

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 è confermata solo all'avvio dell'attività.

Timeout del container

`startTimeout`

Tipo: Integer

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 successive (Linux). Il valore massimo è 120 secondi.

`stopTimeout`

Tipo: Integer

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 successive (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](#)

Obbligatorio: no

Un elenco di parametri kernel associati a un namespace da impostare nel container. Questo parametro è mappato a `Sysctl`s nella creazione container del comando del `docker` e l'opzione `--sysctl` per `docker run`. 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 sono stati impostati `systemControls` per qualsiasi container, si applicano a tutti i container nell'attività. Se sono stati impostati diversi `systemControls` per più container in un'unica attività, il container che viene avviato per ultimo determina quale `systemControls` diventa effettivo.

Se stai impostando un namespace 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` del namespace IPC non sono supportati.
- Per le attività che utilizzano la modalità `IPC task`, i valori di `systemControls` del namespace 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: stringa

Obbligatorio: no

Il parametro kernel associato a un namespace per il quale impostare un parametro value.

Valori del namespace 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 namespace di rete validi: Sysctls che iniziano con "net.*". In Fargate, vengono accettati solo i namespace Sysctls esistenti all'interno del container.

Tutti questi valori sono supportati da Fargate.

value

─Tipo: stringa

Obbligatorio: no

Il valore per il parametro kernel associato a un namespace specificato in namespace.

Interactive

interactive

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è true, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di stdin o tty. Questo parametro è mappato a OpenStdin nella creazione container del comando del docker e l'opzione --interactive per docker run.

Il valore predefinito è `false`.

Pseudoterminale

`pseudoTerminal`

Tipo: Booleano

Obbligatorio: no

Quando il parametro è `true`, è allocato un TTY. Questo parametro è mappato a `Tty` nella creazione container del comando del docker e l'opzione `--tty` per docker run.

Il valore predefinito è `false`.

Nome dell'acceleratore Elastic Inference (obsoleto)

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

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

Obbligatorio: sì

L'acceleratore di inferenza elastica da utilizzare.

Configurazione del proxy

proxyConfiguration

Tipo: oggetto [ProxyConfiguration](#)

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: String

Valori del valore: APPMESH

Obbligatorio: no

Il tipo di proxy. L'unico valore supportato è APPMESH.

containerName

Tipo: stringa

Obbligatorio: sì

Il nome del container che agirà come proxy App Mesh.

properties

Tipo: matrice di oggetti [KeyValuePair](#)

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: stringa

Obbligatorio: no

Nome della coppia chiave-valore.

value

─Tipo: stringa

Obbligatorio: no

Valore della coppia chiave-valore.

Volumi

Quando si registra una definizione di attività, è possibile eventualmente specificare un elenco di volumi da trasferire al daemon Docker su un'istanza di container, che diventerà poi accessibile ad altri container nella stessa istanza.

Di seguito sono elencati i tipi di volumi di dati che è possibile utilizzare:

- Volumi Amazon EBS: forniscono un'archiviazione a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. È possibile collegare 1 volume Amazon EBS a ogni attività Amazon ECS durante l'esecuzione di un'attività isolata oppure 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 [Usare 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. Cresce e si riduce automaticamente man mano che aggiungi e rimuovi file. Le tue applicazioni possono disporre dello spazio di archiviazione di cui hanno bisogno e quando ne hanno bisogno. I volumi Amazon EFS sono supportati per le attività ospitate su istanze di Fargate. Per ulteriori informazioni, consulta [Usare 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.

- Montaggi vincolati: un file o una directory sulla macchina host montato in un container. I volumi host di montaggio vincolato sono supportati durante l'esecuzione delle attività. 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 archiviazione per le attività di Amazon ECS](#).

I seguenti parametri sono consentiti in una definizione del container.

name

─Tipo: stringa

Obbligatorio: no

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (-) e trattini bassi (_). Nel parametro `sourceVolume` dell'oggetto `mountPoints` della definizione del container viene fatto riferimento a questo nome.

host

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: stringa

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

Obbligatorio: no

Specificare 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 `true`, non sarà possibile fornire un'altra configurazione di volume nella definizione dell'attività. Questo parametro deve essere impostato su `true` per configurare un volume Amazon EBS da allegare a un'attività. L'impostazione di `configuredAtLaunch` su `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 dell'attività è riutilizzabile in diversi ambienti di esecuzione. Per ulteriori informazioni consulta [Amazon EBS volumes](#).

dockerVolumeConfiguration

Tipo: oggetto [DockerVolumeConfiguration](#)

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 container Windows supportano solo l'uso del driver `local`. Per utilizzare i montaggi vincolati, specifica invece un `host`.

scope

Tipo: String

Valori validi: `task` | `shared`

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`

Obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo è utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omesso.

`driver`

─Tipo: stringa

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 è utilizzato per il posizionamento dell'attività. Se il driver è stato installato utilizzando la CLI del plugin Docker, utilizzare `docker plugin ls` per richiamare il nome del driver dall'istanza di container. Se il driver è stato installato utilizzando un altro metodo, utilizzare il rilevamento del plugin Docker per richiamare il nome del driver.

`driverOpts`

─Tipo: stringa

Obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da inviare. Questo parametro fa riferimento a `DriverOpts` nella sezione *Crea un volume di Docker*.

`labels`

─Tipo: stringa

Obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`efsVolumeConfiguration`

[Tipo: oggetto di configurazione EFSVolume](#)

Obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

fileSystemId

Tipo: stringa

Obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

rootDirectory

■Tipo: stringa

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 radice deve essere omissso o impostato su / per applicare il percorso impostato sul punto di accesso EFS.

transitEncryption

Tipo: String

Valori validi: ENABLED | DISABLED

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 [Encrypting Data in Transit](#) nella Guida per l'utente di Amazon Elastic File System.

transitEncryptionPort

Tipo: Integer

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, l'attività utilizzerà la strategia di selezione della porta usata 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 [EFSAuthorizationConfig](#)

Obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

■Tipo: stringa

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 `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Amazon EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`iam`

Tipo: String

Valori validi: ENABLED | DISABLED

Obbligatorio: no

Specificare 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 `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](#)

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

Obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

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

Obbligatorio: sì

Le opzioni delle credenziali di autorizzazione.

opzioni:

- Nome della risorsa Amazon (ARN) del segreto [Gestione dei segreti AWS](#).
- ARN di un parametro [AWS Systems Manager](#).

`domain`

Tipo: stringa

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 [Aggiunta di tag alle 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: stringa

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

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 di Amazon ECS attraverso la nuova console](#).

Argomenti

- [Archiviazione temporanea](#)
- [Modalità IPC](#)
- [Modalità PID](#)
- [Iniezione di guasti](#)

Archiviazione temporanea

`ephemeralStorage`

Tipo: oggetto [EphemeralStorage](#)

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 sulla versione della piattaforma 1.4.0 o successive (Linux) oppure 1.0.0 o successive (Windows).

Modalità IPC

`ipcMode`

Non è supportato per le attività eseguite su Fargate.

■Tipo: stringa

Obbligatorio: no

Il namespace 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 del namespace 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 un namespace utilizzando `systemControls` per i container nell'attività, al namespace 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 al namespace 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: String

Valori validi: `host` | `task`

Obbligatorio: no

Il namespace del processo da utilizzare per i container nell'attività. I valori validi sono `host` o `task`. Per i container 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 namespace del processo.

Se non è stato specificato alcun valore, l'impostazione predefinita è un namespace 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 di guasti

`enableFaultInjection`

Tipo: Booleano

Valori validi: `true` | `false`

Obbligatorio: no

Se questo parametro è impostato su `true`, nel payload di un'attività, Amazon ECS e Fargate accettano richieste di iniezione di guasti dai container della stessa. Il valore del parametro è `false` per impostazione predefinita.

Parametri di definizione delle attività di Amazon ECS per Amazon EC2

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 la capacità. 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 Amazon EC2

Family

`family`

Tipo: stringa

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.

Capacity

Quando si registra una definizione di attività, è possibile specificare la capacità che Amazon ECS deve convalidare per tale definizione. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client.

Il parametro seguente è permesso in una definizione di attività.

`requiresCompatibilities`

Tipo: array di stringhe

Obbligatorio: no

Valori validi: EC2

La capacità per la quale è stata convalidata la definizione di attività. Ciò avvia un controllo per garantire che tutti i parametri utilizzati nella definizione dell'attività soddisfino i requisiti di Amazon EC2.

Ruolo del processo

`taskRoleArn`

▪Tipo: stringa

Obbligatorio: no

Quando registri una definizione di attività, puoi fornire un ruolo di attività per un ruolo IAM che consenta ai contenitori inclusi nell'operazione di chiamare per tuo conto i AWS APIs dati specificati nelle politiche associate. 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: String

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

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`, `bridge`, `awsvpc`, 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`.

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 `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 EC2](#).

Se la modalità di rete è `default`, 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: String

Obbligatorio: condizionale

Di default: LINUX

Quando registri una definizione di attività, devi specificare la famiglia del sistema operativo.

I valori validi sono `LINUX`, `WINDOWS_SERVER_2025_FULL`, `WINDOWS_SERVER_2025_CORE`, `WINDOWS_SERVER_2022_CORE`, `WINDOWS_SERVER_2022_FULL`, `WINDOWS_SERVER_2019_FULL`, `WINDOWS_SERVER_2019_CORE`, `WINDOWS_SERVER_2016_FULL`, `WINDOWS_SERVER_2004_CORE` e `WINDOWS_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: String

Obbligatorio: condizionale

Quando si registra una definizione di attività, è possibile specificare l'architettura della CPU. I valori validi sono `X86_64` e `ARM64`. Se non specifichi un valore, Amazon ECS tenta di inserire

attività sull'architettura CPU disponibile in base alla configurazione del provider di capacità. Per garantire che le attività vengano posizionate su un'architettura CPU specifica, specifica un valore per `cpuArchitecture` nella definizione dell'attività.

Tutte le definizioni di attività utilizzate in un servizio devono avere lo stesso valore per questo parametro.

Quando si dispone di attività Linux, è possibile impostare il valore su ARM64. Per ulteriori informazioni, consulta [the section called “Definizioni di 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: String

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: String

Obbligatorio: condizionale

Note

Questo parametro non è supportato per i container Windows.

Il limite rigido di memoria da presentare al processo. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibytes (MiB) o gigabytes (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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

Definizioni del container

Quando registri una definizione di attività, devi specificare un elenco di definizioni del container che vengono trasmesse al daemon Docker in un'istanza di container. 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

- [Name](#)
- [Immagine](#)
- [Memoria](#)
- [Mappature di porte](#)
- [Credenziali del repository privato](#)

Name

name

Tipo: stringa

Obbligatorio: sì

Il nome di un container. Sono consentite fino a 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

Obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene trasmessa direttamente al daemon Docker. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. 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 mappato su Image nel comando creazione container e nel parametro IMAGE del docker di esecuzione del comando.

- 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 contenitore Amazon ECS utilizza il `latest` tag per estrarre l'immagine specificata.
- Gli aggiornamenti successivi a un'immagine del repository non vengono propagati 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`.
- Le immagini in repository ufficiali su Docker Hub utilizzano un singolo nome (ad esempio `ubuntu` o `mongo`).
- Le immagini in altri repository su Docker Hub vengono qualificate con un nome di 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: String

Valori validi: `enabled|disabled`

Obbligatorio: no

Specificare se Amazon ECS risolverà il tag dell'immagine del container fornito nella definizione del container in un digest di immagini. Per impostazione predefinita, questo comportamento è `enabled`. Se si imposta il valore per un container come `disabled`, Amazon ECS non risolverà il tag dell'immagine del container in un digest e utilizzerà l'URI dell'immagine originale specificato

nella definizione del container per l'implementazione. Per ulteriori informazioni sulla risoluzione dell'immagine del container, consultare [Risoluzione dell'immagine del container](#).

Memoria

memory

Tipo: Integer

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 a Memory nella creazione container del comando del docker e l'opzione `--memory` per docker run.

È necessario specificare un valore di memoria a livello di attività o uno a livello di container. 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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

memoryReservation

Tipo: Integer

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 conservare la memoria del container 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 a `MemoryReservation` nella creazione container del comando del docker e l'opzione `--memory-reservation` per `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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

Mappature di porte

portMappings

Tipo: array di oggetti

Obbligatorio: no

Le mappature delle porte espongono le porte di rete del container al mondo esterno. Ciò consente ai client di accedere all'applicazione. Viene anche utilizzato per la comunicazione tra container all'interno 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 container viene mappata automaticamente su una 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 (inclusi `containerPort`, `hostPort` e `protocol`) è associata a `PortBindings` nella sezione creazione container del comando e l'opzione `--publish` per `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: stringa

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: numero intero

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 rete `awsvpc`, utilizzare `containerPort` per specificare le porte esposte.

Supponiamo di utilizzare contenitori in un'attività con i fornitori di EC2 capacità e di specificare una porta container 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: stringa

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 proxy docker nel file di configurazione del daemon Docker quando disponi di un numero elevato di porte.

Per ulteriori informazioni, consultate il [numero #11185](#) su GitHub.

Per informazioni sulla modalità di disattivazione del proxy docker nel file di configurazione del daemon Docker, consulta [Daemon Docker](#) nella Guida per lo sviluppatore di Amazon ECS.

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: stringa

Obbligatorio: no

L'intervallo di numeri di porta sull'host utilizzato con il collegamento di rete. Questo viene assegnato da Docker e consegnato dall'agente Amazon ECS.

`hostPort`

Tipo: Integer

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 la versione Docker. Questa operazione viene definita mappatura dinamica delle porte dell'host.

L'intervallo di porte temporanee predefinite per Docker versione 1.6.0 e successive è elencato nell'istanza in `/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. È possibile utilizzare l'ECS_DYNAMIC_HOST_PORT_RANGE impostazione nella configurazione dell'agente contenitore ECS per specificare un intervallo personalizzato per le porte host assegnate dinamicamente. Ciò può essere utile se le attività non vengono avviate a causa di conflitti di porte con altri processi sull'istanza del contenitore, ad esempio le connessioni in uscita che utilizzano anche porte dell'intervallo di porte temporanee. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

Le porte prenotate di default sono 22 per SSH, le porte Docker 2375 e 2376 e le porte 51678-51680 dell'agente del container di Amazon ECS. 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 attualmente prenotate 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: String

Obbligatorio: no, necessario per configurare 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 di 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 visualizzati entrambi i campi obbligatori per Service Connect e VPC Lattice.

```
"portMappings": [  
  {  
    "name": string,  
    "containerPort": integer  
  }  
]
```

```
]
```

protocol

─Tipo: stringa

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](#)

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: String

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 Console di gestione AWS, 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 del container forniscono funzionalità estese al comando di esecuzione del docker che viene utilizzato per lanciare container nelle istanze di container di Amazon ECS.

Argomenti

- [Politica di riavvio](#)
- [Controllo dell'integrità](#)
- [Ambiente](#)
- [Impostazioni di rete](#)
- [Archiviazione e registrazione](#)
- [Sicurezza](#)
- [Limiti delle risorse](#)
- [Etichette Docker](#)

Politica di riavvio

restartPolicy

La politica di riavvio del container e i parametri di configurazione associati. Quando configuri una politica di riavvio per un container, Amazon ECS può riavviare il container senza dover sostituire l'attività. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

enabled

Tipo: Booleano

Obbligatorio: sì

Specificare se è abilitata una politica di riavvio per il container.

ignoredExitCodes

Tipo: array di numeri interi

Obbligatorio: no

Un elenco di codici di uscita che Amazon ECS ignorerà e non tenterà di riavviare. È possibile specificare fino a 50 codici di uscita dei container. Per impostazione predefinita, Amazon ECS non ignora alcun codice di uscita.

restartAttemptPeriod

Tipo: Integer

Obbligatorio: no

Un periodo di tempo (in secondi) durante il quale il container deve essere eseguito prima che venga tentato un riavvio. Un container può essere riavviato solo una volta ogni `restartAttemptPeriod` secondi. Se un container non è in grado di funzionare per questo periodo di tempo ed esce in anticipo, non verrà riavviato. È possibile specificare un `restartAttemptPeriod` minimo di 60 secondi e un `restartAttemptPeriod` massimo di 1.800 secondi. Per impostazione predefinita, un container deve funzionare per 300 secondi prima di poter essere riavviato.

Controllo dell'integrità

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 dell'integrità 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 Console di gestione AWS, 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 Console di gestione AWS 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 predefinito 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

Obbligatorio: no

Il numero di unità `cpu` che l'agente del container Amazon ECS riserverà per il container. Su Linux, questo parametro è mappato a `CpuShares` nella sezione [Create a container](#).

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.

I container Linux condividono unità CPU non assegnate con altri container nell'istanza di container con lo stesso rapporto della quantità assegnata. 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.

Sulle istanze di container Linux, il daemon Docker nell'istanza di container utilizza il valore CPU per calcolare i relativi rapporti di quote di CPU per i container in esecuzione. Il valore minimo valido per la quota di CPU ammesso dal kernel Linux è 2, mentre quello massimo valido per la quota di CPU ammesso dal kernel Linux è 262.144. Tuttavia, il parametro CPU non è obbligatorio e puoi usare valori di CPU minori di 2 e superiori a 262.144 nelle definizioni del container. Per i

valori di CPU minori di 2 (incluso un valore null) e superiori a 262.144, il comportamento varia in base alla versione dell'agente del container Amazon ECS:

- Versioni dell'agente $\leq 1.1.0$: i valori di CPU null e zero vengono trasmessi a Docker come 0, che Docker successivamente converte in 1.024 quote di CPU. I valori di CPU di 1 vengono trasmessi a Docker come 1 e il kernel Linux li converte in due quote di CPU.
- Versioni dell'agente $\geq 1.2.0$: i valori di CPU null, zero e 1 vengono trasmessi a Docker come due quote di CPU.
- Versioni dell'agente $\geq 1.84.0$: i valori della CPU superiori a 256 vCPU vengono passati a Docker come 256, il che equivale a 262.144 condivisioni CPU.

Sulle istanze di container Windows, la quota di CPU viene applicata come assoluta. I container Windows hanno accesso solo alla quantità di CPU specificata nella definizione delle attività. Un valore di CPU null o zero viene trasmesso a Docker come 0, che Windows interpreta come l'1% di una CPU.

Per ulteriori esempi, consulta [Modalità di gestione delle risorse di CPU e memoria di Amazon ECS](#).

gpu

Tipo: oggetto [ResourceRequirement](#)

Obbligatorio: no

Il numero di GPUs fisiche che l'agente del container Amazon ECS riserva per il container. Il numero di contenitori GPUs riservati a 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](#).

 Note

Questo parametro non è supportato per i container Windows.

Elastic Inference accelerator

Tipo: oggetto [ResourceRequirement](#)

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 Elastic Inference \(obsoleto\)”](#).

 Note

Questo parametro non è supportato per i container Windows.

`essential`

Tipo: Booleano

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 omissso, 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

Obbligatorio: no

Il punto di ingresso che viene trasmesso al container.

```
"entryPoint": ["string", ...]
```

command

Tipo: array di stringhe

Obbligatorio: no

Il comando che viene inviato al container. Questo parametro è mappato a `Cmd` nella creazione container del comando e il parametro `COMMAND` per `docker run`. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

workingDirectory

─Tipo: stringa

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

Obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'opzione `--env-file` al comando esecuzione del docker.

Quando FIPS è abilitato, i nomi dei 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.

Questa funzionalità non è disponibile per i container Windows.

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 [Passare una singola variabile di ambiente a un container Amazon ECS](#).

`value`

Tipo: stringa

Obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

`type`

Tipo: stringa

Obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è `s3`.

`environment`

Tipo: array di oggetti

Obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è mappato a `Env` nella creazione container del comando del docker e l'opzione `--env` al comando per docker run.

 Important

Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

name

Tipo: String

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

value

Tipo: String

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

Obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container Amazon ECS](#).

name

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

Obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del Gestione dei segreti AWS 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

Obbligatorio: no

Quando questo parametro è true, le reti sono disabilitate all'interno del container.

 Note

Questo parametro non è supportato per i container o le attività Windows che utilizzano la modalità di rete awsvpc.

Il valore predefinito è false.

```
"disableNetworking": true|false
```

links

Tipo: array di stringhe

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 costrutto `name:internalName` è analogo a `name:alias` nei collegamenti Docker. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e trattini bassi.

Note

Questo parametro non è supportato per i container o le attività Windows che utilizzano la modalità di rete `awsvpc`.

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: stringa

Obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato a `Hostname` nella creazione container del docker e l'opzione `--hostname` per docker run.

Note

Se utilizzi la modalità di rete `awsvpc`, il parametro `hostname` non è supportato.

```
"hostname": "string"
```

dnsServers

Tipo: array di stringhe

Obbligatorio: no

Un elenco di server DNS presentato al container.

Note

Questo parametro non è supportato per i container o le attività Windows che utilizzano la modalità di rete awsvpc.

```
"dnsServers": ["string", ...]
```

dnsSearchDomains

Tipo: array di stringhe

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 è mappato a `DnsSearch` nella creazione container del comando del docker e l'opzione `--dns-search` per `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

Obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato a `ExtraHosts` nella creazione container del comando del docker e l'opzione `--add-host` per 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: String

Obbligatorio: sì, quando si utilizzano `extraHosts`

Il nome host da utilizzare nella voce `/etc/hosts`.

ipAddress

Tipo: String

Obbligatorio: sì, quando si utilizzano `extraHosts`

L'indirizzo IP da utilizzare nella voce `/etc/hosts`.

Archiviazione e registrazione

readOnlyRootFilesystem

Tipo: Booleano

Obbligatorio: no

Se il parametro è `true`, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro è mappato a `ReadonlyRootfs` nella creazione container del comando del `docker` e l'opzione `--read-only` per `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

Obbligatorio: no

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato ai `Volumes` nella creazione container dell'API Docker e l'opzione `--volume` per `docker run`.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I container Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati tra le unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

sourceVolume

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

containerPath

Tipo: String

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel container in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

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 sul sistema operativo Windows, lasciare il valore predefinito di `false`.

`volumesFrom`

Tipo: array di oggetti

Obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato a `VolumesFrom` nella creazione container del comando del docker e l'opzione `--volumes-from` per docker run.

`sourceContainer`

Tipo: String

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

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](#)

Obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta [Esempio di definizione di attività di Amazon ECS](#).

Questo parametro è mappato a LogConfig nella creazione container del comando del docker e l'opzione `--log-driver` per docker run. Per impostazione predefinita, i container utilizzano lo stesso driver di log utilizzato dal daemon Docker. Tuttavia, il container può utilizzare un driver di log diverso da quello del daemon Docker, specificando un driver di log con questo parametro nella definizione del container. 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 log disponibili per il daemon Docker.
- Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.
- L'agente del container Amazon ECS in esecuzione in un'istanza di container deve registrare i driver di log disponibili in tale istanza con la variabile di ambiente `ECS_AVAILABLE_LOGGING_DRIVERS` prima che i container posizionati sull'istanza possano utilizzare queste opzioni di configurazione di log. 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: String

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 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

Obbligatorio: no

La key/value mappa delle opzioni di configurazione da inviare al driver di registro.

Le opzioni che è possibile specificare dipendono dal driver di log. Alcune delle opzioni che puoi specificare quando utilizzi il `awslogs` router per indirizzare i log ad Amazon CloudWatch includono:

`awslogs-create-group`

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`

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`

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`

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

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 log supportati.

`mode`

Obbligatorio: no

Valori validi: `non-blocking` | `blocking`

Questa opzione definisce la modalità di consegna dei messaggi di log dal container al driver di log specificato utilizzando `logDriver`. La modalità di distribuzione scelta influisce sulla disponibilità dell'applicazione quando il flusso di log dal container viene interrotto.

Se si utilizza la modalità `blocking` e il flusso di log viene interrotto, le chiamate provenienti dal codice del container e utilizzate per la scrittura sui flussi `stdout` e `stderr` 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 la mancata risposta dell'applicazione quando non è possibile inviare i log. Ti consigliamo di utilizzare questa modalità se vuoi garantire la disponibilità del servizio nonostante una certa perdita di log. Per ulteriori informazioni, consultare [Preventing log loss with non-blocking mode in the awslogs container log driver](#).

È possibile impostare un valore predefinito `mode` per tutti i container in un Regione AWS specifico utilizzando le impostazioni dell'account `defaultLogDriverMode`. Se non si specifica l'opzione `mode` in `logConfiguration` o non si configura l'impostazione dell'account, Amazon ECS utilizzerà per impostazione predefinita la modalità non-blocking. Per ulteriori informazioni sulle impostazioni dell'account, consultare [Modalità driver log predefinita](#).

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. La quantità totale di memoria allocata a livello di attività deve essere superiore a quella allocata per tutti i container oltre al driver di log del buffer di memoria.

 Note

Il 25 giugno 2025, Amazon ECS ha cambiato la modalità predefinita del driver di log da `blocking` a `non-blocking` per dare priorità alla disponibilità delle attività rispetto al logging. Per continuare a utilizzare la modalità `blocking` dopo questa modifica, procedere in uno dei seguenti modi:

- Impostare l'opzione `mode` nella definizione del container `logConfiguration` come `blocking`.
- Impostare le impostazioni dell'account `defaultLogDriverMode` su `blocking`.

`max-buffer-size`

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 router di log `splunk`, è necessario specificare `splunk-token` e `splunk-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'`log-driver-buffer-limit` opzione per limitare il numero di righe di registro memorizzate nel buffer prima di essere inviate al contenitore del log router. Può aiutarti a risolvere potenziali problemi di perdita di log perché un throughput elevato potrebbe comportare l'esaurimento della memoria per il buffer all'interno di Docker. Per ulteriori informazioni, consulta [the section called “Configurazione dei log per un throughput elevato”](#).

Le altre opzioni che è possibile specificare quando si utilizza `awsfirelens` per indirizzare i log 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 si esportano i log in flusso di dati Amazon Kinesis, è possibile specificare Regione AWS con `region` 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 si esportano i log in Amazon S3, è possibile specificare il bucket utilizzando l'opzione `bucket`. È possibile 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

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 [Trasferimento di dati sensibili a un container Amazon ECS](#).

name

Tipo: stringa

Obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

valueFrom

Tipo: stringa

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: oggetto [FirelensConfiguration](#)

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

Obbligatorio: no

La key/value mappa delle opzioni da usare durante la configurazione del log router. 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 dell'attività Amazon ECS: indirizzare i log a FireLens](#).

type

Tipo: stringa

Obbligatorio: sì

Il router di log da utilizzare. I valori validi sono fluentd o fluentbit.

Sicurezza

Per ulteriori informazioni sulla sicurezza di container, consultare [Best practice per la sicurezza delle attività e dei container di Amazon ECS](#).

credentialSpecs

Tipo: array di stringhe

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

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 a `Privileged` nella creazione container del comando del docker e l'opzione `--privileged` per 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: stringa

Obbligatorio: no

L'utente da usare all'interno del container. Questo parametro è mappato a `User` nella creazione container del comando del docker e l'opzione `--user` per 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:*value*» | «credentialspec:»
CredentialSpecFilePath

Obbligatorio: no

Un elenco di stringhe per fornire una configurazione personalizzata per più sistemi di sicurezza.

Per le attività Linux, questo parametro può essere utilizzato per fare riferimento a etichette personalizzate per i sistemi di sicurezza su più livelli SELinux e AppArmor .

Questo parametro può essere utilizzato per fare riferimento a un file di specifica delle credenziali che configura un container 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 a SecurityOpt nella creazione container del comando del docker e l'opzione --security-opt per 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

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 a `Ulimits` nella creazione container del comando del docker e l'opzione `--ulimit` per 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: String

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: numero intero

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` di `ulimit`.

softLimit

Tipo: numero intero

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 del `type` di `ulimit`.

Etichette Docker

`dockerLabels`

Tipo: mappatura stringa a stringa

Obbligatorio: no

Una key/value mappa di etichette da aggiungere al contenitore. Questo parametro è mappato a `Labels` nella creazione container del comando del `docker` e l'opzione `--label` per `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 di Amazon ECS attraverso la nuova console](#).

Argomenti

- [Parametri Linux](#)
- [Dipendenze per i container](#)
- [Timeout del container](#)
- [Controlli di sistema](#)
- [Interactive](#)
- [Pseudoterminale](#)

Parametri Linux

linuxParameters

Tipo: oggetto [LinuxParameters](#)

Obbligatorio: no

Linux-opzioni specifiche applicate al contenitore, ad esempio [KernelCapabilities](#).

Note

Questo parametro non è supportato per i container Windows.

```
"linuxParameters": {  
  "capabilities": {  
    "add": ["string", ...],  
    "drop": ["string", ...]  
  }  
}
```

capabilities

Tipo: oggetto [KernelCapabilities](#)

Obbligatorio: no

Le funzionalità di Linux per il container che vengono aggiunte o eliminate dalla configurazione predefinita fornita da Docker. Per ulteriori informazioni su queste funzionalità di Linux, consulta la pagina del manuale Linux [capabilities\(7\)](#).

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

Le funzionalità di Linux per il container da aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato a CapAdd nella creazione container del comando del docker e l'opzione --cap-add per docker run.

add

Tipo: array di stringhe

Valori validi: "SYS_PTRACE"

Campo obbligatorio: no

Le funzionalità di Linux per il container da aggiungere alla configurazione predefinita fornita da Docker. Questo parametro è mappato a CapAdd nella creazione container del comando del docker e l'opzione --cap-add per 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

Le funzionalità di Linux per il container da eliminare dalla configurazione predefinita fornita da Docker. Questo parametro è mappato a CapDrop nella creazione container del comando del docker e l'opzione `--cap-drop` per docker run.

devices

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato a Devices nella creazione container del comando del docker e l'opzione `--device` per docker run.

Tipo: matrice di oggetti [Device](#)

Obbligatorio: no

hostPath

Il percorso per il dispositivo nell'istanza di container dell'host.

Tipo: stringa

Obbligatorio: sì

containerPath

Il percorso nel container in cui esporre il dispositivo dell'host.

■Tipo: stringa

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: array 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'opzione `--init` su 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 convertito nell'opzione `--memory-swap` in `docker run` dove il valore sarebbe la somma della memoria del container più il valore `maxSwap`.

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 omissso, 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'opzione `--shm-size` su `docker run`.

Tipo: numero intero

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'opzione `--memory-swappiness` su `docker run`.

Note

Se utilizzi le attività su Amazon Linux 2023, il parametro `swappiness` non è supportato.

tmpfs

Il percorso del container, le opzioni di montaggio e le dimensioni massime (in MiB) del montaggio `tmpfs`. Questo parametro è mappato all'opzione `--tmpfs` su `docker run`.

Tipo: matrice di oggetti [Tmpfs](#)

Obbligatorio: no

`containerPath`

Percorso assoluto in cui deve essere montato il volume tmpfs.

Tipo: stringa

Obbligatorio: sì

`mountOptions`

L'elenco delle opzioni di montaggio del volume tmpfs.

Tipo: array di stringhe

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`

Le dimensioni massime (in MiB) del volume tmpfs.

Tipo: numero intero

Obbligatorio: sì

Dipendenze per i container

`dependsOn`

Tipo: matrice di oggetti [ContainerDependency](#)

Obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un contenitore 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 del container per abilitare le dipendenze 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](#).

```
"dependsOn": [  
  {  
    "containerName": "string",  
    "condition": "string"  
  }  
]
```

containerName

Tipo: stringa

Obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

condition

Tipo: stringa

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 è confermata solo all'avvio dell'attività.

Timeout del container

`startTimeout`

Tipo: Integer

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

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 parametro `stopTimeout` non è specificato, viene utilizzato il valore impostato per la variabile di configurazione dell'agente del container Amazon ECS `ECS_CONTAINER_STOP_TIMEOUT`. Se non sono impostati né il parametro `stopTimeout` né la variabile di configurazione dell'agente `ECS_CONTAINER_STOP_TIMEOUT`, vengono utilizzati i valori predefiniti di 30 secondi per i container Linux e 30 secondi per quelli Windows. 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](#)

Obbligatorio: no

Un elenco di parametri kernel associati a un namespace da impostare nel container. Questo parametro è mappato a `Sysctl`s nella creazione container del comando del `docker` e l'opzione `--sysctl` per `docker run`. 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`, il namespace della rete `systemControls` non è supportato.

Se stai impostando un namespace 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` del namespace IPC non sono supportati.
- Per le attività che utilizzano la modalità `IPC task`, i valori di `systemControls` del namespace 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: stringa

Obbligatorio: no

Il parametro `kernel` associato a un namespace per il quale impostare un parametro `value`.

Valori del namespace 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 del namespace di rete validi: Sysctls che iniziano con "net.*"

value

─Tipo: stringa

Obbligatorio: no

Il valore per il parametro kernel associato a un namespace specificato in namespace.

Interactive

interactive

Tipo: Booleano

Obbligatorio: no

Quando questo parametro è true, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di stdin o tty. Questo parametro è mappato a OpenStdin nella creazione container del comando del docker e l'opzione --interactive per docker run.

Il valore predefinito è false.

Pseudoterminale

pseudoTerminal

Tipo: Booleano

Obbligatorio: no

Quando il parametro è true, è allocato un TTY. Questo parametro è mappato a Tty nella creazione container del comando del docker e l'opzione --tty per docker run.

Il valore predefinito è false.

Nome dell'acceleratore Elastic Inference (obsoleto)

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

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

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.

È possibile 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 utilizza Amazon ECS per le attività](#).

I seguenti parametri sono consentiti in una definizione del container:

expression

▪Tipo: stringa

Obbligatorio: no

Un'espressione del linguaggio di query del cluster da applicare al vincolo. Per ulteriori informazioni, consulta [Creare espressioni per definire istanze di container per le attività di Amazon ECS](#).

type

Tipo: stringa

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](#)

Obbligatorio: no

I dettagli di configurazione del proxy App Mesh.

Per le attività che utilizzano EC2, 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: String

Valori del valore: APPMESH

Obbligatorio: no

Il tipo di proxy. L'unico valore supportato è APPMESH.

containerName

Tipo: stringa

Obbligatorio: sì

Il nome del container che agirà come proxy App Mesh.

properties

Tipo: matrice di oggetti [KeyValuePair](#)

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: stringa

Obbligatorio: no

Nome della coppia chiave-valore.

value

─Tipo: stringa

Obbligatorio: no

Valore della coppia chiave-valore.

Volumi

Quando si registra una definizione di attività, è possibile eventualmente specificare un elenco di volumi da trasferire al daemon Docker su un'istanza di container, che diventerà poi accessibile ad altri container nella stessa istanza.

Di seguito sono elencati i tipi di volumi di dati che è possibile utilizzare:

- **Volumi Amazon EBS**: forniscono un'archiviazione a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. È possibile collegare 1 volume Amazon EBS a ogni attività Amazon ECS durante l'esecuzione di un'attività isolata oppure durante la creazione o l'aggiornamento di un servizio. I volumi Amazon EBS sono supportati per le attività Linux. Per ulteriori informazioni, consulta [Usare 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. Cresce

e si riduce automaticamente man mano che aggiungi e rimuovi file. Le tue applicazioni possono disporre dello spazio di archiviazione di cui hanno bisogno e quando ne hanno bisogno. I volumi Amazon EFS sono supportati. Per ulteriori informazioni, consulta [Usare 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 container Windows supportano solo l'uso del driver `local`. Per usare i volumi Docker, specifica `dockerVolumeConfiguration` nella definizione di attività.
- Montaggi vincolati: un file o una directory sulla macchina host montato in un container. I volumi host di montaggio vincolato 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 archiviazione per le attività di Amazon ECS](#).

I seguenti parametri sono consentiti in una definizione del container.

`name`

─Tipo: stringa

Obbligatorio: no

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (-) e trattini bassi (_). Nel parametro `sourceVolume` dell'oggetto `mountPoints` della definizione del container viene fatto riferimento a questo nome.

`host`

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: stringa

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

Obbligatorio: no

Specificare 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 `true`, non sarà possibile fornire un'altra configurazione di volume nella definizione dell'attività. Questo parametro deve essere impostato su `true` per configurare un volume Amazon EBS da allegare a un'attività. L'impostazione di `configuredAtLaunch` su `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 dell'attività è riutilizzabile in diversi ambienti di esecuzione. Per ulteriori informazioni consulta [Amazon EBS volumes](#).

`dockerVolumeConfiguration`

Tipo: oggetto [DockerVolumeConfiguration](#)

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 container Windows supportano solo l'uso del driver `local`. Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

Tipo: String

Valori validi: `task` | `shared`

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`

Obbligatorio: no

Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo è utilizzato solo se `scope` è `shared`. Se `scope` è `task`, questo parametro deve essere omesso.

`driver`

─Tipo: stringa

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 è utilizzato per il posizionamento dell'attività. Se il driver è stato installato utilizzando la CLI del plugin Docker, utilizzare `docker plugin ls` per richiamare il nome del driver dall'istanza di container. Se il driver è stato installato utilizzando un altro metodo, utilizzare il rilevamento del plugin Docker per richiamare il nome del driver.

`driverOpts`

─Tipo: stringa

Obbligatorio: no

Una mappa delle opzioni specifiche del driver Docker da inviare. Questo parametro fa riferimento a `DriverOpts` nella sezione *Crea un volume di Docker*.

`labels`

─Tipo: stringa

Obbligatorio: no

Metadati personalizzati da aggiungere al volume Docker.

`efsVolumeConfiguration`

[Tipo: oggetto di configurazione EFSVolume](#)

Obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

─Tipo: stringa

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 radice deve essere omissso o impostato su / per applicare il percorso impostato sul punto di accesso EFS.

`transitEncryption`

Tipo: String

Valori validi: ENABLED | DISABLED

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 [Encrypting Data in Transit](#) nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: Integer

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, l'attività utilizzerà la strategia di selezione della porta usata 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 [EFSAuthorizationConfig](#)

Obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

■Tipo: stringa

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 `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta [Utilizzo dei punti di accesso Amazon EFS](#) nella Guida per l'utente di Amazon Elastic File System.

`iam`

Tipo: String

Valori validi: ENABLED | DISABLED

Obbligatorio: no

Specificare 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 `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](#)

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

Obbligatorio: sì

L' FSx ID del file system Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

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

Obbligatorio: sì

Le opzioni delle credenziali di autorizzazione.

opzioni:

- Nome della risorsa Amazon (ARN) del segreto [Gestione dei segreti AWS](#).
- ARN di un parametro [AWS Systems Manager](#).

`domain`

Tipo: stringa

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 [Aggiunta di tag alle 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: stringa

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

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 di Amazon ECS attraverso la nuova console](#).

Argomenti

- [Modalità IPC](#)
- [Modalità PID](#)

- [Iniezione di guasti](#)

Modalità IPC

ipcMode

─Tipo: stringa

Obbligatorio: no

Il namespace 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 del namespace 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 un namespace utilizzando `systemControls` per i container nell'attività, al namespace 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 al namespace IPC vengono applicati a tutti i container all'interno di un'attività.

Modalità PID

pidMode

Tipo: String

Valori validi: `host` | `task`

Obbligatorio: no

Il namespace 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 namespace del processo.

Se non è stato specificato alcun valore, l'impostazione predefinita è un namespace 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 di guasti

`enableFaultInjection`

Tipo: Booleano

Valori validi: `true` | `false`

Obbligatorio: no

Se questo parametro è impostato su `true`, nel payload di un'attività, Amazon ECS accetta richieste di iniezione di guasti dai container della stessa. Il valore del parametro è `false` per impostazione predefinita.

Modello di definizione di 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 di attività Amazon ECS per Fargate](#).

EC2 modello

```
{
  "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 Fargate

Important

Per Fargate, è necessario includere il parametro `operatingSystemFamily` con uno dei valori seguenti:

- 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
```

Esempio di definizione di attività di Amazon ECS

È possibile copiare gli esempi e i frammenti per iniziare a creare le proprie definizioni di attività.

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 di Amazon ECS attraverso la nuova console](#) e [Parametri di definizione di attività Amazon ECS per Fargate](#).

Per altri esempi di definizione delle attività, consulta [AWS Esempi di definizioni delle attività](#) su GitHub.

Argomenti

- [Server Web](#)
- [Driver di log splunk](#)
- [Driver di log fluentd](#)
- [Driver di log gelf](#)
- [Carichi di lavoro su istanze esterne](#)
- [Immagine Amazon ECR e ruolo IAM della definizione di attività](#)
- [Punto di ingresso con comando](#)
- [Dipendenze per i container](#)
- [Volumi nelle definizioni di attività](#)
- [Definizioni di attività di esempio di Windows](#)

Server Web

Di seguito è riportato un esempio di definizione di attività usando container Linux su 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 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"]
}

```

Driver di log **splunk**

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 [Trasferimento di dati sensibili a un container 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"
}],
```

Driver di log **fluentd**

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 [Trasferimento di dati sensibili a un container 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
  }]
}],
```

Driver di log **gelf**

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 di Amazon ECS attraverso la nuova console](#).

Important

Se i processi richiedono un ruolo IAM di esecuzione del processo, assicurati che sia specificato nella definizione di attività.

Quando distribisci 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 ruolo IAM della definizione di attività

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 di attività

Usare quanto segue per capire come specificare i volumi nelle attività.

- Per informazioni sulla configurazione di un volume Amazon EBS, consultare [Specificare la configurazione del volume Amazon EBS nell'implementazione di Amazon ECS](#).
- Per informazioni sulla configurazione di un volume Amazon EFS, consultare [Configurazione dei file system Amazon EFS per Amazon ECS utilizzando la console](#).
- Per informazioni su come configurare un volume FSx per Windows File Server, vedere [Scopri come configurare FSx i file system Windows File Server per Amazon ECS](#).
- Per informazioni sulla configurazione di un volume docker, consultare [Esempi di volumi Docker per Amazon ECS](#).
- Per informazioni sulla come configurare un montaggio vincolato, consultare [Esempi di montaggio vincolato 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 attività o servizi che fornisce la capacità dell'infrastruttura per le applicazioni containerizzate. Quando crei un cluster, puoi scegliere tra i tre tipi di infrastruttura principali, ciascuno ottimizzato per differenti casi d'uso e requisiti operativi.

Scegliere il tipo di cluster giusto

Amazon ECS fornisce tre tipi di infrastruttura per i tuoi cluster. Scegli il tipo più adatto alle tue esigenze in termini di carico di lavoro, alle tue preferenze operative e ai tuoi obiettivi di ottimizzazione dei costi:

Istanze gestite da Amazon ECS (consigliate)

Ideale per la maggior parte dei carichi di lavoro: gestisce AWS completamente le EC2 istanze Amazon sottostanti, inclusi il provisioning, l'applicazione di patch e la scalabilità. Questa opzione offre un equilibrio ottimale tra prestazioni, convenienza economica e semplicità operativa.

Da usare quando:

- Vuoi gestire la gestione dell'infrastruttura AWS
- Servono risorse di calcolo convenienti con ottimizzazione automatica
- Ci si vuole concentrare sulle proprie applicazioni invece che sull'infrastruttura
- Servono prestazioni prevedibili con dimensionamento flessibile

Fargate

Elaborazione serverless: paghi solo per le risorse usate dalle tue attività senza dover gestire alcuna infrastruttura. Ideale per carichi di lavoro variabili e per iniziare rapidamente.

Da usare quando:

- Si vogliono operazioni completamente serverless
- Si hanno carichi di lavoro imprevedibili o variabili
- Si vuole ridurre al minimo il sovraccarico operativo
- Servono dimensionamento e implementazioni rapide

EC2 Istanze Amazon

Controllo completo: gestisci direttamente le EC2 istanze Amazon sottostanti, inclusa la selezione, la configurazione e la manutenzione delle istanze.

Da usare quando:

- Servono configurazioni o tipi di istanze specifici
- Hai un' EC2 infrastruttura Amazon esistente da sfruttare
- Hai bisogno di software personalizzato AMIs o specializzato
- Serve il massimo controllo sull'infrastruttura sottostante

Note

Amazon ECS Managed Instances è la scelta consigliata per la maggior parte dei nuovi carichi di lavoro in quanto offre la migliore combinazione di prestazioni, ottimizzazione dei costi e semplicità operativa, consentendo AWS al contempo di gestire le attività di gestione dell'infrastruttura.

Componenti del cluster

Oltre alla capacità dell'infrastruttura, un cluster è costituito dalle seguenti risorse:

- La rete (VPC e sottorete) su cui vengono eseguite le attività e i servizi

Quando utilizzi Amazon ECS Managed Instances o Amazon EC2 Instances per la capacità, la sottorete può trovarsi in Availability Zones, Local Zones, Wavelength Zones o. AWS Outposts

- Un namespace facoltativo

Un 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.

Concetti relativi ai cluster

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 in futuro, quindi non dovresti fare affidamento sulla persistenza dei cluster INACTIVE.

- Un cluster può contenere una combinazione di attività ospitate su istanze gestite di Amazon ECS, istanze AWS Fargate Amazon o EC2 istanze esterne. Le attività possono essere eseguite su Amazon ECS Managed Instances, Fargate o sull' EC2 infrastruttura come tipo di lancio o strategia di capacity provider. Se utilizzi fornitori di EC2 capacità, Amazon ECS non monitora e ridimensiona la capacità dei gruppi Amazon EC2 Auto Scaling.
- Un cluster può contenere una combinazione di provider di capacità di Istanze gestite da Amazon ECS, provider di capacità del gruppo Auto Scaling e provider di capacità Fargate. Una strategia di fornitura di capacità può contenere solo provider di capacità di Istanze gestite da Amazon ECS, provider di capacità di gruppo Auto Scaling o provider di capacità Fargate.
- Puoi utilizzare diversi tipi di istanze per i provider di capacità del gruppo Amazon ECS Managed Instances e/o Auto Scaling. EC2 Un'istanza può essere registrata in un solo cluster alla volta.
- È possibile limitare l'accesso ai cluster creando policy IAM personalizzate. Per informazioni, consulta la sezione [Esempi di cluster Amazon ECS](#) in [Esempi di policy basate su identità per Amazon Elastic Container Service](#).

- È possibile usare il dimensionamento automatico del servizio per scalare le attività di Fargate. Per ulteriori informazioni, consulta [Scalabilità automatica del servizio Amazon ECS](#).
- È possibile configurare un namespace Service Connect predefinito per un cluster. Dopo aver impostato un namespace predefinito di Service Connect, tutti i nuovi servizi con Service Connect attivato creati nel cluster saranno aggiunti come servizi client nel namespace. e non sono necessarie ulteriori configurazioni. Per ulteriori informazioni, consulta [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

fornitori di capacità

I provider di capacità Amazon ECS gestiscono il dimensionamento dell'infrastruttura per le attività nei cluster. Ogni cluster ha uno o più fornitori di capacità e una strategia di fornitori di capacità facoltativa. È possibile assegnare una strategia predefinita per il fornitore di capacità al cluster. La strategia del fornitore di capacità determina il modo in cui le attività vengono distribuite tra i fornitori di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del fornitore di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster. La strategia predefinita del provider di capacità del cluster si applica solo quando non viene specificato un tipo di avvio o una strategia del provider di capacità per l'attività o il servizio. Se viene fornito uno di questi parametri, la strategia predefinita non viene usata.

Amazon ECS offre tre tipi di provider di capacità per i cluster:

Provider di capacità di Istanze gestite da Amazon ECS

AWS gestisce completamente le EC2 istanze Amazon sottostanti, inclusi il provisioning, l'applicazione di patch, la scalabilità e la gestione del ciclo di vita. Questo offre un equilibrio ottimale tra prestazioni, convenienza economica e semplicità operativa. I provider di capacità di Amazon ECS Managed Instances ottimizzano automaticamente la selezione e il dimensionamento delle istanze in base ai requisiti del carico di lavoro.

Con Istanze gestite da Amazon ECS, puoi ottenere i seguenti vantaggi:

- Provisioning e dimensionamento automatici delle istanze
- Gestione dell'applicazione di patch e Aggiornamenti di sicurezza
- Ottimizzazione dei costi con la selezione intelligente delle istanze
- Riduzione del sovraccarico operativo

Provider di capacità di Fargate

Elaborazione serverless con cui paghi solo per le risorse usate dalle tue attività senza dover gestire alcuna infrastruttura. Basta associare i provider di capacità predefiniti (Fargate e Fargate Spot) al cluster.

Provider di capacità dei gruppi Auto Scaling

Quando utilizzi EC2 le istanze Amazon per la tua capacità, utilizzi il gruppo Auto Scaling per gestire le istanze Amazon EC2. Auto Scaling aiuta a garantire il numero corretto di EC2 istanze Amazon disponibili per gestire il carico dell'applicazione. Hai il pieno controllo dell'infrastruttura sottostante.

Un cluster può contenere una combinazione di attività ospitate su istanze gestite di Amazon ECS, istanze AWS Fargate Amazon o EC2 istanze esterne. Le attività possono essere eseguite su Amazon ECS Managed Instances, 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.

Un cluster può contenere una combinazione di provider di capacità di Istanze gestite da Amazon ECS, provider di capacità del gruppo Auto Scaling e provider di capacità Fargate. Una strategia di fornitura di capacità può contenere solo provider di capacità di Istanze gestite da Amazon ECS, provider di capacità di gruppo Auto Scaling o provider di capacità Fargate.

Cluster per Istanze gestite da Amazon ECS

I provider di capacità Amazon ECS gestiscono il dimensionamento dell'infrastruttura per le attività nei cluster. Una volta creato il cluster, vengono creati uno o più provider di capacità e una strategia dei provider di capacità opzionale per il cluster. La strategia del fornitore di capacità determina il modo in cui le attività vengono distribuite tra i fornitori di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del fornitore di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster.

Istanze gestite da Amazon ECS dispone dei seguenti provider di capacità.

Tipo	Criteri usati per scegliere le istanze
Predefinita	Le istanze più convenienti che soddisfano i seguenti requisiti relativi alla definizione delle attività e ai parametri di servizio: • Definizione di attività • <code>operatingSystemFamily</code> • <code>cpuArchitecture</code> • <code>cpu</code> • <code>memory</code> • Definizioni di servizi • <code>placementConstraints</code> • <code>placementStrategy</code>
Personalizzato	Le istanze che soddisfano i requisiti di attributo e tipo specificati durante la creazione del cluster. Per ulteriori informazioni relative agli attributi, consulta Attributi delle istanze di container di Amazon ECS. Per informazioni sui tipi di istanza, consulta le specifiche del tipo di EC2 istanza Amazon in Amazon EC2 Instance Types.

Amazon ECS avvia le istanze e le associa al provider di capacità di Istanze gestite da Amazon ECS. Per il provider di capacità personalizzato, Amazon ECS crea anche il provider di capacità.

Creazione di un cluster per Istanze gestite da Amazon ECS

Creazione di un cluster per definire l'infrastruttura in cui eseguire le attività e i servizi.

Quando crei un cluster per Istanze gestite da Amazon ECS, ottieni l'accesso al provider di capacità `FARGATE_MANAGED_INSTANCE` per impostazione predefinita. Il provider di capacità seleziona automaticamente i tipi di istanze più convenienti per i tuoi carichi di lavoro. Se hai bisogno di attributi o tipi di istanze specifici, puoi anche creare provider di capacità personalizzati.

Per semplificare il più possibile il processo di creazione del cluster, la console dispone di selezioni predefinite per molte scelte.

- Crea uno spazio dei nomi predefinito con AWS Cloud Map lo stesso nome del cluster. Un namespace consente ai servizi creati nel cluster di connettersi agli altri servizi nel namespace senza configurazioni aggiuntive.

Per ulteriori informazioni, consulta [Interconnessione dei servizi Amazon ECS](#).

È possibile modificare le seguenti opzioni:

- Modifica il namespace predefinito associato al cluster.

Un namespace consente ai servizi creati nel cluster di connettersi agli altri servizi nel namespace senza configurazioni aggiuntive. Il namespace predefinito ha lo stesso nome del cluster. Per ulteriori informazioni, consulta [Interconnessione dei servizi Amazon ECS](#).

- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- Aggiungi tag per facilitare l'identificazione del cluster.

Prerequisiti

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”](#).

L'utente che crea il cluster deve possedere un'autorizzazione aggiuntiva:`iam:CreateServiceLinkedRole`.

Per impostazione predefinita, Amazon ECS sceglie i tipi di istanza in base ai requisiti specificati nella definizione dell'attività. Questo è il provider di capacità di default. Se ti servono attributi o tipi di istanza specifici, prendi nota di tutti i requisiti. Dovrai usare un provider di capacità personalizzato e quindi specificare i requisiti dell'istanza.

Capisci come scegliere le tue istanze. Per ulteriori informazioni, consulta [Best practice per la selezione delle istanze per le Istanze gestite da Amazon ECS](#).

Disponi dei ruoli IAM richiesti per Istanze gestite da Amazon ECS. Questo include:

- Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

Procedura della console

Creazione di un nuovo cluster (console Amazon ECS)

1. [Apri la console alla v2. 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 il namespace utilizzato per Service Connect sia diverso dal nome del cluster, in Namespace, inserisci un nome univoco.
6. Per Provider di capacità personalizzato, procedi come segue:
 - Per Seleziona un metodo per ottenere EC2 capacità, scegli Amazon ECS Managed Instances.
 - Per il profilo dell'istanza, seleziona il ruolo del profilo dell'istanza.
 - Per il ruolo dell'infrastruttura, seleziona il ruolo dell'infrastruttura.
 - Per usare un provider di capacità personalizzato, per Selezione delle istanze, scegli Usa personalizzato. Quindi inserisci il valore dell'attributo per ogni attributo.
7. (Facoltativo) Usa Container Insights, espandi Monitoraggio, quindi seleziona una delle seguenti opzioni:
 - Per usare la soluzione consigliata di Container Insights con osservabilità migliorata, seleziona Container Insights con osservabilità migliorata.

- Per utilizzare Container Insights scegli Container Insights.
8. (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.
 9. (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.
10. Scegli Create (Crea).

AWS CLI procedura

Puoi creare un cluster per Istanze gestite da Amazon ECS utilizzando la AWS CLI. Usa la versione più recente dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the 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 [the section called "Utilizzo degli endpoint dual-stack"](#).

Come creare un nuovo cluster (AWS CLI)

1. Crea il tuo cluster con un nome univoco usando il comando seguente:

```
aws ecs create-cluster --cluster-name managed-instances-cluster
```

Output:

```
{
  "cluster": {
    "status": "ACTIVE",
    "defaultCapacityProviderStrategy": [],
    "statistics": [],
    "capacityProviders": [],
```

```
"tags": [],
"clusterName": "managed-instances-cluster",
"settings": [
  {
    "name": "containerInsights",
    "value": "disabled"
  }
],
"registeredContainerInstancesCount": 0,
"pendingTasksCount": 0,
"runningTasksCount": 0,
"activeServicesCount": 0,
"clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/managed-instances-cluster"
}
```

2. (Facoltativo) Per abilitare Container Insights con una osservabilità migliorata per il tuo cluster, usa il seguente comando:

```
aws ecs put-account-setting --name containerInsights --value enhanced
```

3. (Facoltativo) Per aggiungere tag al cluster, usa il seguente comando:

```
aws ecs tag-resource --resource-arn
arn:aws:ecs:region:aws_account_id:cluster/managed-instances-cluster --tags
key=Environment,value=Production
```

Fasi successive

Crea una definizione dell'attività per Istanze gestite da Amazon ECS. Per ulteriori informazioni, consulta [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#).

Esegui le 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'implementazione di aggiornamenti continui di Amazon ECS](#)

Aggiornare un cluster per usare Istanze gestite da Amazon ECS

Puoi aggiornare un cluster esistente per usare Istanze gestite da Amazon ECS.

Quando aggiungi Istanze gestite da Amazon ECS al tuo cluster, per impostazione predefinita ottieni l'accesso al provider di capacità `FARGATE_MANAGED_INSTANCE`. Questo provider di capacità seleziona automaticamente i tipi di istanze per uso generico più ottimizzati in termini di costi per i tuoi carichi di lavoro. Se hai bisogno di attributi o tipi di istanze specifici, puoi anche creare provider di capacità personalizzati.

Prerequisiti

Per impostazione predefinita, Amazon ECS sceglie i tipi di istanza in base ai requisiti specificati nella definizione dell'attività. Questo è il provider di capacità di default. Se ti servono attributi o tipi di istanza specifici, prendi nota di tutti i requisiti. Dovrai usare un provider di capacità personalizzato e quindi specificare i requisiti dell'istanza.

Disponi dei ruoli IAM richiesti per Istanze gestite da Amazon ECS. Questo include:

- Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

Considerazioni sugli aggiornamenti

Durante l'aggiornamento di un cluster per Istanze gestite da Amazon ECS, tieni presente quanto segue:

- Attività in esecuzione: l'aggiornamento delle impostazioni del cluster non influisce sulle attività attualmente in esecuzione. Le modifiche saranno applicate alle nuove attività avviate dopo l'aggiornamento.
- Modifiche al provider di capacità: se modifichi le impostazioni del provider di capacità, le istanze gestite esistenti continueranno a funzionare, ma le nuove istanze useranno la configurazione aggiornata.

- Monitoraggio delle modifiche: l'attivazione o la disattivazione di Container Insights influirà sulla raccolta delle metriche per l'intero cluster.

Procedura della console

Per aggiornare un cluster (console Amazon ECS)

1. [Apri la console alla v2. 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 Cluster, seleziona il cluster che desideri aggiornare.
5. Seleziona Aggiorna cluster.
6. (Facoltativo) Per modificare le impostazioni del provider di capacità, in Provider di capacità personalizzato, aggiorna quanto segue in base alle esigenze:
 - Per Profilo dell'istanza, scegli un ruolo diverso nel profilo dell'istanza, se necessario.
 - Per ruolo dell'infrastruttura, scegli un ruolo dell'infrastruttura diverso, se necessario.
 - Per usare un provider di capacità personalizzato, per Selezione delle istanze, aggiorna le impostazioni del Valore dell'attributo.
7. Scegliere Aggiorna.

AWS CLI procedura

Puoi aggiornare un cluster per Istanze gestite da Amazon ECS utilizzando la AWS CLI. Usa la versione più recente dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the 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 [the section called “Utilizzo degli endpoint dual-stack”](#).

Come aggiornare un cluster (AWS CLI)

1. Creazione di un provider di capacità per . Esegui il comando seguente:

Sostituiscili con i tuoi valori. *user-input*

```
aws ecs create-capacity-provider \
  --name my-managed-instances-provider \
  --managed-instances-provider \
  --instance-profile arn:aws:iam::123456789012:instance-profile/ecsInstanceProfile \
  --infrastructure-role-arn arn:aws:iam::123456789012:role/ecsInfrastructureRole \
  --instance-requirements '{
    "vCpuCount": {"min": 2, "max": 8},
    "memoryMiB": {"min": 4096, "max": 16384}
  }
```

2. Aggiungi il provider di capacità al cluster. Usa il seguente comando:

user-input Sostituiscili con i tuoi valori.

```
aws ecs put-cluster-capacity-providers --cluster managed-instances-cluster --
capacity-providers my-managed-instances-provider --default-capacity-provider-
strategy capacityProvider=my-managed-instances-provider,weight=1
```

Provider di capacità di Istanze gestite da Amazon ECS

I provider di capacità di Amazon ECS Managed Instances forniscono un modello di calcolo containerizzato che consente di accedere all'intera gamma di funzionalità AWS e EC2 offerte Amazon, AWS gestendo al contempo le responsabilità operative e di sicurezza. AWS gestisce le patch del software e del sistema operativo, il ridimensionamento delle istanze e la manutenzione, offrendoti i vantaggi operativi di Fargate pur mantenendo l'accesso a tutte le AWS funzionalità e le integrazioni.

Amazon ECS crea modelli di avvio per le Istanze gestite da Amazon ECS. Questo definisce il modo in cui Amazon ECS avvia le istanze gestite da Amazon ECS, compreso il profilo dell'istanza per le attività, la configurazione di rete e storage, le opzioni di capacità e i requisiti dell'istanza per una selezione flessibile del tipo di istanza.

Quando usare provider di capacità personalizzati

Prendi in considerazione i provider di capacità personalizzati quando i carichi di lavoro richiedono:

- Requisiti di elaborazione specifici: applicazioni che richiedono elaborazione accelerata, set di istruzioni CPU specifici, prestazioni di rete elevate o configurazioni di memoria di grandi dimensioni che non sono disponibili con le opzioni Fargate standard.
- Carichi di lavoro GPU: inferenza di machine learning, rendering di immagini in tempo reale, codifica video o altre applicazioni accelerate da GPU che richiedono l'accesso a NVIDIA o AMD. GPUs
- Prenotazioni di capacità: carichi di lavoro mission critical che richiedono una disponibilità prevedibile della capacità.
- Osservabilità avanzata: strumenti di sicurezza e monitoraggio che richiedono un accesso privilegiato al sistema operativo sottostante, come soluzioni di monitoraggio basate su EBPF o strumenti di analisi della rete.
- Ottimizzazione dei costi: carichi di lavoro che possono trarre vantaggio dal posizionamento in più attività, dai componenti dell'infrastruttura condivisi o che devono massimizzare l'utilizzo di tipi di istanze più grandi.

Opzioni di monitoraggio

Le istanze gestite da Amazon ECS offrono funzionalità di monitoraggio complete che consentono di tenere traccia delle prestazioni, dello stato di integrità e dell'utilizzo delle risorse dei carichi di lavoro containerizzati. Puoi scegliere tra diversi livelli di monitoraggio in base alle tue esigenze operative.

- Monitoraggio di base: fornisce metriche essenziali a intervalli di 5 minuti per la maggior parte delle metriche e a intervalli di 1 minuto per i controlli dello stato. Questa funzionalità è abilitata per impostazione predefinita e non comporta costi aggiuntivi.
- Monitoraggio dettagliato: offre una maggiore visibilità con tutte le metriche disponibili a intervalli di 1 minuto, permettendo un rilevamento e una risposta più rapidi ai problemi operativi. Per ulteriori informazioni, consulta [Monitoraggio dettagliato delle istanze gestite da Amazon ECS](#).

Entrambe le opzioni di monitoraggio si integrano perfettamente CloudWatch per fornire dashboard, allarmi e risposte automatiche per aiutare a mantenere prestazioni e disponibilità ottimali delle applicazioni.

Considerazioni sul provider di capacità

Un cluster può contenere una combinazione di provider di capacità di Istanze gestite da Amazon ECS, provider di capacità del gruppo Auto Scaling e provider di capacità Fargate. Una strategia di fornitura di capacità può contenere solo provider di capacità di Istanze gestite da Amazon ECS, provider di capacità di gruppo Auto Scaling o provider di capacità Fargate.

Propagazione di tag

I provider di capacità per le istanze gestite da Amazon ECS supportano la propagazione dei tag. Con la propagazione dei tag, tutte le risorse gestite dal provider di capacità (istanza gestita, istanza di container Amazon ECS, modello di avvio, volumi, interfacce di rete elastiche) sono contrassegnate con gli stessi tag specificati a livello del provider di capacità. Puoi specificare i tag durante la creazione del provider di capacità e abilitare la propagazione dei tag specificando il parametro `propagateTags` come `CAPACITY_PROVIDER`.

Per ulteriori informazioni sull'aggiunta di tag nelle Istanze gestite da Amazon ECS, consulta [Tag per le istanze gestite da Amazon ECS](#).

Best practice per l'aggiornamento dei provider di capacità per le istanze gestite da Amazon ECS

Per garantire il massimo livello di sicurezza e supporto al rollback, consigliamo di trattare i provider di capacità come risorse non modificabili. Per aggiornare la configurazione di un provider di capacità, segui il seguente flusso di lavoro consigliato:

1. Crea un nuovo provider di capacità con la configurazione aggiornata invece di modificare quello esistente.
2. Aggiorna ogni servizio per usare il nuovo provider di capacità e consenti il completamento delle implementazioni.
3. Elimina il precedente provider di capacità dopo aver verificato che la nuova configurazione funzioni come previsto.

Questo approccio offre diversi vantaggi:

- Rollout controllato: puoi aggiornare i servizi uno alla volta e monitorarne l'impatto.
- Rollback semplificato: in caso di problemi, è possibile ripristinare rapidamente i servizi in modo che usino il provider di capacità precedente.

- Raggio d'azione ridotto: i problemi con la nuova configurazione non influiscono immediatamente su tutti i carichi di lavoro.

Note

Se lo utilizzi CloudFormation, valuta la possibilità di mantenere il vecchio provider di capacità fino a una distribuzione successiva per preservare la possibilità di ripristinare le modifiche apportate allo stack.

Sebbene sia possibile aggiornare i provider di capacità in locale, questo approccio crea un raggio d'azione incontrollato più ampio. Gli aggiornamenti in locale applicano le nuove impostazioni a tutta la nuova capacità fornita in futuro, ma non attivano l'implementazione dei servizi. Questo significa che potresti scoprire i problemi di configurazione solo molto più tardi, quando i tuoi servizi dovranno essere scalati.

Creazione di un provider di capacità per le istanze gestite da Amazon ECS

Istanze gestite da Amazon ECS usa provider di capacità per gestire la capacità di calcolo dei carichi di lavoro. Per impostazione predefinita, Amazon ECS fornisce un provider di capacità predefinito che seleziona automaticamente i tipi di istanze per uso generico più ottimizzati in termini di costi. Tuttavia, puoi creare provider di capacità personalizzati per specificare attributi delle istanze come tipi di istanze, produttori di CPU, tipi di acceleratori e altri requisiti.

I provider di capacità personalizzati usano la selezione del tipo di istanza basata sugli attributi, che consente di esprimere i requisiti dell'istanza come un set di attributi. Questi requisiti vengono tradotti automaticamente in tutti i tipi di EC2 istanze Amazon corrispondenti, semplificando la creazione e la manutenzione delle configurazioni dei tipi di istanza. Per ulteriori informazioni sui requisiti delle istanze e sulla selezione basata sugli attributi, consulta la [documentazione sulla selezione del tipo di istanza basata sugli attributi di Amazon EC2 Fleet nella Amazon User Guide](#). EC2

Prerequisiti

Prima di iniziare, assicurati di aver completato quanto segue:

- Determina il tipo di monitoraggio da usare. Per ulteriori informazioni, consulta [the section called "Monitoraggio dettagliato delle istanze gestite da Amazon ECS"](#).
- Disponi di un cluster esistente o pianifica di crearne uno. Per ulteriori informazioni, consulta [Creazione di un cluster per Istanze gestite da Amazon ECS](#).

- Disponi dei ruoli IAM richiesti per Istanze gestite da Amazon ECS. Questo include:
 - Ruolo dell'infrastruttura: consente ad Amazon ECS di effettuare chiamate ai AWS servizi per tuo conto per gestire l'infrastruttura Amazon ECS Managed Instances.

Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS](#).

- Profilo dell'istanza: fornisce le autorizzazioni per l'agente del container di Amazon ECS e il daemon Docker in esecuzione su istanze gestite.

Per ulteriori informazioni, consulta [Profilo delle istanze gestite da Amazon ECS](#).

Capisci come scegliere le tue istanze. Per ulteriori informazioni, consulta [Best practice per la selezione delle istanze per le Istanze gestite da Amazon ECS](#).

Procedura della console

Per creare un provider di capacità per le istanze gestite da Amazon ECS (console Amazon ECS)

1. [Apri la console alla v2. 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 Cluster, seleziona il nome del tuo cluster.
5. Nella pagina Cluster, scegli la scheda Infrastruttura.
6. Nella sezione Provider di capacità, seleziona Crea provider di capacità.
7. In Configurazione del provider di capacità, configura i seguenti elementi:
 - In Nome del provider di capacità, inserisci un nome univoco per il tuo provider di capacità.
 - Per il Tipo di provider di capacità, seleziona Istanze gestite da Amazon ECS.
8. In Configurazione istanze, configura gli elementi seguenti:
 - Per Profilo dell'istanza, seleziona il ruolo del profilo dell'istanza creato per le istanze gestite da Amazon ECS.
 - In Ruolo dell'infrastruttura, scegli il ruolo dell'infrastruttura creato per le istanze gestite da Amazon ECS.
9. In Requisiti dell'istanza, specifica gli attributi per le istanze. Puoi configurare qualsiasi combinazione dei seguenti elementi:
 - Numero vCPU: specifica il numero di v CPUs (ad esempio, 4 o 8-16 per un intervallo).

- Memoria (MiB): specifica la quantità di memoria in MiB (ad esempio, 8192 o 16384-32768 per un intervallo).
- Tipi di istanza: specifica tipi di istanza specifici (ad esempio, `m5.large`, `m5.xlarge`, `c5.large`).
- Produttori di CPU: scegli tra `intel`, `amd` o `amazon-web-services`.
- Tipi di acceleratori: specifica i tipi di acceleratori come `gpu`, `fpga` o `inference`.
- Numero di acceleratori: specifica il numero di acceleratori (ad esempio, 1 o 2-4 per un intervallo).

10. In Configurazione avanzata, scegli una delle seguenti opzioni di monitoraggio:

- Per impostare le metriche di controllo dello stato dell' CloudWatch invio, scegli Basic.
- Per CloudWatch inviare tutte le metriche delle metriche, scegli Dettagliato.

11. (Facoltativo) Per identificare il tuo provider di capacità, espandi la sezione Tag, quindi configura i tag.

Per abilitare la propagazione dei tag dal provider di capacità alle risorse gestite, come le istanze avviate dal provider di capacità, seleziona Provider di capacità in Propaga tag da.

[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.

12. Scegli Create (Crea).

AWS CLI procedura

È possibile creare un provider di capacità per le istanze gestite da Amazon ECS usando AWS CLI. Usa la versione più recente dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the AWS CLI](#).

Per creare un provider di capacità per le istanze gestite da Amazon ECS (AWS CLI)

1. Esegui il comando seguente:


```
aws ecs create-capacity-provider --cli-input-json file:///capacity-provider-definition.json
```

Il seguente file `capacity-provider-definition.json` può essere utilizzato per specificare i requisiti di base dell'istanza, la dimensione di archiviazione dell'istanza e abilitare la propagazione dei tag:

```
{
  "name": "my-managed-instances-provider",
  "cluster": "my-cluster",
  "tags": [
    {
      "key": "version",
      "value": "test"
    }
  ],
  "managedInstancesProvider": {
    "infrastructureRoleArn": "arn:aws:iam::123456789012:role/ecsInfrastructureRole",
    "instanceLaunchTemplate": {
      "ec2InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
      "instanceRequirements": {
        "vCpuCount": {
          "min": 4,
          "max": 8
        },
        "memoryMiB": {
          "min": 8192,
          "max": 16384
        }
      },
      "networkConfiguration": {
        "subnets": [
          "subnet-abcdef01234567",
          "subnet-bcdefa98765432"
        ],
        "securityGroups": [
          "sg-0123456789abcdef"
        ]
      }
    }
  },
}
```

```
        "storageConfiguration": {
            "storageSizeGiB": 100
        },
        "monitoring": "basic"
    },
    "propagateTags": "CAPACITY_PROVIDER"
}
```

2. Verifica che il tuo provider di capacità sia stato creato in modo corretto:

```
aws ecs describe-capacity-providers \
    --capacity-providers my-managed-instances-provider
```

Fasi successive

Dopo aver creato il tuo provider di capacità, puoi usarlo durante la creazione di servizi o l'esecuzione di attività:

- Per usare il provider di capacità con un servizio, consulta [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#).
- Per usare il provider di capacità con attività autonome, consulta [Esecuzione di un'applicazione come attività Amazon ECS](#).

Aggiornamento del monitoraggio delle istanze gestite da Amazon ECS

Puoi modificare l'opzione di monitoraggio per il provider di capacità delle istanze gestite da Amazon ECS per passare dal monitoraggio di base a quello dettagliato. Ciò permette di regolare il livello dei dati di monitoraggio raccolti senza ricreare il provider di capacità.

Per ulteriori informazioni sulle opzioni di monitoraggio, consulta [Monitoring Amazon ECS Managed Instances](#).

Procedura della console

Per aggiornare il monitoraggio per Istanze gestite da Amazon ECS (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 Cluster, seleziona il nome del tuo cluster.
5. Nella pagina Cluster, scegli la scheda Infrastruttura.
6. In Configurazione avanzata, scegli una delle seguenti opzioni di monitoraggio:
 - Per impostare le metriche di controllo dello stato dell' CloudWatch invio, scegli Basic.
 - Per CloudWatch inviare tutte le metriche delle metriche, scegli Dettagliato.
7. Scegliere Aggiorna.

Per aggiornare i tag associati a un provider di capacità Istanze gestite da Amazon ECS esistente, procedi come segue:

1. Nel pannello di navigazione scegliere Cluster.
2. Nella pagina Cluster, seleziona Infrastruttura.
3. Nella pagina Infrastruttura, seleziona il provider di capacità che hai creato.
4. Nella pagina del provider di capacità, seleziona Tag.
5. In Tag, scegli Gestisci tag.
6. Per aggiungere un tag, specifica la chiave e il valore del tag da aggiungere e scegli Salva. Per aggiungere più tag contemporaneamente, scegli Aggiungi nuovo tag per ogni tag da aggiungere. È possibile aggiungere un massimo di 50 tag.

Per rimuovere un tag, scegli Remove (Rimuovi).

Note

Se la propagazione dei tag è abilitata, i tag aggiunti o rimossi dopo la creazione del provider di capacità non si applicano alle risorse create in precedenza dal provider di capacità.

AWS CLI procedura

È possibile aggiornare un provider di capacità per le istanze gestite da Amazon ECS usando AWS CLI. Usa la versione più recente dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the AWS CLI](#).

Per aggiornare il monitoraggio per Istanze gestite da Amazon ECS (AWS CLI)

1. Per abilitare il monitoraggio dettagliato, usa il seguente comando:

```
aws ecs update-capacity-provider \  
  --name my-managed-instances-provider \  
  --managed-instances-provider '{  
    "instanceLaunchTemplateUpdate": {  
      "monitoring": "DETAILED"  
    }  
  }'
```

2. Per abilitare il monitoraggio di base, usa il seguente comando:

```
aws ecs update-capacity-provider \  
  --name my-managed-instances-provider \  
  --managed-instances-provider '{  
    "instanceLaunchTemplateUpdate": {  
      "monitoring": "BASIC"  
    }  
  }'
```

Eliminazione di un provider di capacità di Istanze gestite da Amazon ECS

Se hai finito di utilizzare un provider di capacità Amazon ECS Istanze gestite da Amazon ECS, puoi eliminarlo. Dopo l'eliminazione del gruppo, il provider di capacità di Istanze gestite da Amazon ECS 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 il provider di capacità Amazon ECS Istanze gestite da Amazon ECS, è necessario rimuoverlo 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. L'opzione `Forza nuova implementazione` può essere utilizzata per garantire che tutte le attività che utilizzano la capacità delle istanze gestite da Amazon ECS fornita dal provider di capacità vengano autorizzate a utilizzare la capacità dei provider di capacità rimanenti.

Per eliminare un provider di capacità per il cluster (console Amazon ECS)

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.

2. Nel pannello di navigazione scegliere Cluster.
3. Nella pagina Cluster, scegliere il cluster.
4. Nella **name** pagina Cluster:, scegli Infrastruttura, il provider di capacità Amazon ECS Managed Instances, quindi scegli Elimina.
5. Nella casella di conferma, inserisci delete **Amazon ECS Managed Instances capacity provider name**
6. Scegli Elimina.

Interrompi in modo sicuro i carichi di lavoro Amazon ECS in esecuzione su Istanze gestite da Amazon ECS

La protezione scalabile di Amazon ECS consente di proteggere le attività dall'interruzione da parte di eventi di riduzione orizzontale derivanti da dimensionamento del servizio o implementazioni varie.

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. 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 i server di gioco come attività 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`. Quando si imposta `ProtectionEnabled` su `true`, le attività vengono protette per 2 ore per impostazione predefinita. Poi è possibile personalizzare il periodo di protezione utilizzando l'attributo `ExpiresInMinutes`. 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. Al termine dell'attività, è possibile annullare l'attributo `ProtectionEnabled` utilizzando lo stesso endpoint, rendendo l'attività idonea all'interruzione durante i successivi eventi di scalabilità.

Per ulteriori informazioni sull'endpoint dell'agente del container Amazon ECS, consultare [Endpoint di protezione scalabile di Amazon ECS](#).

- API Amazon ECS

È possibile utilizzare l'API Amazon ECS per impostare e recuperare la protezione scalabile delle attività se l'applicazione dispone di un componente che tiene traccia dello stato delle attività attive. Utilizzare `UpdateTaskProtection` per contrassegnare una o più attività come protette. Utilizzare `GetTaskProtection` per recuperare 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:

- La protezione scalabile è supportata solo con le attività implementate da un servizio.
- La protezione scalabile è supportata con attività implementate da un servizio in esecuzione su istanze gestite da Amazon ECS.
- 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.
- 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 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 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 di 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 container inizia il lavoro di elaborazione, 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.

Usare una richiesta `PUT` in questo URI dall'interno di un container per impostare la protezione scalabile. Una richiesta `GET` a questo URI restituisce lo stato di protezione attuale di un'attività.

Parametri di richiesta di protezione scalabile

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

Specificare `true` per contrassegnare un'attività per la protezione. Specificare `false` per rimuovere la protezione e rendere l'attività idonea per l'interruzione.

Tipo: Booleano

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

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 risposta seguente:

```
{
  "protection": {
    "ExpirationDate": "2023-12-20T21:57:44.837Z",
    "ProtectionEnabled": true,
    "TaskArn": "arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"
  }
}
```

Parametri di risposta della protezione scalabile

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"
}
```

Cluster Amazon ECS per Fargate

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 dei processi 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 [Provider di capacità dei gruppi Auto Scaling](#).

Quando si utilizzano provider di capacità:

- È necessario associare un provider di capacità a un cluster prima di associarlo alla strategia del provider di capacità.
- È possibile specificare un massimo di 20 provider di capacità per una strategia del 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. Tutti i provider di capacità con un peso pari a zero non vengono utilizzati per inserire 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 Auto Scaling o Fargate, ma non entrambi.
- Un cluster può contenere una combinazione di servizi e attività autonome che utilizzano i provider di capacità. 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 terminazione 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 accade ciò, 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 i carichi di lavoro Fargate

Creazione di un cluster per definire l'infrastruttura in cui eseguire le attività e i 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 CloudFormation stack.

La console associa in modo automatico i provider di capacità Fargate e Fargate Spot al cluster.

È possibile modificare le seguenti opzioni:

- Aggiungi un namespace al cluster.

Un namespace consente ai servizi creati nel cluster di connettersi agli altri servizi nel namespace senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnessione dei servizi Amazon ECS](#).

- Abilita gli eventi delle attività per ricevere EventBridge notifiche relative alle modifiche dello stato delle attività.
- 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 [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- Assegna una AWS KMS chiave per la tua memoria effimera Fargate. Per informazioni su come creare una chiave, consulta [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- Configura la AWS KMS chiave e la registrazione per ECS Exec.

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 impostare un namespace diverso dal nome del cluster per Service Connect, in Impostazioni predefinite di Service Connect, per Namespace predefinito, seleziona o immetti il nome di un namespace. Per utilizzare un namespace condiviso, seleziona o inserisci un ARN del namespace. Per ulteriori informazioni sull'uso dei namespace condivisi, consulta [Amazon ECS Service Connect con namespace AWS Cloud Map condivisi](#).
6. (Facoltativo) Usa Container Insights, espandi Monitoraggio, quindi seleziona una delle seguenti opzioni:
 - Per usare la soluzione consigliata di Container Insights con osservabilità migliorata, seleziona Container Insights con osservabilità migliorata.
 - Per utilizzare Container Insights scegli Container Insights.
 7. (Facoltativo) Per abilitare gli eventi delle attività, espandi Eventi delle attività, quindi attiva Abilita gli eventi delle attività.

Quando abiliti gli eventi delle attività, Amazon ECS invia gli eventi di modifica dello stato delle attività a EventBridge. Ciò consente di monitorare e rispondere automaticamente alle modifiche del ciclo di vita delle attività.

8. (Facoltativo) Per usare ECS Exec per eseguire il debug delle attività nel cluster, espandi Configurazione della risoluzione dei problemi. Quindi configura quanto segue:
 - (Facoltativo) Per AWS KMS key for ECS Exec, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati della sessione ECS Exec.
 - (Facoltativo) Per la registrazione di ECS Exec, scegli la destinazione del log:

- Per inviare i log a CloudWatch Logs, scegli Amazon. CloudWatch
- Per inviare i log ad Amazon S3, seleziona Amazon S3.
- Per disabilitare la registrazione, seleziona Nessuno.

9. (Facoltativo) In Crittografia, puoi configurare quanto segue:

- Crittografa i tuoi dati sull'archiviazione temporanea di 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.

10. (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.

11. Scegli Create (Crea).

Fasi successive

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'implementazione di aggiornamenti continui di Amazon ECS](#)

Fornitori di capacità Amazon ECS per carichi di lavoro EC2

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 usare la funzionalità di dimensionamento gestito per permettere ad Amazon ECS di gestire le operazioni di riduzione e aumento orizzontale del gruppo Auto Scaling oppure puoi gestire le operazioni di dimensionamento autonomamente. Per ulteriori informazioni, consulta [Gestisci automaticamente la capacità di Amazon ECS con il dimensionamento automatico dei cluster](#).

Ti consigliamo di creare un nuovo gruppo Auto Scaling. 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. CloudFormation Sono preceduti da `EC2ContainerService-<ClusterName>`. Puoi utilizzare il gruppo Auto Scaling 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 Auto Scaling 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 Auto Scaling 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 un provider di capacità a un cluster prima di associarlo alla strategia del provider di capacità.
- È possibile specificare un massimo di 20 provider di capacità per una strategia del 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, viene utilizzato il valore predefinito di 0.
- 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. Tutti i provider di capacità con un peso pari a zero non vengono utilizzati per inserire 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. Questo consente l'esecuzione di qualsiasi processo di inizializzazione finale prima della messa in servizio dell'istanza. Per ulteriori informazioni, consulta [Configurazione delle istanze preinizializzate per il tuo gruppo Auto Scaling Amazon ECS](#).

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 carichi di lavoro Amazon EC2

Creazione di un cluster per definire l'infrastruttura in cui eseguire le attività e i 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 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 un namespace al cluster.

Un namespace consente ai servizi creati nel cluster di connettersi agli altri servizi nel namespace senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnessione dei servizi Amazon ECS](#).

- Abilita gli eventi delle attività per ricevere EventBridge notifiche relative alle modifiche dello stato delle attività.
- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- Assegna una AWS KMS chiave per la tua memoria effimera Fargate. Per informazioni su come creare una chiave, consulta [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- Configura la AWS KMS chiave e la registrazione per ECS Exec.
- 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. 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 di container, attualmente associati alle sottoreti. Puoi modificare le sottoreti e i gruppi di sicurezza per le istanze.

Puoi scegliere una sottorete esistente. È possibile usare un gruppo di sicurezza esistente o crearne uno nuovo. Per creare attività in una configurazione IPv6 solo, utilizza sottoreti che includono solo un blocco CIDR. IPv6

Quando crei un nuovo gruppo di sicurezza, devi 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 (origine)

Per consentire il traffico in entrata da un indirizzo o da un intervallo 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 CIDR: :/0 IPv4 . 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 impostare un namespace diverso dal nome del cluster per Service Connect, in Impostazioni predefinite di Service Connect, per Namespace predefinito, seleziona o immetti il nome di un namespace. Per utilizzare un namespace condiviso, seleziona o inserisci un ARN del namespace. Per ulteriori informazioni sull'uso dei namespace condivisi, consulta [Amazon ECS Service Connect con namespace AWS Cloud Map condivisi](#).
6. Aggiungi EC2 istanze Amazon al tuo cluster, espandi l'infrastruttura e quindi seleziona Fargate e le istanze autogestite.

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 decidi di usare le istanze spot, per la strategia di allocazione, scegli quali pool di capacità spot (tipi di istanza e zone di disponibilità) usare per le istanze.

Per la maggior parte dei carichi di lavoro, puoi scegliere l'opzione Capacità di prezzo ottimizzata.

Per ulteriori informazioni, consulta [Strategie di allocazione per istanze Spot](#) nella Amazon EC2 User Guide.

- Per l'istanza di 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 EBS root, immetti 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. Per una configurazione IPv6 solo, scegli un VPC con un blocco CIDR e sottoreti che hanno solo IPv6 un blocco CIDR. IPv6
- 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 Monitoraggio, quindi seleziona una delle seguenti opzioni:
- Per usare la soluzione consigliata di Container Insights con osservabilità migliorata, seleziona Container Insights con osservabilità migliorata.
 - Per utilizzare Container Insights scegli Container Insights.
9. (Facoltativo) Per abilitare gli eventi delle attività, espandi Eventi delle attività, quindi attiva Abilita gli eventi delle attività.

Quando abiliti gli eventi delle attività, Amazon ECS invia gli eventi di modifica dello stato delle attività a EventBridge. Ciò consente di monitorare e rispondere automaticamente alle modifiche del ciclo di vita delle attività.

10. (Facoltativo) Per usare ECS Exec per eseguire il debug delle attività nel cluster, espandi Configurazione della risoluzione dei problemi. Quindi configura quanto segue:
- (Facoltativo) Per AWS KMS key for ECS Exec, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati della sessione ECS Exec.
 - (Facoltativo) Per la registrazione di ECS Exec, scegli la destinazione del log:
 - Per inviare i log a CloudWatch Logs, scegli Amazon. CloudWatch

- Per inviare i log ad Amazon S3, seleziona Amazon S3.
- Per disabilitare la registrazione, seleziona Nessuno.

11. (Facoltativo)

Se utilizzi Runtime Monitoring con l'opzione manuale e desideri che questo cluster venga monitorato da GuardDuty, scegli Aggiungi tag e procedi come segue:

- In Chiave, inserisci **guardDutyRuntimeMonitoringManaged**
- In Valore, specifica **true**.

12. (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.

13. (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.

14. Scegli Create (Crea).

Fasi successive

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'implementazione di aggiornamenti continui di Amazon ECS](#)

Gestisci automaticamente la capacità di Amazon ECS con il dimensionamento automatico dei 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. Quando crei il provider di capacità del gruppo Auto Scaling Amazon ECS, attivi il dimensionamento gestito. Quindi, imposti una percentuale di destinazione (`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 operazioni di dimensionamento e scalabilità orizzontale in base all'utilizzo delle risorse utilizzate 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 un 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 usa i seguenti parametri per determinare quali operazioni intraprendere:

`CapacityProviderReservation`

La percentuale di istanze di container in uso per un provider di capacità specifico. 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 Auto Scaling. 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](#).

```
CapacityProviderReservation = (number of instances needed) / (number of running instances) x 100
```

DesiredCapacity

La quantità di capacità del gruppo Auto Scaling. 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 Auto Scaling 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. CloudWatch. 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 annullarne la registrazione dal cluster.

Il gruppo Auto Scaling 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 aumenta orizzontalmente 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 sarà in grado di gestire il dimensionamento.

- 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 strategia binpack è la più efficiente in termini di capacità.
- Quando la capacità di destinazione è inferiore al 100%, nell'ambito della strategia di collocamento, la strategia binpack deve avere un ordine superiore rispetto alla strategia spread. Ciò impedisce al provider di capacità di aumentare orizzontalmente fino a quando ogni attività non dispone di un'istanza dedicata o non viene raggiunto il limite.

Attivazione del dimensionamento automatico del cluster

Puoi attivare il dimensionamento automatico del cluster usando la console o la AWS CLI.

Quando crei un cluster che utilizza fornitori di EC2 capacità 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 carichi di lavoro Amazon EC2](#).

Puoi anche creare un gruppo Auto Scaling e poi assegnarlo a un cluster. Per ulteriori informazioni, consulta [Aggiornare un provider di capacità Amazon ECS](#).

Quando usi 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. Usa `put-cluster-capacity-providers` per modificare il provider di capacità del cluster. Per ulteriori informazioni, consulta [Attivazione del Dimensionamento automatico del cluster Amazon ECS](#).

Ottimizzazione del dimensionamento automatico del 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 il dimensionamento automatico dei cluster, puoi configurare Amazon ECS per scalare automaticamente il tuo gruppo Auto Scaling e concentrarti esclusivamente sull'esecuzione delle tue attività. Amazon ECS garantisce che il gruppo Auto Scaling aumenti e si riduca orizzontalmente in base alle necessità senza ulteriori interventi. I provider di capacità Amazon ECS vengono utilizzati per gestire l'infrastruttura nel cluster,

garantendo che vi siano istanze di container sufficienti a soddisfare le esigenze dell'applicazione. Per scoprire come funziona il dimensionamento automatico 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 eseguire le seguenti azioni per rendere il dimensionamento automatico dei cluster più reattivo e velocizzare le implementazioni:

Scalabilità graduale delle dimensioni del provider di capacità

I fornitori di capacità di Amazon ECS sceglieranno grow/shrink 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 potrebbe comportare un aumento dei tempi di implementazione, qualora fossero necessarie diverse istanze per l'esecuzione delle attività in sospeso. Puoi aumentare il valore [minimumScalingStepSize](#) tramite l'API Amazon ECS per aumentare il numero minimo di istanze che Amazon ECS scala in entrata o in uscita alla volta. Un valore [maximumScalingStepSize](#) troppo basso può limitare il numero di istanze di container scalate in entrata o in uscita alla volta, rallentando le implementazioni.

Note

Questa configurazione è attualmente disponibile solo tramite o.
[CreateCapacityProviderUpdateCapacityProvider](#) APIs

Periodo di preparazione dell'istanza

Il periodo di riscaldamento dell'istanza è il periodo di tempo dopo il quale un' EC2istanza Amazon appena lanciata può contribuire ai CloudWatch parametri per il gruppo Auto Scaling. Allo scadere del periodo di preparazione specificato, l'istanza viene conteggiata nelle metriche aggregate del gruppo Auto Scaling e il cluster di dimensionamento automatico procede con la successiva iterazione dei calcoli per stimare il numero di istanze richieste.

Il valore predefinito per [instanceWarmupPeriod](#) è 300 secondi, che puoi configurare su un valore inferiore tramite [CreateCapacityProvider](#) o [UpdateCapacityProvider](#) APIs per un ridimensionamento più reattivo. Consigliamo di impostare un valore superiore a 60 secondi in modo da evitare un eccessivo provisioning.

Capacità di riserva

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. Questo può ridurre significativamente la velocità di avvio delle attività. Sono quindi disponibili due opzioni.

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 usare la configurazione `Target Capacity` per indicare che vuoi mantenere una capacità di riserva nei cluster. Ad esempio, impostando `Target Capacity` all'80%, indichi che il cluster necessita di una capacità di riserva del 20% in ogni momento. Questa capacità di riserva consente di avviare immediatamente qualsiasi attività autonoma, garantendo che l'avvio delle attività non sia limitato. Il compromesso, nel caso di questo approccio, è il potenziale aumento dei costi legati al mantenimento della capacità di riserva del cluster.

Un approccio alternativo che si può prendere in considerazione è quello di aggiungere margine di manovra al tuo servizio, e non al provider di capacità. Ciò significa che invece di ridurre la configurazione della `Target Capacity` per avviare la capacità di riserva, puoi aumentare il numero di repliche nel servizio modificando la metrica di dimensionamento di monitoraggio delle destinazioni o le soglie di dimensionamento graduale del dimensionamento automatico. Tieni presente che questo approccio sarà utile solo per i carichi di lavoro soggetti a picchi, ma non avrà alcun effetto quando si implementano nuovi servizi e si passa da 0 a N attività per la prima volta. Per ulteriori informazioni sui criteri di dimensionamento correlati, consulta le [Target Tracking Scaling Policies](#) o [Step Scaling Policies](#) nella Guida per gli sviluppatori di Amazon Elastic Container Service.

Comportamento del dimensionamento gestito da Amazon ECS

Quando utilizzi provider di capacità del gruppo Auto Scaling che utilizzano il dimensionamento gestito, Amazon ECS stima il numero ottimale di istanze da aggiungere al cluster e utilizza questo valore per determinare il numero di istanze da richiedere o rilasciare.

Comportamento dell'aumento orizzontale gestito

Amazon ECS seleziona un provider di capacità per ogni attività seguendo la strategia del provider 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 per i provider di capacità vengono ignorate dai provider 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 Auto Scaling, 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 Auto Scaling, quel gruppo di attività non può essere eseguito con questo provider di capacità. Il provider di capacità non dimensiona il gruppo Auto Scaling. 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 Auto Scaling. 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 all'incremento minimo del dimensionamento.
 - Il valore inferiore di `maximumScalingStepSize` o il numero massimo di istanze calcolato.
- 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.

Il valore `targetCapacity` di default 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). L'aumento orizzontale è bloccato per le istanze che si trovano all'interno del `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 Auto Scaling 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 di container del provider di capacità non dispongono di risorse sufficienti per avviare l'attività, Amazon ECS limita il numero di attività con questo stato per ciascun cluster e impedisce l'esecuzione di qualsiasi attività al di sopra di tale limite. Per ulteriori informazioni, consulta [Service Quotas](#).

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 Auto Scaling 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. EC2 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 Auto Scaling per utilizzare la funzionalità di protezione da terminazione gestita del dimensionamento automatico del cluster.

La protezione dalla terminazione gestita consente al dimensionamento automatico del cluster di 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 percentuale `targetCapacity` in modo da avere capacità di riserva. Ciò consente di avviare più rapidamente le attività future, poiché il gruppo Auto Scaling non deve avviare ulteriori 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 affinché non sia richiesta alcuna operazione di dimensionamento. 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 Auto Scaling 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 Auto Scaling associato al provider di capacità. Le azioni manuali possono interrompere il processo di ridimensionamento del provider di capacità. Se scolleghi un'istanza dal gruppo Auto Scaling, devi anche [annullare la registrazione dell'istanza scollegata](#) dal cluster Amazon ECS.

Aggiornamento della protezione dalla terminazione gestita per i provider di capacità Amazon ECS

Quando si utilizza la protezione dalla cessazione gestita, è necessario aggiornare l'impostazione per i provider di capacità esistenti.

Console

1. [Apri la console nella versione 2. `https://console.aws.amazon.com/ecs/`](https://console.aws.amazon.com/ecs/)
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina Cluster, scegli la scheda Infrastruttura.
4. Seleziona il provider di capacità.
5. Seleziona Aggiorna per modificare le impostazioni del provider di capacità.
6. Nelle impostazioni del gruppo Auto Scaling, attiva la protezione dalla terminazione gestita per abilitare o disabilitare la funzionalità.
7. Scegliere Aggiorna.

AWS CLI

Puoi aggiornare l'impostazione di protezione dalla terminazione gestita di un provider di capacità con il comando: `update-capacity-provider`

Per abilitare la protezione dalla terminazione gestita:

```
aws ecs update-capacity-provider \
  --name CapacityProviderName \
  --auto-scaling-group-provider
  "managedScaling={status=ENABLED,targetCapacity=70,minimumScalingStepSize=1,maximumScalingStepS
```

Per disabilitare la protezione dalla terminazione gestita:

```
aws ecs update-capacity-provider \
  --name CapacityProviderName \
  --auto-scaling-group-provider
  "managedScaling={status=DISABLED,targetCapacity=70,minimumScalingStepSize=1,maximumScalingStepS
```

Note

Affinché le modifiche abbiano effetto su tutto il cluster potrebbero essere necessari alcuni minuti. Quando viene abilitata la protezione gestita dalla terminazione, le istanze che stanno già eseguendo attività saranno protette dagli eventi di riduzione orizzontale. Quando viene disabilitata la protezione dalla terminazione gestita, il flag di protezione verrà rimosso dalle istanze durante il ciclo di gestione del provider di capacità ECS successivo.

Console per l'esecuzione delle attività

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina del cluster, seleziona la scheda Attività.
4. Seleziona l'attività.
5. Nelle Configurazione, attiva la protezione dalla terminazione gestita per abilitare o disabilitare la funzionalità.
6. Seleziona Configura la protezione scalabile.

Viene visualizzata la finestra di dialogo Configura protezione scalabile

- a. In Protezione scalabile, seleziona Attiva.
- b. Per Scadenza in minuti, inserisci il numero di minuti della fine della protezione scalabile.
- c. Scegli Update (Aggiorna).

Attivazione del Dimensionamento automatico 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 vuoi usare la console per attivare il dimensionamento automatico del cluster, consulta [Creazione di un provider di capacità per Amazon ECS](#).

Prima di iniziare, crea un gruppo con scalabilità automatica e un provider di capacità. Per ulteriori informazioni, consulta [the section called “Fornitori di capacità Amazon ECS per carichi di lavoro EC2”](#).

Per attivare il dimensionamento automatico del cluster, associa il provider di capacità al cluster, quindi attiva il dimensionamento automatico del cluster.

1. Esegui il comando `put-cluster-capacity-providers` per associare uno o più provider di capacità al cluster.

Per aggiungere i fornitori AWS Fargate di capacità, includi i FARGATE fornitori di FARGATE_SPOT capacità nella richiesta. Per ulteriori informazioni, consulta [put-cluster-capacity-providers](#) nella documentazione di riferimento dei comandi 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 EC2, includi il nome del gruppo Auto Scaling nella richiesta. Per ulteriori informazioni, consulta [put-cluster-capacity-providers](#) nella documentazione di riferimento dei comandi della AWS CLI .

```
aws ecs put-cluster-capacity-providers \  
  --cluster ClusterName \  
  --default-capacity-provider-strategy capacityProvider=CapacityProvider,weight=1
```

```
--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 [describe-clusters](#) nella documentazione di riferimento dei comandi 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 [update-capacity-provider](#) nella documentazione di riferimento dei comandi della AWS CLI .

```
aws ecs update-capacity-provider \  
--name CapacityProviderName \  
--auto-scaling-group-provider "managedScaling={status=ENABLED}"
```

Disattivazione del Dimensionamento automatico 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 il dimensionamento automatico del cluster per un cluster, puoi annullare l'associazione del provider di capacità con il dimensionamento gestito attivato dal cluster o aggiornare il provider di capacità per disattivare il dimensionamento gestito.

Annullare l'associazione del 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 di capacità. AWS Fargate Per ulteriori informazioni, consulta [put-cluster-capacity-providers](#) nella documentazione di riferimento dei comandi 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 [put-cluster-capacity-providers](#) nella documentazione di riferimento dei comandi 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 [update-capacity-provider](#) nella documentazione di riferimento dei comandi della AWS CLI .

```
aws ecs update-capacity-provider \  
  --name CapacityProviderName \  
  --auto-scaling-group-provider "managedScaling={status=DISABLED}"
```

Creazione di un provider di capacità per Amazon ECS

Al termine della creazione del cluster, è possibile creare un nuovo provider di capacità (gruppo Auto Scaling) per EC2. I provider di capacità ti aiutano a gestire e scalare l'infrastruttura per le tue 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 Cluster.
3. Nella pagina Cluster, scegliere 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).

Aggiornare 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.

Aggiornamento 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 Cluster.

3. Nella pagina Cluster, scegliere 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. Scegliere Aggiorna.

Eliminazione di un provider di capacità Amazon ECS

Al termine dell'utilizzo di un provider di capacità del gruppo Auto Scaling, puoi eliminarlo. Una volta eliminato il 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 Cluster.
3. Nella pagina Cluster, scegliere 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. Scegli 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 di non terminazione. La manutenzione e gli aggiornamenti dell'infrastruttura vengono eseguiti senza preoccuparsi di interrompere i 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 delle istanze gestite da Amazon ECS funziona con le istanze di gruppo Auto Scaling sostitutive. In base all'aggiornamento delle istanze e alla durata massima delle stesse, i clienti possono garantire la conformità con i più recenti requisiti di sistema operativo e sicurezza per la loro capacità.

Il drenaggio delle istanze gestite da Amazon ECS può essere usato 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.

Gli eventi riportati di seguito sono coperti dal drenaggio delle istanze gestite da 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 numero elevato di istanze. L'aggiornamento dell'istanza viene avviato tramite la EC2 console Amazon o l'`StartInstanceRefreshAPI`. Assicurati di selezionare `Replace` per ottenere la protezione scalabile quando chiami `StartInstanceRefresh` se usi la protezione dalla terminazione gestita.
- [Durata massima dell'istanza](#): puoi definire una durata massima quando devi sostituire le istanze del gruppo Auto Scaling. Ciò è utile per pianificare le istanze sostitutive in base a policy di sicurezza interne o alla conformità.
- Riduzione orizzontale automatica del gruppo Auto Scaling: basato su policy di dimensionamento e azioni di dimensionamento pianificate, il gruppo Auto Scaling supporta il dimensionamento automatico delle istanze. Usando un gruppo Auto Scaling come provider di capacità Amazon ECS, puoi ridurre orizzontalmente le istanze di gruppo Auto Scaling quando non è in esecuzione alcuna attività.

- [Controlli dell'integrità del gruppo Auto Scaling](#): il gruppo Auto Scaling supporta molti controlli di integrità per gestire la terminazione delle istanze non integre.
- [CloudFormation aggiornamenti dello stack](#) - Puoi aggiungere un `UpdatePolicy` attributo allo 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 l'istanza precedente quando quella sostitutiva viene avviata e risulta funzionante. Il drenaggio delle istanze gestite da Amazon ECS drena le istanze spot allo stesso modo in cui drena le istanze non spot.
- [Interruzione spot](#): le istanze Spot vengono terminate con un preavviso di due minuti. Il drenaggio delle istanze gestite 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 ai clienti di creare soluzioni che vengono attivate da determinati eventi nel ciclo di vita dell'istanza ed eseguono 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. 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, è anche possibile configurare hook di terminazione del ciclo di vita personalizzati. Gli hook del ciclo di vita hanno default `action` e suggeriamo di impostare `continue` come impostazione predefinita per garantire che altri hook, come l'hook gestito da Amazon ECS, non siano influenzati da eventuali errori degli 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 da Amazon ECS, vengono eseguiti entrambi gli hook del ciclo di vita. Tuttavia, le tempistiche relative non sono garantite. Gli hook del ciclo di vita hanno un'impostazione default `action` per specificare l'azione da intraprendere allo scadere del timeout. In caso di errori, consigliamo di usare `continue` come risultato predefinito nel tuo hook personalizzato. Ciò garantisce che altri hook, in particolare quelli gestiti da Amazon ECS, non siano influenzati da eventuali errori nel tuo hook del ciclo di vita personalizzato. Il risultato alternativo di `abandon` comporta che tutti gli altri hook vengano ignorati, e questo 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

Drenaggio delle istanze gestite e delle attività

Il drenaggio delle istanze gestite da Amazon ECS usa la funzionalità di drenaggio esistente presente nelle istanze di container. La funzionalità di [drenaggio delle istanze di container](#) esegue la sostituzione e l'interruzione delle attività di replica che appartengono a un servizio Amazon ECS. Un'attività autonoma, come quella invocata da RunTask, che si trova nello stato PENDING o RUNNING, rimane inalterata. È necessario attendere che queste attività vengano completate o interromperle manualmente. L'istanza di container rimane nello stato DRAINING fino a quando tutte le attività non vengono interrotte o non sono trascorse 48 ore. Le attività daemon sono le ultime a essere interrotte dopo che tutte le attività di replica sono state interrotte.

Drenaggio delle istanze gestite e protezione dalla terminazione gestita

Il drenaggio delle istanze gestite funziona anche se la terminazione gestita è disabilitata. Per informazioni sulla protezione da terminazione gestita, consulta [Controlla le istanze terminate da Amazon ECS](#).

La seguente tabella riassume il comportamento per diverse combinazioni di terminazione gestita e drenaggio gestito.

Terminazione gestita	Drenaggio gestito	Outcome
Abilitato	Abilitato	Amazon ECS protegge EC2 le istanze Amazon che eseguono attività dall'interruzione a causa di eventi di scalabilità. Tutte le istanze

Terminazione gestita	Drenaggio gestito	Outcome
		in fase di terminazione, ad esempio quelle per cui non è impostata la protezione e dalla terminazione, che hanno subito una terminazione spot o sono state forzate dall'aggiornamento dell'istanza, vengono drenate correttamente.

Terminazione gestita	Drenaggio gestito	Outcome
Disabilitato	Abilitato	Amazon ECS non protegge EC2 le istanze Amazon che eseguono attività dalla scalabilità. Tuttavia, tutte le istanze che vengono terminate vengono svuotate in modo corretto.

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 terminate da un'interruzione spot o da un aggiornamento forzato dell'istanza, oppure se non stanno eseguendo alcuna attività. Amazon

Terminazione gestita	Drenaggio gestito	Outcome
		ECS non esegue il drenaggio graduale per queste istanze e avvia attività di servizio sostitutive dopo il loro arresto.

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 servizio sostituite dopo la loro interruzione.

Drenaggio delle istanze gestite e drenaggio delle istanze spot

Con il drenaggio delle istanze spot, è possibile impostare una variabile di ambiente `ECS_ENABLE_SPOT_INSTANCE_DRAINING` sull'agente Amazon ECS che consente ad Amazon ECS di mettere un'istanza in 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 separatamente il drenaggio delle istanze spot, quindi `ECS_ENABLE_SPOT_INSTANCE_DRAINING` nei dati utente del gruppo Auto Scaling è ridondante. Per ulteriori informazioni sulle 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 provider di capacità del gruppo Auto Scaling usando la console e la AWS CLI Amazon ECS.

Note

Il drenaggio delle istanze gestite è attivo per impostazione predefinita quando si crea 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.

Creare un provider di capacità con il drenaggio delle istanze gestite abilitato

Per creare un provider di capacità con lo svuotamento delle istanze gestito abilitato, usa il comando `create-capacity-provider`. Imposta il parametro `managedDraining` su `ENABLED`.

```
aws ecs create-capacity-provider \
--name capacity-provider \
--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 delle istanze gestite per il provider di capacità esistente di un cluster

Per abilitare il drenaggio delle istanze gestite per il provider di capacità esistente di un cluster, usa il comando `update-capacity-provider`. Come puoi vedere, `managedDraining` indica `DISABLED` e `updateStatus` indica `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 comando `describe-clusters` e includi ATTACHMENTS. Lo status dell'allegato di drenaggio dell'istanza gestita è PRECREATED e lo `attachmentsStatus` generale 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": []
}

```

Al termine dell'aggiornamento, utilizza `describe-capacity-providers` e vedrai che `managedDraining` è 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 gestite da Amazon ECS

Potrebbe capitarti di dover risolvere i problemi relativi al drenaggio delle istanze gestite. Di seguito è riportato un esempio di un problema che potresti incontrare durante l'utilizzo e della relativa risoluzione.

Le istanze non terminano dopo aver superato la durata massima dell'istanza quando si utilizza il dimensionamento automatico.

Se le istanze non vengono terminate neppure dopo aver raggiunto e superato la durata massima prevista durante l'utilizzo di un gruppo Auto Scaling, è possibile che siano protette dalla riduzione orizzontale. È possibile disattivare la terminazione gestita e consentire il drenaggio gestito per gestire il riciclo delle istanze.

Comportamento di drenaggio per le istanze gestite da Amazon ECS

Le istanze gestite da Amazon ECS implementano sofisticati processi di drenaggio e terminazione che garantiscono transizioni fluide dei carichi di lavoro, ottimizzando i costi e mantenendo l'integrità del sistema. Il sistema di terminazione offre tre percorsi decisionali distinti per la terminazione delle istanze, ciascuno con caratteristiche temporali e profili di impatto sul cliente diversi.

Percorsi decisionali per la terminazione

Tutti i percorsi di terminazione convergono sullo stesso meccanismo di esecuzione tramite l'hook del ciclo di vita `POST_DEREGISTER` che attiva l'API di Node Manager per la chiusura immediata delle istanze Amazon. `ReleaseNode EC2`

Terminazione avviata dal cliente

Consente il controllo diretto sulla rimozione delle istanze quando bisogna rimuovere immediatamente le istanze di container dal servizio. Si richiama l' `DeregisterContainerInstance` API con il flag `force` impostato su `true`, a indicare che è necessaria la terminazione immediata nonostante i carichi di lavoro in esecuzione.

Terminazione dello stato di inattività avviata dal sistema

Ottimizza i costi grazie al rilevamento intelligente dei periodi di inattività, il quale identifica le istanze che non gestiscono più carichi di lavoro. Elastic Workload Service (EWS) implementa sofisticati algoritmi di rilevamento dell'inattività che monitorano l'uso delle istanze e avviano la terminazione delle istanze che rimangono inattive per periodi configurabili.

Terminazione dell'aggiornamento dell'infrastruttura

Implementa la gestione proattiva dell'infrastruttura tramite la policy di decadimento naturale di Node Manager, in base alla quale le istanze vengono aggiornate periodicamente per garantire che funzionino sulle versioni più recenti della piattaforma e mantengano il livello di sicurezza. Node Manager implementa policy `time-to-live (TTL)` che avviano una terminazione graduale per le istanze che hanno raggiunto la loro durata operativa massima.

Drenaggio regolare e migrazione del carico di lavoro

Il sistema di drenaggio regolare implementa un sofisticato coordinamento con la gestione dei servizi Amazon ECS per garantire che le attività gestite dal servizio vengano correttamente migrate dalle istanze pianificate per la terminazione.

Coordinamento del drenaggio delle attività di servizio

Quando un'istanza passa allo stato `DRAINING`, il pianificatore di Amazon ECS interrompe automaticamente l'assegnazione di nuove attività all'istanza, implementando al contempo procedure di spegnimento graduale per le attività di servizio esistenti. Il drenaggio delle attività di servizio include il coordinamento con le strategie di implementazione dei servizi, i requisiti di controllo

dell'integrità e le preferenze di drenaggio dell'utente, al fine di garantire tempi di migrazione ottimali e tassi di successo elevati.

Gestione delle attività autonome

Per le attività autonome è necessaria una gestione diversa perché non beneficiano della gestione automatica dei servizi. Il sistema valuta le caratteristiche delle attività autonome, tra cui le stime della durata delle attività, l'analisi della probabilità di completamento e la valutazione dell'impatto sui clienti. La strategia di completamento graduale consente alle attività autonome di completarsi naturalmente durante un periodo di tolleranza prolungato, mentre la terminazione forzata garantisce che l'aggiornamento dell'infrastruttura avvenga entro tempi accettabili quando le attività non sono state completate naturalmente.

Strategia di completamento in due fasi

Il sistema di terminazione implementa un approccio in due fasi che bilancia la continuità del carico di lavoro e i requisiti di gestione dell'infrastruttura.

Fase 1: periodo di completamento graduale

Durante questa fase, il sistema implementa strategie di drenaggio graduale che danno priorità alla continuità del carico di lavoro. Le attività di servizio subiscono un drenaggio graduale attraverso le normali attività di pianificazione di Amazon ECS, le attività autonome continuano ad essere eseguite e possono essere completate naturalmente, mentre il sistema monitora tutte le attività affinché raggiungano lo stato di arresto attraverso attività di completamento naturali.

Fase 2: applicazione di scadenze rigide

Quando il completamento graduale non consente di raggiungere gli obiettivi di terminazione entro tempi accettabili, il sistema applica una scadenza rigida. Solitamente, la scadenza rigida è fissata al momento dell'avvio del drenaggio più sette giorni, in modo da garantire un tempo sufficiente per il completamento dell'operazione nel rispetto dei requisiti operativi. L'applicazione prevede l'invocazione automatica delle procedure di annullamento forzato della registrazione e l'immediata terminazione di tutte le attività rimanenti, indipendentemente dal loro stato di completamento.

Coordinamento tra servizi e gestione dello stato

Il processo di terminazione richiede un coordinamento sofisticato tra il Cluster Management Backend Service (CMBS) e Node Manager per garantire che l'annullamento della registrazione dell'istanza del contenitore e la pulizia delle risorse EC2 Amazon avvengano nella sequenza corretta, mantenendo la coerenza.

Esecuzione dell'hook POST_DEREGISTER

L'hook del ciclo di vita POST_DEREGISTER rappresenta il punto di convergenza in cui tutti e tre i percorsi decisionali per la terminazione eseguono la stessa logica di pulizia. Quando un'istanza del contenitore raggiunge lo stato DEREGISTERED, l'hook POST_DEREGISTER attiva automaticamente l'API di Node Manager ReleaseNode per avviare le operazioni di pulizia delle risorse Amazon. EC2 L'implementazione dell'hook include una sofisticata gestione degli errori per vari scenari di errore, tra cui problemi di connettività di rete, problemi di disponibilità dei EC2 servizi Amazon e errori di coordinamento tra i componenti del sistema.

Pulizia e deallocazione EC2 delle risorse Amazon

Il processo di terminazione delle EC2 istanze Amazon implementa un coordinamento completo con AWS i servizi per garantire che le risorse di elaborazione sottostanti vengano deallocate correttamente. Ciò include la pulizia dell'interfaccia di rete per prevenire perdite di risorse, la gestione dei record del database con audit trail completi e meccanismi adeguati di gestione e ripristino degli errori per diversi scenari di errore.

Creazione di risorse per il dimensionamento automatico dei cluster Amazon ECS usando la Console di gestione AWS

Scopri come creare le risorse per il dimensionamento automatico dei cluster usando la Console di gestione AWS. Se le risorse richiedono un nome, viene utilizzato il prefisso ConsoleTutorial per assicurare che tutti i nomi siano univoci e 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](#).

- L'utente IAM ha le autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS FullAccess](#).
- 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. 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 Auto Scaling 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 Auto Scaling `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 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 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 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 Auto Scaling 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 provider di capacità, un provider di capacità esterno o un provider di capacità del gruppo Auto Scaling, le attività vengono posizionate sulle 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 usando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'ultima AMI associata al sistema operativo selezionato.

Quando lo utilizzi 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 in modo da usare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized implementare. 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 usare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per l'utente AWS CloudFormation .

- [Recupero dei metadati dell'AMI Linux ottimizzata per Amazon ECS](#)
- [Recupero dei metadati dell'AMI Bottlerocket ottimizzata per Amazon ECS](#)
- [Recupero dei metadati dell'AMI Windows ottimizzata per Amazon ECS](#)

Puoi scegliere tra i tipi di istanza compatibili con la tua applicazione. Con istanze più grandi, puoi avviare più attività contemporaneamente. Le istanze più piccole ti consentono di impiegare l'aumento orizzontale in modo più dettagliato per ridurre i costi. Non è necessario scegliere un singolo tipo di EC2 istanza Amazon adatto a tutte le applicazioni del cluster. Puoi creare più gruppi 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.

Usa le seguenti linee guida per determinare i tipi di famiglia di istanza e il tipo di istanza da utilizzare:

- Elimina i tipi di istanza 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.
- Considera i requisiti, tra cui lo storage e il throughput di rete
- Prendi in considerazione 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 terminate 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 tue 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

Important

[L'AMI Amazon Linux 2 ottimizzata per Amazon ECS arriva il 30 end-of-life giugno 2026, rispecchiando la stessa data EOL del sistema operativo Amazon Linux 2 upstream \(per ulteriori informazioni, consulta Amazon Linux 2\). FAQs](#) Invitiamo i clienti ad aggiornare le loro applicazioni per usare Amazon Linux 2023, con supporto a lungo termine incluso fino al 2028. Per informazioni sulla migrazione da Amazon Linux 2 ad Amazon Linux 2023, consulta [Migrazione dall'AMI ottimizzata per Amazon ECS di Amazon Linux 2 all'AMI ottimizzata per Amazon ECS di Amazon Linux 2023](#).

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 istanze Amazon EC2. 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 usando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'ultima AMI associata al sistema operativo selezionato.

Quando lo utilizzi 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 in modo da usare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized implementare. 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 usare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per l'utente AWS CloudFormation .

Se devi personalizzare l'AMI ottimizzata per Amazon ECS, consulta [Amazon ECS Optimized AMI Build Recipes](#) su. GitHub

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	Description	Configurazione di archiviazione
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 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.

Sistema operativo	AMI	Description	Configurazione di archiviazione
			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	Description	Configurazione di archiviazione
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 processori AWS Graviton/Graviton 2/Graviton 3/Graviton 4 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	Description	Configurazione di archiviazione
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 la AWS CLI preinstallata.	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	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
			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	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
Amazon Linux 2 (GPU)	AMI Amazon ECS con kernel 5.10 ottimizzata per GPU	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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
Amazon Linux 2 (Neuron)	AMI Amazon Linux 2 (Neuron) kernel 5.10 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 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	Description	Configurazione di archiviazione
			xfs e Docker utilizza il driver di archiviazione overlay2. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Description	Configurazione di archiviazione
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 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 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	Description	Configurazione di archiviazione
			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. È possibile recuperare il nome AMI per ciascuna variante interrogando l'API Archivio dei parametri Systems Manager. Per ulteriori informazioni, consulta [Recupero dei metadati dell'AMI Linux ottimizzata 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 pacchetti ottimizzati per Amazon ECS AMIs basati su Amazon Linux contengono una versione personalizzata di cloud-init. Cloud-init è un pacchetto 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. Invece, ti consigliamo di eseguire l'aggiornamento a una nuova AMI ottimizzata per Amazon ECS non appena saranno disponibili le nuove versioni. 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 dei pacchetti e gli aggiornamenti di sicurezza più recenti, e che le versioni dei pacchetti rimangano invariate durante l'avvio delle istanze. Per ulteriori informazioni sul recupero delle AMI ottimizzate per Amazon ECS più recenti, consulta [Recupero dei metadati dell'AMI Linux ottimizzata per Amazon ECS](#).

Ti consigliamo di automatizzare l'ambiente per eseguire l'aggiornamento a una nuova AMI non appena viene resa disponibile. 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
```

Warning

L'aggiornamento dei pacchetti docker o containerd interromperà tutti i container in esecuzione sull'host, il che significa che tutte le attività in esecuzione di Amazon ECS verranno interrotte. Pianifica di conseguenza per ridurre al minimo le interruzioni del servizio.

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 la sezione [Uso di cloud-init su Amazon Linux 2](#) nella Guida per l'utente di Amazon Linux.

```
#cloud-config
repo_upgrade: security
```

Pacchetti con versione bloccata nella GPU 023 ottimizzata per Amazon ECS AL2 AMIs

Alcuni pacchetti sono fondamentali per il comportamento corretto e performante delle funzionalità della GPU nella GPU 023 ottimizzata per Amazon ECS AL2. AMIs Ciò include:

- `nvidia*Driver NVIDIA ()`
- `Moduli kernel () kmod*`
- `Librerie NVIDIA () libnvidia*`
- `Pacchetti kernel () kernel*`

Note

Questo elenco non è esaustivo. L'elenco completo dei pacchetti bloccati è disponibile con `dnf versionlock list`

Questi pacchetti sono vincolati alla versione per garantire la stabilità e prevenire modifiche involontarie che potrebbero interrompere i carichi di lavoro della GPU. Di conseguenza, questi pacchetti devono generalmente essere modificati entro i limiti di un processo gestito che gestisca correttamente i potenziali problemi e mantenga la funzionalità della GPU.

Per evitare modifiche involontarie, il `dnf versionlock` plugin viene utilizzato su questi pacchetti.

Se desideri modificare un pacchetto bloccato, puoi:

```
# unlock a single package
sudo dnf versionlock delete $PACKAGE_NAME

# unlock all packages
sudo dnf versionlock clear
```

Important

Quando sono necessari aggiornamenti a questi pacchetti, i clienti devono prendere in considerazione l'utilizzo della versione AMI più recente che include gli aggiornamenti richiesti.

Se è necessario aggiornare le istanze esistenti, è necessario adottare un approccio attento che preveda lo sblocco, l'aggiornamento e il riblocco dei pacchetti, assicurando sempre che la funzionalità della GPU sia mantenuta durante tutto il processo.

Recupero dei metadati dell'AMI Linux ottimizzata per Amazon ECS

È possibile recuperare i metadati dell'AMI ottimizzata per Amazon ECS a livello di programmazione. I metadati includono il nome AMI, la versione dell'agente container Amazon ECS e la versione runtime Amazon ECS che include la versione Docker.

Quando crei un cluster usando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'ultima AMI associata al sistema operativo selezionato.

Quando lo utilizzi 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 in modo da usare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized implementare. 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 usare il parametro Systems Manager, consulta [Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS](#) nella Guida per l'utente AWS CloudFormation .

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
- sms: GetParameter
- sms: 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 Amazon ECS con kernel 5.10 ottimizzato per GPU:

```
/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) kernel 5.10 ottimizzata 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 parametro secondario `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'AMI kernel 5.10 Amazon Linux 2 ottimizzata per Amazon ECS consigliata più recente

Puoi recuperare l'ID immagine dell'ID dell'AMI kernel 5.10 Amazon Linux 2 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-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 CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello 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
```

Migrazione da Amazon Linux 2 a un'AMI Amazon Linux 2023 ottimizzata per Amazon ECS

Dopo [Amazon Linux](#), Amazon ECS termina il supporto standard per Amazon Linux 2 ottimizzato per Amazon ECS a AMIs partire dal 30 giugno 2026. Dopo questa data, la versione dell'agente Amazon ECS viene bloccata e le nuove AMI Amazon Linux 2 ottimizzate per Amazon ECS AMIs vengono pubblicate solo quando viene aggiornata l'AMI Amazon Linux 2 di origine. La fine del ciclo di vita (EOL) completa avverrà il 30 giugno 2026, dopodiché non verrà più pubblicato Amazon Linux AMIs 2 ottimizzato per Amazon ECS, anche se l'AMI di origine viene aggiornata.

Amazon Linux 2023 offre un secure-by-default approccio con policy di sicurezza preconfigurate, SELinux in modalità permissiva, modalità IMDSv2 solo abilitata per impostazione predefinita, tempi di avvio ottimizzati e una migliore gestione dei pacchetti per una maggiore sicurezza e prestazioni.

Esiste un elevato grado di compatibilità tra Amazon Linux 2 e Amazon Linux 2023 ottimizzato per Amazon ECS AMIs e la maggior parte dei clienti riscontrerà minimal-to-zero cambiamenti nei carichi di lavoro tra i due sistemi operativi.

Per ulteriori informazioni, consulta la sezione [Confronto tra Amazon Linux 2 e Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023 e nella versione [AL2023 FAQs](#).

Considerazioni sulla compatibilità

Gestione dei pacchetti e aggiornamenti del sistema operativo

A differenza delle versioni precedenti di Amazon Linux, Amazon Linux 2023 AMIs ottimizzato per Amazon ECS è bloccato su una versione specifica del repository Amazon Linux. Ciò impedisce agli utenti di aggiornare inavvertitamente pacchetti che potrebbero apportare modifiche indesiderate

o sostanziali. Per ulteriori informazioni, consulta la sezione [Gestione dei repository e degli aggiornamenti del sistema operativo in Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023.

Versioni kernel Linux

Amazon Linux 2 AMIs si basa sui kernel Linux 4.14 e 5.10, mentre Amazon Linux 2023 utilizza i kernel Linux 6.1 e 6.12. Per ulteriori informazioni, consulta [Confronto tra Amazon Linux 2 e kernel Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023.

Modifiche alla disponibilità dei pacchetti

Di seguito sono riportate le modifiche più significative apportate ai pacchetti in Amazon Linux 2023:

- Alcuni pacchetti binari di origine in Amazon Linux 2 non sono più disponibili in Amazon Linux 2023. Per ulteriori informazioni, consulta [Pacchetti rimossi da Amazon Linux 2023](#) nelle Note di rilascio di Amazon Linux 2023.
- Modifiche nel modo in cui Amazon Linux supporta diverse versioni dei pacchetti. Il sistema `amazon-linux-extras` utilizzato in Amazon Linux 2 non esiste in Amazon Linux 2023. Tutti i pacchetti sono direttamente disponibili nel repository “core”.
- I pacchetti aggiuntivi per Enterprise Linux (EPEL) non sono supportati in Amazon Linux 2023. Per ulteriori informazioni, consulta la sezione [Compatibilità EPEL in Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023.
- Le applicazioni a 32 bit non sono supportate in Amazon Linux 2023. Per ulteriori informazioni, consulta [Funzionalità obsolete di Amazon Linux 2](#) nella Guida per l'utente di Amazon Linux 2023.

Modifiche ai gruppi di controllo (cgroup)

Un gruppo di controllo (cgroup) è una funzionalità del kernel Linux per organizzare gerarchicamente i processi e distribuire le risorse di sistema tra di essi. I gruppi di controllo vengono usati estensivamente per implementare un runtime di container e da `systemd`.

L'agente Amazon ECS, Docker e `containerd` supportano sia `cgroupv1` che `cgroupv2`. L'agente Amazon ECS e il runtime del container gestiscono i cgroup per te, quindi i clienti Amazon ECS non devono apportare modifiche per questo aggiornamento del cgroup sottostante.

Per ulteriori dettagli su `cgroupv2`, consulta [Gruppi di controllo v2 in Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023.

Modifiche al servizio di metadati di istanza (IMDS)

Amazon Linux 2023 richiede Instance Metadata Service versione 2 (IMDSv2) per impostazione predefinita. IMDSv2 offre diversi vantaggi che aiutano a migliorare il livello di sicurezza. Usa un metodo di autenticazione orientato alla sessione che prevede la creazione di un token segreto in una semplice richiesta HTTP PUT per avviare la sessione. Il token di una sessione può avere una validità compresa tra 1 secondo e 6 ore.

Per ulteriori informazioni su come passare da IMDSv1 a IMDSv2, consulta [Transition to using Instance Metadata Service Version 2](#) nella Amazon EC2 User Guide.

Se desideri utilizzarlo IMDSv1, puoi comunque farlo sovrascrivendo manualmente le impostazioni utilizzando le proprietà di avvio dell'opzione dei metadati dell'istanza.

Modifiche della swappiness di memoria

La swappiness di memoria per container non è supportata su Amazon Linux 2023 e cgroups v2. Per ulteriori informazioni, consulta [Gestione dello spazio di memoria di swap dei container su Amazon ECS](#).

Modifiche alla convalida FIPS

Amazon Linux 2 è certificato secondo lo standard FIPS 140-2 e Amazon Linux 2023 è certificato secondo lo standard FIPS 140-3.

Per abilitare la modalità FIPS su Amazon Linux 2023, installa i pacchetti necessari sulla tua EC2 istanza Amazon e segui i passaggi di configurazione utilizzando le istruzioni in [Abilita la modalità FIPS su Amazon Linux 2023](#) nella Guida per l'utente di Amazon Linux 2023.

Supporto delle istanze accelerato

Amazon Linux 2023 ottimizzato per Amazon ECS AMIs supporta sia i tipi di istanza accelerata Neuron che quelli con GPU. Per ulteriori informazioni, consulta [Linux ottimizzato per Amazon ECS AMIs](#).

Edificio personalizzato AMIs

Sebbene ti consigliamo di passare a Amazon ECS ottimizzato per AMIs Amazon Linux 2023, supportato e pubblicato ufficialmente, puoi continuare a creare versioni personalizzate di Amazon Linux 2 ottimizzate per Amazon ECS AMIs utilizzando gli script di build open source utilizzati per creare le varianti Linux dell'AMI ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta [Script di build per AMI Linux ottimizzata per Amazon ECS](#).

Strategie di migrazione

Consigliamo di creare e implementare un piano di migrazione che includa test approfonditi delle applicazioni. Le sezioni seguenti descrivono diverse strategie di migrazione in base alle modalità di gestione dell'infrastruttura Amazon ECS.

Migrazione con i provider di capacità Amazon ECS

1. Crea un nuovo provider di capacità con un nuovo modello di avvio. Questo dovrebbe fare riferimento a un gruppo Auto Scaling con un modello di avvio simile a quello esistente, ma invece dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS, dovrebbe specificare una delle varianti Amazon Linux 2023. Aggiungi questo nuovo provider di capacità al tuo cluster Amazon ECS esistente.
2. Aggiorna la strategia del provider di capacità predefinito del tuo cluster in modo da includere sia il provider di capacità Amazon Linux 2 esistente che il nuovo provider di capacità Amazon Linux 2023. Inizia con un peso maggiore sul provider Amazon Linux 2 e uno inferiore sul provider Amazon Linux 2023 (ad esempio, Amazon Linux 2: peso 80, Amazon Linux 2023: peso 20). Questo fa sì che Amazon ECS inizi a fornire istanze Amazon Linux 2023 man mano che vengono pianificate le nuove attività. Verifica che le istanze vengano registrate correttamente e che le attività possano essere eseguite correttamente sulle nuove istanze.
3. Regola gradualmente i pesi dei provider di capacità nella strategia predefinita del tuo cluster, aumentando il peso del provider Amazon Linux 2023 e diminuendo quello del provider Amazon Linux 2 nel tempo (ad esempio: 60/40, poi 40/60, poi 20/80). È inoltre possibile aggiornare le strategie dei singoli provider di capacità di servizio per dare priorità alle istanze Amazon Linux 2023. Monitora il posizionamento delle attività per accertarti che vengano eseguite correttamente sulle istanze di Amazon Linux 2023.
4. Facoltativamente, puoi svuotare le istanze di container Amazon Linux 2 per accelerare la migrazione delle attività. Se disponi di una capacità sostitutiva sufficiente di Amazon Linux 2023, puoi drenare manualmente le istanze di container Amazon Linux 2 tramite la console Amazon ECS o AWS CLI accelerare la transizione delle tue attività da Amazon Linux 2 ad Amazon Linux 2023. Quando la migrazione è completata, rimuovi il provider di capacità Amazon Linux 2 dal tuo cluster ed elimina il gruppo Auto Scaling associato.

Migrazione con un gruppo Amazon EC2 Auto Scaling

1. Crea un nuovo gruppo Amazon EC2 Auto Scaling con un nuovo modello di lancio. Questo dovrebbe essere simile a un modello di avvio esistente, ma invece dell'AMI Amazon Linux 2

ottimizzata per Amazon ECS, dovrebbe specificare una delle varianti Amazon Linux 2023.

Questo nuovo gruppo Auto Scaling può avviare istanze sul tuo cluster esistente.

2. Aumenta verticalmente il gruppo Auto Scaling in modo da iniziare a registrare le istanze di Amazon Linux 2023 nel tuo cluster. Verifica che le istanze vengano registrate correttamente e che le attività possano essere eseguite correttamente sulle nuove istanze.
3. Dopo aver verificato che le attività funzionano su Amazon Linux 2023, aumenta verticalmente il gruppo Auto Scaling Amazon Linux 2023 riducendo verticalmente il gruppo Auto Scaling Amazon Linux 2 in modo graduale, fino a sostituire completamente tutte le istanze Amazon Linux 2.
4. Se disponi di una capacità di sostituzione di Amazon Linux 2023 sufficiente, potresti voler drenare esplicitamente le istanze di container per accelerare la transizione delle tue attività da Amazon Linux 2 ad Amazon Linux 2023. Per ulteriori informazioni, consulta [Drenare le istanze di container di Amazon ECS](#).

Migrazione con istanze gestite manualmente

1. Avvia manualmente (o modifica gli script che avviano) nuove EC2 istanze Amazon utilizzando l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS anziché Amazon Linux 2. Verifica che queste istanze usino gli stessi gruppi di sicurezza, sottoreti, ruoli IAM e configurazione del cluster delle istanze Amazon Linux 2 esistenti. Le istanze dovrebbero registrarsi nel cluster Amazon ECS esistente automaticamente al momento dell'avvio.
2. Verifica che le nuove istanze Amazon Linux 2023 siano state registrate correttamente nel tuo cluster Amazon ECS e che siano nello stato ACTIVE. Verifica che le attività possano essere pianificate ed eseguite correttamente su queste nuove istanze aspettando il posizionamento naturale delle attività o manualmente che stopping/starting alcune attività attivino la riprogrammazione.
3. Sostituisci gradualmente le istanze di Amazon Linux 2 avviando istanze Amazon Linux 2023 aggiuntive in base alle necessità, quindi svuotando e terminando manualmente le istanze di Amazon Linux 2 una ad una. È possibile drenare le istanze tramite la console Amazon ECS impostando l'istanza sullo stato DRAINING, che interromperà l'assegnazione di nuove attività e consentirà alle attività esistenti di essere completate o ripianificate altrove.

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-amis](#) 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 [HashiCorppacker](#) 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. AMIs 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 a12
```

- AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS

```
make a12arm
```

- AMI Amazon ECS ottimizzata per GPU

```
make a12gpu
```

- AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS


```
make al2inf
```

- AMI Amazon Linux 2023 ottimizzata per Amazon ECS

```
make al2023
```

- AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS

```
make al2023arm
```

- AMI GPU Amazon Linux 2023 ottimizzata per Amazon ECS

```
make al2023gpu
```

- AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS

```
make al2023neu
```

Ottimizzato per Amazon ECS Bottlerocket AMIs

Bottlerocket è un sistema operativo open source Linux basato sullo scopo di AWS eseguire container su macchine virtuali o host bare metal. L'AMI Bottlerocket ottimizzata per Amazon ECS è sicura e include solo il numero minimo di pacchetti necessari per l'esecuzione dei container. Ciò migliora l'utilizzo delle risorse, riduce la superficie di attacco della sicurezza e aiuta a ridurre il sovraccarico di gestione. L'AMI Bottlerocket è inoltre integrato con Amazon ECS per contribuire a ridurre il sovraccarico operativo legato all'aggiornamento delle istanze di container in un cluster.

Bottlerocket differisce da Amazon Linux nei seguenti modi:

- Bottlerocket non include un gestore di pacchetti e il suo software può essere eseguito solo come container. Gli aggiornamenti a Bottlerocket vengono entrambi applicati e possono essere ripristinati in un unico passaggio, ciò riduce la probabilità di errori di aggiornamento.
- Il meccanismo principale per gestire gli host Bottlerocket è l'utilizzo di uno strumento di pianificazione di container. A differenza di Amazon Linux, l'accesso a singole istanze Bottlerocket è destinato a essere un'operazione poco frequente solo per scopi avanzati di debug e risoluzione dei problemi.

Per ulteriori informazioni su Bottlerocket, consulta la [documentazione](#) e le [versioni](#) su GitHub.

Esistono delle varianti dell'AMI Bottlerocket ottimizzata per Amazon ECS per kernel 6.1 e kernel 5.10.

Le seguenti varianti usano il kernel 6.1:

- `aws-ecs-2`
- `aws-ecs-2-nvidia`

Le seguenti varianti usano il kernel 5.10:

- `aws-ecs-1`
- `aws-ecs-1-nvidia`

Per ulteriori informazioni sulla variante `aws-ecs-1-nvidia`, consulta [Annuncio del supporto GPU NVIDIA per Bottlerocket su Amazon ECS](#).

Considerazioni

Quando utilizzi un'AMI Bottlerocket con Amazon ECS, considera gli aspetti seguenti.

- Bottlerocket supporta EC2 istanze Amazon con x86_64 arm64 processori. L'BottlerocketAMI non è consigliato per l'uso con EC2 istanze Amazon con un chip Inferentia.
- Le immagini Bottlerocket non includono un server SSH o uno shell (interprete di comandi). Tuttavia, puoi utilizzare strumenti di out-of-band gestione per ottenere l'accesso come amministratore SSH ed eseguire il bootstrap.

Per ulteriori informazioni, consulta le seguenti sezioni nel [README.md di Bottlerocket](#) su GitHub:

- [Esplorazione](#)
- [Container amministratore](#)
- Per impostazione predefinita, Bottlerocket ha un [container di controllo](#) abilitato. Questo contenitore esegue l'[AWS Systems Manager agente](#) che puoi utilizzare per eseguire comandi o avviare sessioni di shell su EC2 Bottlerocket istanze Amazon. 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 caratteristiche di sicurezza e sulle linee guida, consulta [Funzionalità di sicurezza](#) e [linee guida sulla sicurezza](#) su GitHub.

- La modalità di rete `awsvpc` 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'attività `initProcessEnabled` è supportato per l'AMI Bottlerocket versione `1.19.0` o successive.
- Bottlerocket AMIs Inoltre, non supportano i seguenti servizi e funzionalità:
 - ECS Anywhere
 - Service Connect
 - Amazon EFS in modalità crittografata
 - Amazon EFS in modalità di rete `awsvpc`
 - I volumi Amazon EBS non possono essere montati
 - Acceleratore di inferenze elastiche

Recupero dei metadati dell'AMI Bottlerocket ottimizzata per Amazon ECS

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.

Variante AMI Bottlerocket **aws-ecs-2**

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2` Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. Console di gestione AWS

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima Bottlerocket AMI ottimizzata per Amazon ECS consigliata con il AWS CLI seguente 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
```

- Console di gestione AWS: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella Console di gestione AWS. 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#
```

Variante AMI Bottlerocket **aws-ecs-2-nvidia**

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2-nvidia` Bottlerocket per regione e architettura con AWS CLI o. Console di gestione AWS

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima Bottlerocket AMI ottimizzata per Amazon ECS consigliata con il AWS CLI seguente 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
```

- Console di gestione AWS: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella Console di gestione AWS. 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#
```

Variante AMI Bottlerocket **aws-ecs-1**

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-1` Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. Console di gestione AWS

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima Bottlerocket AMI ottimizzata per Amazon ECS consigliata con il AWS CLI seguente 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
```

- Console di gestione AWS: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella Console di gestione AWS. 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
```

Variante AMI Bottlerocket **aws-ecs-1-nvidia**

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-1-nvidia` Bottlerocket per regione e architettura con AWS CLI o. Console di gestione AWS

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima Bottlerocket AMI ottimizzata per Amazon ECS consigliata con il AWS CLI seguente 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 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
```

- Console di gestione AWS: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella Console di gestione AWS. 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#
```

Fasi successive

Per un tutorial dettagliato su come iniziare a [usare il sistema Bottlerocket operativo su Amazon ECS](#), consulta [Using a Bottlerocket AMI with Amazon ECS](#) GitHub on [e Getting started with Bottlerocket e Amazon ECS](#) sul sito del blog. AWS

Per ulteriori informazioni sull'avvio di un'istanza Bottlerocket, consulta [Avvio di un'istanza Bottlerocket per Amazon ECS](#)

Avvio di un'istanza Bottlerocket per Amazon ECS

Puoi avviare un'istanza Bottlerocket in modo da poter eseguire i carichi di lavoro dei container.

Puoi usare il AWS CLI per avviare l'istanza Bottlerocket.

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 dei metadati dell'AMI Bottlerocket ottimizzata per Amazon ECS”](#) per ottenere l'ID dell'AMI Bottlerocket. Ti servirà per la fase successiva.
3. Esegui il comando seguente per avviare l'istanza Bottlerocket. Ricordati di sostituire i seguenti parametri:
 - Sostituiscilo *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 sistema Bottlerocket operativo su Amazon ECS](#), consulta [Using a Bottlerocket AML with Amazon ECS on e GitHub Getting started with e BottlerocketAmazon ECS](#) sul sito del blog. AWS

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](#)
- [Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei dati](#)
- [Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze spot](#)
- [Esecuzione di uno script quando si avvia un'istanza di container Linux di Amazon ECS](#)
- [Aumento delle interfacce di rete delle istanze di container Linux di Amazon ECS](#)
- [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#)
- [Gestione remota delle istanze di container Amazon ECS usando AWS Systems Manager](#)
- [Uso di un proxy HTTP per istanze di container Linux Amazon ECS](#)
- [Configurazione delle istanze preinizializzate per il tuo gruppo Auto Scaling Amazon ECS](#)
- [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 scoprire 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.

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 i [tipi di istanza](#) nella Amazon EC2 User Guide. Se desideri eseguire IPv6 solo un carico di lavoro, alcuni tipi di istanza non supportano IPv6 gli indirizzi. Per ulteriori informazioni, [IPv6consulta gli indirizzi](#) nella Amazon EC2 User Guide.

- 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 creane 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

Configura le impostazioni di rete in base alle necessità dopo aver selezionato il pulsante Modifica nella sezione Impostazioni di rete del modulo.

- Per VPC, seleziona il VPC nel quale desideri avviare l'istanza. Per eseguire IPv6 solo un carico di lavoro, scegli un VPC dual stack che includa sia IPv4 un blocco CIDR che uno. IPv6

- Per Sottorete, seleziona la sottorete in cui avviare l'istanza. Puoi 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.

Per eseguire un carico di lavoro IPv6 solo, scegli una sottorete che includa solo un blocco CIDR. IPv6

- **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 [Uso di un proxy HTTP per istanze di container Linux Amazon ECS](#) in questa guida.

- Se scegli un VPC dual stack e IPv6 una sottorete solo, per IPv6 Assegnazione automatica dell'IP, scegli **Abilita**.
- **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](#).
- Se avvii l'istanza per un IPv6 solo carico di lavoro, scegli **Configurazione di rete avanzata**, quindi per **Assegna IP primario**, scegli **Sì. IPv6**

Note

Senza un IPv6 indirizzo primario, le attività in esecuzione sull'istanza del contenitore nelle modalità di rete **host** o **bridge** non verranno registrate con i sistemi di bilanciamento del carico o con **AWS Cloud Map**

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 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, quando l'istanza viene avviata per la prima volta. 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 [Archiviazione della configurazione di un'istanza di container 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 di 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 impostazione predefinita, l'agente container Amazon ECS cercherà di rilevare la compatibilità dell'istanza del contenitore per una configurazione IPv6 solo esaminando i valori predefiniti IPv4 e IPv6 i percorsi dell'istanza. Per ignorare questo comportamento, puoi impostare il parametro `ECS_INSTANCE_IP_COMPATIBILITY` su `ipv4` o `ipv6` nel file dell'istanza `/etc/ecs/ecs.config`.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=your_cluster_name
ECS_INSTANCE_IP_COMPATIBILITY=ipv6
EOF
```

Per ulteriori informazioni, consulta [Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei dati](#).

Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei 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](#).

È possibile configurare l'istanza di container per trasmettere i dati nella configurazione dell'agente del container o nella configurazione del daemon 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](#).

 Note

AWS non supporta configurazioni Docker personalizzate, perché a volte possono entrare in conflitto con le future modifiche o funzionalità di Amazon ECS senza preavviso.

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 interrompe, 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 la funzione di drenaggio dell'istanza Spot di Amazon ECS è abilitata sull'istanza, Amazon ECS riceve l'avviso di interruzione dell'istanza spot e posiziona l'istanza nello stato 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.

La funzione di esaurimento dell'istanza spot è disattivata per impostazione predefinita.

Puoi attivare lo svuotamento dell'istanza spot all'avvio di un'istanza. Aggiungi il seguente script nel campo Dati utente. *MyCluster* Sostituiscilo 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 del container”](#).

```
curl http://localhost:51678/v1/metadata
```

Esecuzione di uno script quando si avvia un'istanza di container Linux di Amazon ECS

Potresti dover eseguire un container specifico per ogni istanza di container per gestire operazioni o problemi di sicurezza, ad esempio il monitoraggio, la sicurezza, i parametri, il rilevamento servizi o la creazione di log.

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 autorizzazioni per un ruolo](#) nella AWS Identity and Access Management Guida per l'utente .
2. Avvia una o più istanze di container usando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Avvia nuove istanze del contenitore e usa il seguente script di esempio nei dati EC2 utente. 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 delle istanze di container Linux di Amazon ECS

Note

Questa funzione non è disponibile su Fargate.

Ogni attività che utilizza la modalità di rete `awsvpc` riceve una propria interfaccia (ENI) di rete elastica, che è collegata all'istanza di container 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. 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.

Amazon ECS supporta il lancio di istanze di container con maggiore ENI densità utilizzando i tipi di istanze Amazon EC2 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 usare la console per attivare la funzionalità, consulta [Modifica delle impostazioni dell'account Amazon ECS](#). Per utilizzare l'opzione AWS CLI per attivare la funzionalità, consulta [Gestione delle impostazioni dell'account Amazon ECS tramite AWS CLI](#).

Ad esempio, un'istanza `c5.large` con `awsvpcTrunking` ha un limite ENI aumentato 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 EC2](#).

Considerazioni

Tieni in considerazione le seguenti informazioni quando usi la funzionalità di trunking ENI.

- Solo le varianti Linux dell'AMI ottimizzata per Amazon ECS o altre varianti Amazon Linux con versione 1.28.1 o successiva dell'agente del container e versione 1.28.1-2 o successiva del pacchetto ecs-init, supportano i limiti ENI aumentati. 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 i ENI limiti aumentati e l'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 Le attività che usano Fargate hanno sempre ricevuto un'ENI dedicata a prescindere da quante vengono avviate, pertanto questa funzionalità non è necessaria.
- Le attività devono essere avviate nello stesso Amazon VPC dell'istanza di container. 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 annullata la registrazione o terminata l'istanza di container, Amazon ECS eliminerà l'ENI trunk.

 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 delle istanze di container di 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 container in una sottorete pubblica con limiti ENI aumentati viene interrotta e quindi riavviata, l'istanza perde il suo indirizzo IP pubblico e l'agente del container perde la sua connessione.
- Quando abiliti `awsvpcTrunking`, le istanze di container ricevono un'ENI aggiuntiva che utilizza il gruppo di sicurezza predefinito del VPC ed è gestita 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 pubblica perché 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, in questo caso, nessuna istanza di container in esecuzione in tale sottorete può accedere a Internet. Puoi aggiungere o eliminare le regole dei gruppi di sicurezza per controllare il traffico in entrata e in uscita dalle sottoreti. Per ulteriori informazioni, consulta [Regole del gruppo di sicurezza](#) nella Guida per l'utente del Amazon Virtual Private Cloud.

Prerequisiti

Prima di avviare un'istanza di container con limiti ENI aumentati, è necessario completare i seguenti requisiti preliminari.

- 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 Console di gestione AWS. 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`. Consigliamo di creare 2 ruoli dell'istanza di container (`ecsInstanceRole`). È quindi possibile abilitare l'impostazione dell'account `awsvpcTrunking` per un ruolo e usare quel ruolo per attività che richiedono il trunking ENI. Per ulteriori informazioni sulle istanze di container, consulta [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 i ENI limiti maggiori. Per una lista di tipi di istanze supportate, consulta [Istanze supportate per un potenziamento delle interfacce di rete dei container Amazon ECS](#). 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 dei container con ENI limiti più elevati con il AWS CLI

Ogni istanza di container dispone di un'interfaccia di rete predefinita, nota anche come interfaccia di rete trunk. Utilizza il comando seguente per visualizzare un elenco di istanze di container con limiti ENI aumentati eseguendo una query dell'attributo `ecs.awsvpc-trunk-id`, che indica che dispone di 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-ECSAttributeList](#) ()AWS Tools for Windows PowerShell

```
Get-ECSAttributeList -TargetType container-instance -AttributeName ecs.aws-vpc-trunk-id -Region us-east-1
```

Istanze supportate per un potenziamento delle interfacce di rete dei container Amazon ECS

Di seguito vengono mostrati i tipi di EC2 istanze Amazon supportati e quante attività che utilizzano la modalità di aws-vpc rete possono essere avviate su ciascun tipo di istanza prima e dopo l'attivazione dell'impostazione dell'aws-vpcTrunkingaccount.

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 trunking ENI	Limite di attività con trunking ENI
a1.medium	1	10
a1.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m5a.16xlarge	14	120
m5a.24xlarge	14	120
m5ad.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m6a.xlarge	3	20
m6a.2xlarge	3	40
m6a.4xlarge	7	60
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m6gd.large	2	10
m6gd.xlarge	3	20
m6gd.2xlarge	3	40
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m6id.2xlarge	3	40
m6id.4xlarge	7	60
m6id.8xlarge	7	90
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m6in.2xlarge	3	40
m6in.4xlarge	7	60
m6in.8xlarge	7	90
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m7g.medium	1	4
m7g.large	2	10
m7g.xlarge	3	20
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.metal	14	120
m7i.large	2	10
m7i.xlarge	3	20

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m7i.2xlarge	3	40
m7i.4xlarge	7	60
m7i.8xlarge	7	90
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.12xlarge	7	120
m7i-flex.16xlarge	14	120
m8a.medium	1	4
m8a.large	2	10
m8a.xlarge	3	20
m8a.2xlarge	3	40

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m8a.4xgrande	7	60
m8a.8x grande	9	90
m8a.12 x grande	11	120
m8a. 16 x grande	15	120
m8a. 24x grande	15	120
m8a. 48 x grande	23	120
m8a.metal-24xl	15	120
m8a.metal-48xl	23	120
m8g.medium	1	4
m8g.large	2	10
m8g.xlarge	3	20
m8g.2xlarge	3	40
m8g.4xlarge	7	60
m8g.8xlarge	7	60
m8g.12xlarge	7	60
m8g.16xlarge	14	120
m8g.24xlarge	14	120
m8g.48xlarge	14	120
m8g.metal-24xl	14	120
m8g.metal-48xl	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m8 gb. Medio	1	4
m8 gb. grande	2	10
m8gb.xlarge	3	20
m8 gb.2xlarge	3	40
m8 gb.4xlarge	7	60
m8 gb.8xlarge	9	60
m8 gb.12xlarge	11	60
m8 gb. 16 x grande	15	120
m8 gb.24xlarge	23	120
m8gb.48xlarge	23	120
m8gd.medium	1	4
m8gd.large	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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m8gd.metal-24xl	14	120
m8gd.metal-48xl	14	120
m8 gn. Medio	1	4
m8gn.grande	2	10
m8gn.xlarge	3	20
m8gn.2x grande	3	40
m8gn. 4x grande	7	60
m8gn. 8x grande	9	60
m8 gn. 12 x grande	11	60
m8 gn. 16 x grande	15	120
m8 gn. 24 x grande	23	120
m8 gn. 48 x grande	23	120
m8i.large	2	10
m8i.xlarge	3	20
m8i.2xlarge	3	40
m8i.4xlarge	7	60
m8i.8xlarge	9	90
m8i.12xlarge	11	120
m8i.16xlarge	15	120
m8i.24xlarge	15	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
m8i.32xlarge	23	120
m8i.48xlarge	23	120
m8i.96xlarge	23	120
m8i.metal-48xl	23	120
m8i.metal-96xl	23	120
m8i-flex.large	2	4
m8i-flex.xlarge	3	10
m8i-flex.2xlarge	3	20
m8i-flex.4xlarge	7	40
m8i-flex.8xlarge	9	60
m8i-flex.12xlarge	11	120
m8i-flex.16xlarge	15	120
mac2.metal	7	12
mac2-m1ultra.metal	7	12
mac2-m2.metal	7	12
mac2-m2pro.metal	7	12
mac-m4.metal	7	12
mac-m4pro.metal	7	12

Calcolo ottimizzato

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c5ad.large	2	10
c5ad.xlarge	3	20
c5ad.2xlarge	3	40
c5ad.4xlarge	7	60

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c6a.12xlarge	7	120
c6a.16xlarge	14	120
c6a.24xlarge	14	120
c6a.32xlarge	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c6gd.8xlarge	7	60
c6gd.12xlarge	7	60
c6gd.16xlarge	14	120
c6gd.metal	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c6i.32xlarge	14	120
c6i.metal	14	120
c6id.large	2	10
c6id.xlarge	3	20

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c6in.32xlarge	15	120
c6in.metal	15	120
c7a.medium	1	4
c7a.large	2	10

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
c7g.12xlarge	7	60
c7g.16xlarge	14	120
c7g.metal	14	120
c7gd.medium	1	4

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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.metal	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
c7gn.metal	14	120
c7i.large	2	10
c7i.xlarge	3	20
c7i.2xlarge	3	40

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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.2xlarge	3	20
c7i-flex.4xlarge	7	40
c7i-flex.8xlarge	7	60
c7i-flex.12xlarge	7	120
c7i-flex.16xlarge	14	120
c8a. medio	1	4
c8a. grande	2	10
c8a.xlarge	3	20
c8a.2x grande	3	40
c8a.4x grande	7	60

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
c8a.8x grande	9	90
c8a.12 x grande	11	120
c8a.16 x grande	15	120
c8a. 24 x grande	15	120
c8a.48 x grande	23	120
c8a.metal-24xl	15	120
c8a.metal-48xl	23	120
c8g.medium	1	4
c8g.large	2	10
c8g.xlarge	3	20
c8g.2xlarge	3	40
c8g.4xlarge	7	60
c8g.8xlarge	7	60
c8g.12xlarge	7	60
c8g.16xlarge	14	120
c8g.24xlarge	14	120
c8g.48xlarge	14	120
c8g.metal-24xl	14	120
c8g.metal-48xl	14	120
c8 gb. medio	1	4

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
c8 gb. grande	2	10
c8gb.xlarge	3	20
c8 gb.2x grande	3	40
c8 gb.4xlarge	7	60
c8 gb.8xlarge	9	60
c8gb.12xlarge	11	60
c8 gb.16 x grande	15	120
c8 gb.24xlarge	23	120
c8 gb.48 x grande	23	120
c8gb.metal-24xl	23	120
c8gb.metallo-48xl	23	120
c8gd.medium	1	4
c8gd.large	2	10
c8gd.xlarge	3	20
c8gd.2xlarge	3	40
c8gd.4xlarge	7	60
c8gd.8xlarge	7	60
c8gd.12xlarge	7	60
c8gd.16xlarge	14	120
c8gd.24xlarge	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
c8gd.48xlarge	14	120
c8gd.metal-24xl	14	120
c8gd.metal-48xl	14	120
c8gn.medium	1	4
c8gn.large	2	10
c8gn.xlarge	3	20
c8gn.2xlarge	3	40
c8gn.4xlarge	7	60
c8gn.8xlarge	9	60
c8gn.12xlarge	11	60
c8gn.16xlarge	15	120
c8gn.24xlarge	23	120
c8gn.48xlarge	23	120
c8gn.metal-24xl	23	120
c8gn.metal-48xl	23	120
c8i.large	2	10
c8i.xlarge	3	20
c8 i.2x grande	3	40
c8i.4xgrande	7	60
c8i.8xgrande	9	90

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
c8i.12 x grande	11	120
c8i.16 x grande	15	120
c8i.24 x grande	15	120
c8i.32xlarge	23	120
c8i.48 x grande	23	120
c8i.96xlarge	23	120
c8i.metal-48xl	23	120
c8i.metal-96xl	23	120
c8i-flex.large	2	4
c8i-flex.xlarge	3	10
c8i-flex.2xgrande	3	20
c8i-flex.4xgrande	7	40
c8i-flex.8x grande	9	60
c8i-flex.12xgrande	11	120
c8i-flex.16 x grande	15	120

Memoria ottimizzata

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r5.large	2	10

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
r6id.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r6id. Metallo	14	120
r6idn.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r6in.metal	15	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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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.metal	14	120
r7i.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
r8a.medium	1	4
r8a.large	2	10
r8a.xlarge	3	20
r8a.2x grande	3	40
r8a.4xgrande	7	60
r8a.8xgrande	9	90
r8a.12 x grande	11	120
r8a.16 x grande	15	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r8a.24xlarge	15	120
r8a.48 x grande	23	120
r8a.metal-24xl	15	120
r8a.metal-48xl	23	120
r8g.medium	1	4
r8g.large	2	10
r8g.xlarge	3	20
r8g.2xlarge	3	40
r8g.4xlarge	7	60
r8g.8xlarge	7	60
r8g.12xlarge	7	60
r8g.16xlarge	14	120
r8g.24xlarge	14	120
r8g.48xlarge	14	120
r8g.metal-24xl	14	120
r8g.metal-48xl	14	120
r8gb.medium	1	4
r8gb.large	2	10
r8gb.xlarge	3	20
r8gb.2xlarge	3	40

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r8gb.4xlarge	7	60
r8gb.8xlarge	9	60
r8gb.12xlarge	11	60
r8gb.16xlarge	15	120
r8gb.24xlarge	23	120
r8gb. 48 x grande	23	120
r8gb.metal-24xl	23	120
r8gb.metal-48xl	23	120
r8gd.medium	1	4
r8gd.large	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
r8gd.24xlarge	14	120
r8gd.48xlarge	14	120
r8gd.metal-24xl	14	120
r8gd.metal-48xl	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r8gn.medium	1	4
r8gn.large	2	10
r8gn.xlarge	3	20
r8gn.2xlarge	3	40
r8gn.4xlarge	7	60
r8gn.8xlarge	9	60
r8gn.12xlarge	11	60
r8gn.16xlarge	15	120
r8gn.24xlarge	23	120
r8gn.48xlarge	23	120
r8gn.metal-24xl	23	120
r8gn.metal-48xl	23	120
r8i.large	2	10
r8i.xlarge	3	20
r8i.2xlarge	3	40
r8i.4xlarge	7	60
r8i.8xlarge	9	90
r8i.12xlarge	11	120
r8i.16xlarge	15	120
r8i.24xlarge	15	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r8i.32xlarge	23	120
r8i.48xlarge	23	120
r8i.96xlarge	23	120
r8i.metal-48xl	23	120
r8i.metal-96xl	23	120
r8i-flex.large	2	4
r8i-flex.xlarge	3	10
r8i-flex.2xlarge	3	20
r8i-flex.4xlarge	7	40
r8i-flex.8xlarge	9	60
r8i-flex.12xlarge	11	120
r8i-flex.16xlarge	15	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-6tb.112xlarge	14	120
u7i-8tb.112xlarge	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
u7i-12tb.224xlarge	14	120
u7in-16tb.224xlarge	15	120
u7in-24tb.224xlarge	15	120
u7in-32tb.224xlarge	15	120
u7inh-32tb.480xlarge	15	120
x2gd.medium	1	10
x2gd.large	2	10
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
x2iezn.4xlarge	7	120
x2iezn.6xlarge	7	120
x2iezn.8xlarge	7	120
x2iezn.12xlarge	14	120
x2iezn.metal	14	120
x8g.medium	1	4
x8g.large	2	10
x8g.xlarge	3	20
x8g.2xlarge	3	40
x8g.4xlarge	7	60
x8g.8xlarge	7	60
x8g.12xlarge	7	60
x8g.16xlarge	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
x8g.24xlarge	14	120
x8g.48xlarge	14	120
x8g.metal-24xl	14	120
x8g.metal-48xl	14	120
x8.aedz.large	3	10
x8aедz.xlarge	3	20
x8aедz.3x grande	7	40
x8aедz.6x grande	7	60
x8aедz.12x grande	15	120
x8aедz.24x grande	15	120
x8aедz.metallo-12xl	15	120
x8aедz.metallo-24xl	15	120
x8i.large	2	10
x8i.xlarge	3	20
x8i.2xlarge	3	40
x8i.4xlarge	7	60
x8i.8xlarge	9	90
x8i.12xlarge	11	120
x8i.16 x grande	15	120
x8i.24xlarge	15	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
x8i.32xlarge	23	120
x8i.48 x grande	23	120
x8i.64xlarge	23	120
x8i.96xlarge	23	120
x8i.metal-48xl	23	120
x8i.metallo-96xl	23	120

Archiviazione ottimizzata

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
i4i.12xlarge	7	118
i4i.16xlarge	14	248
i4i.24xlarge	14	118
i4i.32xlarge	14	498
i4i.metal	14	498
i7i.large	2	10
i7i.xlarge	3	20
i7i.2xlarge	3	40
i7i.4xlarge	7	60
i7i.8xlarge	7	90
i7i.12xlarge	7	90
i7i.16xlarge	14	120
i7i.24xlarge	14	120
i7i.48xlarge	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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
i8g.large	2	10
i8g.xlarge	3	20
i8g.2xlarge	3	40
i8g.4xlarge	7	60
i8g.8xlarge	7	60
i8g.12xlarge	7	60
i8g.16xlarge	14	120
i8g.24xlarge	14	120
i8g.48xlarge	14	120
i8g.metal-24xl	14	120
i8ge.large	2	20
i8ge.xlarge	3	29
i8ge.2xlarge	3	29

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
i8ge.3xlarge	5	29
i8ge.6xlarge	9	60
i8ge.12xlarge	11	60
i8ge.18xlarge	15	120
i8ge.24xlarge	15	120
i8ge.48xlarge	23	120
i8ge.metal-24xl	15	120
i8ge.metal-48xl	23	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 trunking ENI	Limite di attività con trunking ENI
d1.24xlarge	59	120
d12q.24xlarge	14	120
f2.6xlarge	7	90
f2.12xlarge	7	120
f2.48xlarge	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 trunking ENI	Limite di attività con trunking ENI
g5g.2xlarge	3	40
g5g.4xlarge	7	60
g5g.8xlarge	7	60
g5g.16xlarge	14	120
g5g.metal	14	120
g6.xlarge	3	20
g6.2xlarge	3	40
g6.4xlarge	7	60
g6.8xlarge	7	90
g6.12xlarge	7	120
g6.16xlarge	14	120
g6.24xlarge	14	120
g6.48xlarge	14	120
g6e.xlarge	3	20
g6e.2xlarge	3	40
g6e.4xlarge	7	60
g6e.8xlarge	7	90
g6e.12xlarge	9	120
g6e.16xlarge	14	120
g6e.24xlarge	19	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
g6e.48xlarge	39	120
g6f.large	1	10
g6f.xlarge	3	20
g6f.2xlarge	3	40
g6f.4xlarge	7	60
gr6.4xlarge	7	60
gr6.8xlarge	7	90
gr6f.4xlarge	7	60
g 7. 2x grande	3	242
G 7. 4x grande	7	242
G 7. 8x grande	7	242
G7 e.12 x grande	9	242
G7 e.24 x grande	19	242
G7 e.48 x grande	39	242
inf2.xlarge	3	20
inf2.8xlarge	7	90
inf2.24xlarge	14	120
inf2.48xlarge	14	120
p4d.24xlarge	59	120
p4de.24xlarge	59	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
p5.4xlarge	3	60
p5.48xlarge	63	242
p5e.48xlarge	63	242
p5en.48xlarge	63	242
p6-b200.48xlarge	31	242
p6-b 300,48 x grande	67	242
p6e-gb200.36xlarge	38	120
trn1.2xlarge	3	19
trn1.32xlarge	39	120
trn1n.32xlarge	79	242
trn2.48xlarge	31	242
trn2u.48xlarge	31	242
vt1.3xlarge	3	40
vt1.6xlarge	7	60
vt1.24xlarge	14	120

High Performance Computing

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
hpc6a.48xlarge	1	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
<code>hpc6id.32xlarge</code>	1	120
<code>hpc7g.4xlarge</code>	3	120
<code>hpc7g.8xlarge</code>	3	120
<code>hpc7g.16xlarge</code>	3	120

Allocazione della memoria di un'istanza di container Amazon ECS Linux

Quando l'agente container Amazon ECS registra un'istanza di container in un cluster, l'agente deve determinare quanta memoria è disponibile nell'istanza di container per essere allocata alle 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 si traduce sempre in esattamente 8192 MiB di memoria disponibile per le attività quando l'istanza di container viene registrata.

Determinazione delle risorse di memoria di ECS Managed Instances

Amazon ECS Managed Instances utilizza un approccio gerarchico per determinare i requisiti di risorse di memoria per le attività. A differenza di ECS, EC2 che si basa sull'introspezione della memoria di Docker, ECS Managed Instances calcola i requisiti di memoria direttamente dal payload dell'attività durante le decisioni di pianificazione.

Quando l'agente ECS Managed Instances riceve un'attività, calcola il fabbisogno di memoria utilizzando il seguente ordine di priorità:

1. Memoria a livello di attività (priorità massima): se la memoria a livello di attività è specificata nella definizione dell'attività, l'agente utilizza direttamente questo valore. Ciò ha la precedenza su tutte le impostazioni di memoria a livello di contenitore.
2. Somma di memoria a livello di contenitore (fallback): se la memoria a livello di attività non è specificata (o è 0), l'agente somma i requisiti di memoria di tutti i contenitori dell'operazione. Per ogni contenitore, utilizza:

- a. Prenotazione di memoria (soft limit): se un contenitore lo specifica `memoryReservation` nella sua configurazione, l'agente utilizza questo valore.
- b. Memoria del contenitore (limite rigido): se non `memoryReservation` è specificato, l'agente utilizza il `memory` campo del contenitore.

Example Memoria a livello di attività specificata

Quando viene specificata la memoria a livello di attività, ha la precedenza sulle impostazioni a livello di contenitore:

```
{
  "family": "my-task",
  "memory": "2048",
  "containerDefinitions": [
    {
      "name": "container1",
      "memory": 1024,
      "memoryReservation": 512
    }
  ]
}
```

L'agente riserva 2048 MiB (la memoria a livello di attività ha la precedenza).

Example Memoria a livello di contenitore con prenotazioni

Quando non viene specificata la memoria a livello di attività, l'agente somma i requisiti di memoria del contenitore:

```
{
  "family": "my-task",
  "containerDefinitions": [
    {
      "name": "container1",
      "memory": 1024,
      "memoryReservation": 512
    },
    {
      "name": "container2",
      "memory": 512
    }
  ]
}
```

```
]
}
```

L'agente riserva 512 MiB (prenotazione container1) + 512 MiB (memoria container2) = 1024 MiB totali.

L'agente ECS Managed Instances esegue il calcolo della memoria in tre fasi:

1. Ricezione delle attività: quando il payload di un'operazione arriva dal piano di controllo ECS, l'agente calcola immediatamente la memoria richiesta.
2. Archiviazione delle risorse: il fabbisogno di memoria calcolato viene memorizzato nel modello di attività per un uso successivo nelle operazioni di contabilità delle risorse.
3. Decisione sulla pianificazione: prima di accettare un'attività, l'agente verifica se è disponibile memoria sufficiente. Se la memoria disponibile è insufficiente, l'attività viene rifiutata e rimane nella coda dei servizi ECS fino a quando le risorse non diventano disponibili.

Note

A differenza di ECS on EC2, ECS Managed Instances non utilizza la variabile di configurazione `ECS_RESERVED_MEMORY`. La prenotazione della memoria per i processi di sistema viene gestita tramite la gestione delle risorse della piattaforma sottostante e l'agente esegue una contabilità accurata delle risorse in base alle definizioni delle attività.

Per ECS attivo EC2, 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 allocato alle tue attività. In questo modo si riserva la memoria per i processi di sistema critici.

Se occupi tutta la memoria di un'istanza di container con le tue attività, è possibile che queste ultime entrino in conflitto con le attività di sistema critiche per la memoria e causino 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 [Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei dati](#).

Se specifichi 8192 MiB per l'attività e nessuna delle istanze di container dispone di 8192 MiB o più di memoria disponibile per soddisfare questo requisito, l'attività non può essere inserita nel cluster. Se utilizzi un ambiente di elaborazione gestito, AWS Batch devi avviare un tipo di istanza più grande per soddisfare la richiesta.

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 a 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 il server AMI Amazon Windows ottimizzato per Amazon ECS.

```
TotalPhysicalMemory  
8589524992
```

Questa istanza ha 8589524992 byte di memoria totale, che si traducono in 8191 MiB disponibili per le attività.

Visualizzare la memoria di un'istanza di container

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 la maggior quantità possibile di memoria per un particolare tipo di istanza, puoi osservare la memoria disponibile per quell'istanza di container e quindi assegnare alle tue attività tale quantità di memoria.

Per visualizzare la memoria dell'istanza di container

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 e seleziona il cluster che ospita l'istanza di container.
3. Scegli Infrastruttura, quindi in Istanze di container scegli un'istanza di container.
4. La sezione Risorse mostra la memoria registrata e disponibile per l'istanza di container.

Il valore di memoria Registrato corrisponde a quello che l'istanza di container ha registrato in Amazon ECS la prima volta che è stata avviata e il valore di memoria Disponibile è quello che non è ancora stato allocato alle attività.

Gestione remota delle istanze di container Amazon ECS usando 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'uso 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 associare una policy a un ruolo, consulta [Aggiornare le autorizzazioni per un ruolo](#) nella Guida per l'utente di AWS Identity and Access Management
2. Verifica che SSM Agent sia installato nelle istanze di container. Per ulteriori informazioni, consulta [Installazione e disinstallazione manuale dell'agente SSM 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 dell'istanza di container 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	<code>\$ yum update -y</code>	
Istanze di destinazione	Le tue istanze di container	

Uso di un proxy HTTP per istanze di container Linux Amazon ECS

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). È anche possibile modificare manualmente il file di configurazione e poi 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 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/ecs.service.d/http-proxy.conf` (solo Amazon Linux 2)

```
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 proveniente 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 (solo Amazon Linux 2)

```
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 (solo AML Amazon Linux e Amazon Linux 2)

```
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 delle istanze preinizializzate per il tuo gruppo Auto Scaling Amazon ECS

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

In alcuni casi, potrebbe essere necessario aggiornare l'agente del container di Amazon ECS per ottenere correzioni dei 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 alla <https://console.aws.amazon.com/ecs/v2>.
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 del container di 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). Questo fornisce una nuova istanza di container con le versioni testate e convalidate più aggiornate di Amazon Linux, Docker, `ecs-init` e l'agente del container di 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 testate e convalidate più aggiornate di Docker e `ecs-init` che sono disponibili nei repository di Amazon Linux e la versione più recente dell'agente del container di 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 l'operazione `UpdateContainerAgent` API, tramite la console o con AWS CLI o 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 AWS CLI o AWS SDKs. Amazon ECS controlla la versione corrente dell'agente rispetto all'ultima versione disponibile e, se è possibile 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 alla <https://console.aws.amazon.com/ecs/v2>.
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

In alcuni casi, potrebbe essere necessario aggiornare l'agente del container di Amazon ECS per ottenere correzioni dei 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 [Archiviazione della configurazione di un'istanza di container 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 dei metadati dell'AMI Windows ottimizzata 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 completa per Windows Server 2025 ottimizzata per Amazon ECS
- AMI Core per Windows Server 2025 ottimizzata per Amazon ECS
- 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 l'ultima versione di Docker, ad esempio la 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti suggeriamo 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 2025, 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(ecs.os-type=='windows')`.
- I contenitori Windows sono supportati per le attività che utilizzano Fargate 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, consultare [Compatibilità delle versioni dei 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 dei metadati dell'AMI Windows ottimizzata per Amazon ECS”](#).

Recupero dei metadati dell'AMI Windows ottimizzata 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 [GetParameter](#) 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 AMI completi di Windows Server 2025:

```
/aws/service/ami-windows-latest/Windows_Server-2025-English-Full-ECS_Optimized
```

- Metadati AMI Core di Windows Server 2025:

```
/aws/service/ami-windows-latest/Windows_Server-2025-English-Core-ECS_Optimized
```

- 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 completa di Windows Server 2025 ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2025-English-Full-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2025 Core ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2025-English-Core-ECS_Optimized --region us-east-1
```

- 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 CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello 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 delle 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 dei metadati dell'AMI Windows ottimizzata 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 CloudFormation modello, vedere [Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello 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 l'ultima versione di Docker, ad esempio la 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti suggeriamo di passare a una delle seguenti piattaforme Windows:

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

 **Note**

La registrazione del plug-in gMSA è stata migrata dalla registrazione basata su file (C : \ProgramData\Amazon\gmsa) a Windows Event logging con la versione AMI di agosto 2025. Lo script di raccolta dei log pubblici raccoglierà tutti i log gMSA. Per ulteriori informazioni, consulta [Raccolta log dei container con il raccoglitore di log di Amazon ECS](#) .

Windows Server 2025 Full AMI versions

La tabella seguente elenca le versioni attuali e precedenti dell'AMI Windows Server 2025 Full ottimizzata per Amazon ECS e le versioni corrispondenti dell'agente contenitore Amazon ECS e Docker.

AMI completa per Windows Server 2025 ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2025-inglese-Full-ECS_Optimized-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-Full-ECS_Optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-Full-ECS_Optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-Full-ECS_Optimized-2025.07.16	1.96.0	25.0.6 (Docker CE)	Pubblica

AMI completa per Windows Server 2025 ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2025-inglese-Full-ECS_Optimized-2025.06.13	1.94.0	25.0.6 (Docker CE)	Pubblica

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2025 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2025-English-Full-ECS_Optimized
```

Windows Server 2025 Core AMI versions

La tabella seguente elenca le versioni attuali e precedenti dell'AMI Windows Server 2025 Core ottimizzata per Amazon ECS e le versioni corrispondenti dell'agente contenitore Amazon ECS e Docker.

AMI Core per Windows Server 2025 ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2025-inglese-core-ECS_Optimized-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-core-ecs_optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica

AMI Core per Windows Server 2025 ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2025-inglese-core-ecs-optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-core-ecs-optimized-2025.07.16	1.96.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2025-inglese-core-ecs-optimized-2025.06.13	1.94.0	25.0.6 (Docker CE)	Pubblica

Usa il seguente AWS CLI comando per recuperare l'attuale AMI Windows Server 2025 Core ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2025-English-Core-ECS_Optimized
```

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-inglese-Full-ECS_Optimized-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2025.07.16	1.95.0	25.0.6 (Docker CE)	Pubblica

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-English-Core-ECS_Optimized-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2025.07.16	1.95.0	25.0.6 (Docker CE)	Pubblica

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-English-Full-ECS_Optimized-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-English-Full-ECS_Optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-English-Full-ECS_Optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-English-Full-ECS_Optimized-2025.07.16	1.95.0	25.0.6 (Docker CE)	Pubblica

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-2025.09.13	1.99.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2025.08.24	1.98.0	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2025.08.16	1.97.1	25.0.6 (Docker CE)	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2025.07.16	1.95.0	25.0.6 (Docker CE)	Pubblica

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 l'ultima versione di Docker, ad esempio la 25.x.x. Pertanto, Windows Server 2016 Full non AMIs riceverà patch di sicurezza o di bug nel runtime Docker. Ti suggeriamo 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-Ingleso-Full-ECS_Optimized-2025.09.13	1.99.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2025.08.16	1.97.1	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2025.07.16	1.95.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2025.06.13	1.94.0	20.10.23 (Docker CE)	Pubblica

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 di origine 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 Console di gestione AWS.

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 Windows Amazon ECS

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](#)
- [Bootstrap di istanze di container Windows Amazon ECS per il trasferimento dei dati](#)
- [Uso di un proxy HTTP per istanze di container Windows Amazon ECS](#)
- [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 [Uso di un proxy HTTP per istanze di container Linux Amazon ECS](#) 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 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 avvia 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, quando l'istanza viene avviata per la prima volta. 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 nel namespace 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>
```

Bootstrap di istanze di container Windows Amazon ECS per il trasferimento dei 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 i carichi di lavoro 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>

```

Uso di un proxy HTTP per istanze di container Windows Amazon ECS

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 interrompe, 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 lo svuotamento dell'istanza 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 istanze esterne

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premises 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. Le architetture CPU `x86_64` e `ARM64` sono supportate.

- Amazon Linux 2023
- Ubuntu 20, Ubuntu 22, Ubuntu 24
- RHEL 9 — È necessario assicurarsi che Docker sia installato prima di eseguire lo script di installazione di [ECS Anywhere](#). Per ulteriori informazioni, consulta [Installare Docker Engine su RHEL](#) nella documentazione Docker.

A partire dal 7 agosto 2026, i seguenti sistemi operativi non sono più supportati da Amazon ECS Anywhere:

- Amazon Linux 2
- CentOS Stream 9
- RIGA 7, RIGA 8
- Fedora 32, Fedora 33, Fedora 40
- openSUSE Tumbleweed
- Ubuntu 18
- Debian 9, Debian 10, Debian 11, Debian 12
- SUSE Enterprise Server 15
- 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 tue istanze esterne richiedono un ruolo IAM che consenta loro di comunicare con 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 [Aggiunta di tag alle istanze di container esterne per Amazon ECS](#).
- ECS Exec è supportato sulle istanze esterne. Per ulteriori informazioni, consulta [Monitora i container 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.
- Il rilevamento servizi non è supportato.
- 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-premises e macchine virtuali con un'impronta digitale hardware](#) nella Guida per l'utente di AWS Systems Manager .
- È possibile eseguire attività Linux su istanze esterne in una configurazione IPv6 -only purché le istanze si trovino in sottoreti -only. IPv6 Per ulteriori informazioni, consulta [Utilizzo di un VPC in IPv6 modalità -only](#).
- 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 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 dell'istanza esterna aggiuntiva](#).

```
$ 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 bilanciatore del carico.

Di seguito sono riportate considerazioni aggiuntive specifiche per la rete con le istanze esterne.

- Il bilanciamento del carico dei servizi non è supportato.
- Il rilevamento servizi non è supportato.
- 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 le EC2 istanze](#).

- Puoi eseguire attività Linux su istanze esterne in una configurazione IPv6 -only purché le istanze si trovino in sottoreti -only. IPv6 Per ulteriori informazioni, consulta [Utilizzo di un VPC in IPv6 modalità -only](#).
- 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-premises 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

Important

Il supporto Windows per Amazon ECS Anywhere è obsoleto. Questa sezione non è più applicabile.

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

Important

Il supporto Windows per Amazon ECS Anywhere è obsoleto. Questa sezione non è più applicabile.

I seguenti casi d'uso erano supportati per ECS Anywhere quando Windows era un sistema operativo supportato.

- Active Directory è in Cloud AWS : Per questa configurazione, si crea una connessione tra la rete locale e l' Cloud AWS utilizzo di una AWS Direct Connect connessione. 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-premises. - Per questa configurazione, unisci le istanze esterne all'Active Directory on-premises. Quindi utilizzi le credenziali disponibili localmente quando esegui i processi di Amazon ECS.

Creazione di un cluster Amazon ECS per i carichi di lavoro delle istanze esterne

Creazione di un cluster per definire l'infrastruttura in cui eseguire le attività e i 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 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 un namespace al cluster.

Un namespace consente ai servizi creati nel cluster di connettersi agli altri servizi nel namespace senza configurazioni aggiuntive. Per ulteriori informazioni, consulta [Interconnessione dei servizi Amazon ECS](#).

- Configurazione del cluster per le istanze esterne

- Assegna una AWS KMS chiave per lo storage gestito. Per informazioni su come creare una chiave, consulta [Create a KMS key](#) nella Guida per l'utente di AWS Key Management Service .
- 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 il namespace utilizzato per Service Connect sia diverso dal nome del cluster, in Namespace, inserisci un nome univoco.

6. (Facoltativo) Usa Container Insights, espandi Monitoraggio, quindi seleziona una delle seguenti opzioni:

- Per usare la soluzione consigliata di Container Insights con osservabilità migliorata, seleziona Container Insights con osservabilità migliorata.
- Per utilizzare Container Insights scegli Container Insights.

7. (Facoltativo) Per usare ECS Exec per eseguire il debug delle attività nel cluster, espandi Configurazione della risoluzione dei problemi. Quindi configura quanto segue:

- Seleziona Attiva ECS Exec.
- (Facoltativo) Per AWS KMS key for ECS Exec, inserisci l'ARN della AWS KMS chiave che desideri utilizzare per crittografare i dati della sessione ECS Exec.
- (Facoltativo) Per la registrazione di ECS Exec, scegli la destinazione del log:
 - Per inviare i log a CloudWatch Logs, scegli Amazon. CloudWatch
 - Per inviare i log ad Amazon S3, seleziona Amazon S3.
 - Per disabilitare la registrazione, seleziona Nessuno.

8. (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.

9. (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.

10. Scegli Create (Crea).

Fasi successive

Devi registrare le istanze nel cluster. Per ulteriori informazioni, consulta [Registrazione di un'istanza esterna in un cluster Amazon ECS](#).

Crea una definizione dell'attività per il tipo di avvio esterno. Per ulteriori informazioni, consulta [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#)

Esegui le 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'implementazione di aggiornamenti continui di Amazon ECS](#)

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-premises 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](#).

Console di gestione AWS

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. 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-premises 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-premises 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
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-premises 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-premises 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-premises 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-premises 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 terminato 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. Potrai quindi registrarla in un nuovo cluster.

Procedura

Console di gestione AWS

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), 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 [Annullare la registrazione di nodi gestiti in ambienti ibridi e multicloud](#) nella Guida per l'utente di AWS Systems Manager .

8. Dopo aver annullato la registrazione dell'istanza, pulisci AWS le 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> 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	

Sistema operativo	Fasi	
	<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>	
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>	

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. sudo systemctl stop ecs amazon-ssm-agent b. Rimuovi i pacchetti Amazon ECS e Systems Manager. sudo (yum/apt/ zypper) remove amazon-ecs-init amazon-ssm-agent c. Rimuovi le directory rimanenti. sudo rm -rf /var/ lib/ecs /etc/ecs / var/lib/amazon/ss m /var/log/ecs / var/log/amazon/ssm	
Windows	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza.	

Sistema operativo	Fasi	
	Stop-Service AmazonECS Stop-Service AmazonSSMAgent b. Rimuovi il pacchetto Amazon ECS. .\ecs-anywhere-installer.ps1 -Uninstall	

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 <code>ecs-init</code>
CentOS 7 (x86_64)	<code>amazon-ecs-init-latest.x86_64.rpm</code>
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)	<code>amazon-ecs-init-latest.aarch64.rpm</code>
CentOS 8 (aarch64)	
CentOS Stream 9 (aarch64)	
RHEL 7 (aarch64)	
Debian 9 (x86_64)	<code>amazon-ecs-init-latest.amd64.deb</code>
Debian 10 (x86_64)	

Sistemi operativi (architettura)	Pacchetto ecs-init
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)	
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](#). 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 fornitore di capacità di default

Ogni cluster ha uno o più fornitori di capacità e una strategia di fornitori di capacità facoltativa. La strategia del fornitore di capacità determina il modo in cui le attività vengono distribuite tra i fornitori di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del fornitore 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 container di 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 Cluster.
3. Nella pagina Cluster, scegliere 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. Scegliere Aggiorna.

Eliminare 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 il namespace . Per ulteriori informazioni, consulta [Eliminazione degli spazi dei nomi](#) nella Guida per gli sviluppatori di AWS Cloud Map .

Procedura

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 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 bilanciatore del carico o nel gruppo di destinazione. Se un processo orfano utilizza la modalità di rete `aws-vpc`, 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 CloudFormation stack di Auto Scaling, interrompi l'istanza aggiornando il gruppo o lo stack Auto Scaling. CloudFormation In caso contrario, il gruppo Auto Scaling o 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, interrompi l' EC2istanza Amazon sottostante. Per ulteriori informazioni, consulta [Terminate Your Instance](#) nella Amazon EC2 User Guide.

Drenare le istanze di container di Amazon ECS

In alcuni casi, potresti dover rimuovere un'istanza di container da un cluster; ad esempio, per eseguire aggiornamenti di sistema o ridurre verticalmente 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 di 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 bilanciatore del carico sono in stato RUNNING, vengono considerate integre. Le attività per i servizi che utilizzano un bilanciatore del carico vengono considerate integre se sono in stato RUNNING e se il bilanciatore del carico 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.

Comportamento di drenaggio per le istanze gestite da Amazon ECS

I processi di terminazione delle istanze gestite da Amazon ECS garantiscono transizioni fluide dei carichi di lavoro, ottimizzando i costi e mantenendo l'integrità del sistema. Il sistema di terminazione offre tre percorsi decisionali distinti per la terminazione delle istanze, ciascuno con caratteristiche temporali e profili di impatto sul cliente diversi.

Terminazione avviata dal cliente

Consente il controllo diretto sulla rimozione delle istanze quando bisogna rimuovere immediatamente le istanze di container dal servizio. Si esegue `deregister-container-instance` con il parametro di richiesta `force` impostato su `true`. Ciò significa che è richiesta la terminazione immediata nonostante eventuali carichi di lavoro in esecuzione.

Terminazione dello stato di inattività avviata dal sistema

Ottimizza i costi grazie al rilevamento intelligente dei periodi di inattività, il quale identifica le istanze che non gestiscono più carichi di lavoro. Elastic Workload Service (EWS) implementa sofisticati algoritmi di rilevamento dell'inattività che monitorano l'uso delle istanze e avviano la terminazione delle istanze che rimangono inattive per periodi configurabili.

Terminazione dell'aggiornamento dell'infrastruttura

Implementa la gestione proattiva dell'infrastruttura tramite la policy di decadimento naturale di Node Manager, in base alla quale le istanze vengono aggiornate periodicamente per garantire che funzionino sulle versioni più recenti della piattaforma e mantengano il livello di sicurezza.

Node Manager implementa politiche time-to-live (TTL) che avviano una terminazione graduale per le istanze che hanno raggiunto la loro durata operativa massima.

Il sistema di terminazione implementa un approccio in due fasi che bilancia la continuità del carico di lavoro e i requisiti di gestione dell'infrastruttura.

Fase 1: periodo di completamento graduale

Durante questa fase, il sistema implementa strategie di drenaggio graduale che danno priorità alla continuità del carico di lavoro. Le attività di assistenza vengono eseguite regolarmente tramite le normali attività di pianificazione di Amazon ECS. Le attività autonome continuano a funzionare perché potrebbero completarsi naturalmente. Il sistema controlla che tutte le attività raggiungano lo stato di arresto tramite attività di completamento naturali.

Fase 2: applicazione di scadenze rigide

Quando il completamento graduale non consente di raggiungere gli obiettivi di terminazione entro tempi accettabili, il sistema applica una scadenza rigida. Solitamente, la scadenza rigida è fissata al momento dell'avvio del drenaggio più sette giorni, in modo da garantire un tempo sufficiente per il completamento dell'operazione nel rispetto dei requisiti operativi. L'applicazione prevede delle procedure automatiche di annullamento forzato della registrazione e l'immediata terminazione di tutte le attività rimanenti, indipendentemente dal loro stato di completamento.

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 considera nuovamente l'istanza di container per il posizionamento dell'attività.

Procedura

Le seguenti fasi possono essere utilizzate per impostare lo svuotamento di un'istanza di container utilizzando la nuova Console di gestione AWS.

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.

Console di gestione AWS

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere 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.

Modifica del tipo o della dimensione dell'istanza all'esterno del gruppo Auto Scaling

AWS consiglia di mantenere l'infrastruttura immutabile. Se devi modificare le dimensioni delle istanze, puoi:

- Scalate orizzontalmente e aggiungete altre istanze. Quindi, posiziona attività aggiuntive su quelle istanze o su
- Scalate verticalmente lanciando una nuova larger/smaller istanza, quindi svuotate quella precedente.

Entrambi questi approcci contribuiranno a ridurre al minimo l'impatto sulla disponibilità delle applicazioni.

Se hai utilizzato un altro metodo per modificare l'istanza, potresti ricevere il seguente errore:

```
Container instance type changes are not supported.
```

Quando ricevi questo errore, esegui le seguenti operazioni:

1. Avvia nuove istanze con il tipo di istanza desiderato.
2. Svuota i tuoi vecchi tipi di istanza. Per ulteriori informazioni, consulta [the section called "Esaurimento dell'istanza di container"](#).

Istanze di EC2 container Amazon ECS

L'agente Amazon ECS è un processo che viene eseguito su tutte le istanze di container registrate nel cluster. Facilita la comunicazione tra le istanze di container e Amazon ECS.

Note

Nelle istanze di container Linux, il container dell'agente monta directory di primo livello come `/lib`, `/lib64` e `/proc`. Questo è necessario per caratteristiche e funzionalità ECS come i volumi Amazon EBS, la modalità di rete `awsvpc`, 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 container Amazon ECS contiene l'immagine `amazon-ecs-pause`. Amazon ECS usa questa immagine per attività che sfruttano la modalità di rete di `awsvpc`.

[Per scoprire quali funzionalità e miglioramenti sono inclusi in ogni versione 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'EC2istanza 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 [Drenare le istanze di container di 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.

Supporto Docker

Amazon ECS supporta le ultime due principali versioni di Docker pubblicate su Amazon Linux. Attualmente, sono Docker 20.10.x e Docker 25.x.

La versione Docker minima richiesta per Amazon ECS è disponibile nel file delle [specifiche di Amazon ECS Agent](#) su GitHub

Quando si usa l'AMI ottimizzata per Amazon ECS, Docker è preinstallato e configurato per funzionare con l'agente container Amazon ECS. L'AMI comprende una versione Docker che è stata testata ed è supportata da Amazon ECS.

Note

Sebbene Amazon ECS supporti più versioni di Docker, suggeriamo di usare la versione di Docker fornita con l'AMI ottimizzata per Amazon ECS per ottenere la massima compatibilità e il miglior supporto.

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 dei metadati dell'AMI Linux ottimizzata 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 del container di Amazon ECS supporta una serie di opzioni di configurazione, la maggior parte delle quali viene impostata 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 [Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei 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 “Bootstrap delle istanze di 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
```

Esegui l'agente Amazon ECS con il namespace PID host

Per impostazione predefinita, l'agente Amazon ECS viene eseguito con il proprio namespace PID. Nelle seguenti configurazioni, puoi configurare l'agente Amazon ECS per l'esecuzione con il namespace PID host:

- SELinux la modalità di applicazione è abilitata.

- La politica di SELinux sicurezza di Docker è impostata su true.

Puoi configurare questo comportamento impostando la variabile di ambiente `ECS_AGENT_PID_NAMESPACE_HOST` su `true` nel tuo file `/etc/ecs/ecs.config`. Quando questa variabile è abilitata, `ecs-init` avvia il contenitore dell'agente Amazon ECS con lo spazio dei nomi PID dell'host (`--pid=host`), permettendo all'agente di avviarsi correttamente negli ambienti - enforcing. SELinux Questa caratteristica è disponibile nelle versioni 1.94.0 e successive dell'agente Amazon ECS.

Per abilitare questa funzionalità, aggiungi la seguente riga al file `/etc/ecs/ecs.config`:

```
ECS_AGENT_PID_NAMESPACE_HOST=true
```

Dopo aver apportato la modifica, riavvia l'agente Amazon ECS per renderla effettiva:

```
sudo systemctl restart ecs
```

Le seguenti funzionalità non funzioneranno: la modalità di SELinux imposizione è abilitata e la policy di sicurezza Docker è impostata su true, anche quando è impostata.

`ECS_AGENT_PID_NAMESPACE_HOST=true`

- Amazon ECS Exec
- allega attività Amazon EBS
- Service Connect
- FireLens per Amazon ECS

Parametri disponibili

Per informazioni sui parametri di configurazione dell'agente container Amazon ECS disponibili, consulta [Amazon ECS Container Agent on. GitHub](#)

Archiviazione della configurazione di un'istanza di container Amazon ECS in Amazon S3

La configurazione dell'agente del container Amazon ECS viene 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 `AmazonS3ReadOnlyAccess` a `ecsInstanceRole`. Per informazioni su come associare una policy a un ruolo, consulta [Aggiornare le autorizzazioni per un ruolo](#) nella Guida per l'utente di 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=dockerconfig
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 [Creazione di 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 utente. EC2 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. Per farlo, puoi scaricare l'agente da uno dei bucket Amazon S3 regionali o da Amazon Elastic Container Registry Public. Se esegui il download da uno dei bucket Amazon S3 regionali, puoi eventualmente verificare la validità del file dell'agente del container 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 una non istanza Amazon Linux EC2

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.

Region	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)
us-west-1	Stati Uniti occidentali (California settentrionale)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
us-west-2	Stati Uniti occidentali (Oregon)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ap-east-1	Asia Pacifico (Hong Kong)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)

Region	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ap-northeast-1	Asia Pacifico (Tokyo)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ap-northeast-2	Asia Pacifico (Seul)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
ap-south-1	Asia Pacifico (Mumbai)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)
		Amazon ECS init arm64 (arm64)	Amazon ECS init aarch64 (aarch64)
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)

Region	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
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)
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)

Region	Nome della Regione	File di init deb di Amazon ECS	File di init rpm di Amazon ECS
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 agente non AMI Amazon Linux

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. Per ulteriori informazioni sul supporto di Docker, consulta [Istanze di EC2 container Amazon ECS](#).

```
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 file `/lib/systemd/system/ecs.service` e aggiungi la seguente riga alla fine della sezione `[Unit]`.

```
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 [Archiviazione della configurazione di un'istanza di container 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 \(http://169.254.169.254\)](http://169.254.169.254) del servizio di [metadati dell' EC2 istanza Amazon](#) per i contenitori avviati dall'agente contenitore. 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 log 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 informazioni dettagliate su tutti i parametri di configurazione disponibili, consulta [Configurazione dell'agente del container Amazon ECS](#) o il file [README di Amazon ECS Agent](#) su GitHub.

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"
```

Configurare 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 EC2.

Un altro metodo per abilitare l'autenticazione del registro privato consiste nell' **Gestione dei segreti AWS** archiviare in modo sicuro le credenziali del registro privato e quindi farvi riferimento nella definizione del contenitore. In questo modo le attività possono utilizzare immagini da repository privati. Questo metodo supporta le attività che utilizzano uno 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 [Archiviazione della configurazione di un'istanza di container 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 del container”](#).

```
curl http://localhost:51678/v1/metadata
```

Pulizia automatica dell'immagine e dell'attività 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

Il tempo di attesa predefinito per eliminare i container per un'attività interrotta. Se il valore è inferiore a 1 secondo, il valore impostato verrà ignorato. 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.

Note

Il processo di pulizia delle immagini non può eliminare un'immagine finché c'è un container che fa riferimento alla stessa. Dopo la rimozione dei container, tutte le immagini senza riferimenti diventano candidate alla pulizia in base ai parametri di configurazione della pulizia delle immagini.

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.

Pianificare i container su Amazon ECS

Amazon Elastic Container Service (Amazon ECS) è un sistema di simultaneità ottimistica di stato condiviso che fornisce funzionalità di pianificazione flessibili per 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. Consente inoltre di eseguire le attività indipendenti o pianificate per processi 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 utilizzare	Ulteriori informazioni
Servizio	Il pianificatore di servizi è ideale per le applicazioni e i servizi stateless di lunga durata. Il pianificatore di servizi, facoltativamente, assicura anche che le attività siano registrate nel bilanciamento del carico di bilanciamento del carico elastico. 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	Servizi Amazon ECS

Opzione	Quando utilizzare	Ulteriori informazioni
	vincoli e strategie di posizionamento dei processi per personalizzare le decisioni riguardo al posizionamento dei processi.	
Processo autonomo	Un'attività autonoma è adatta a processi come i processi batch che eseguono un lavoro e quindi si arrestano . 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 utilizzare	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, è possibile specificare l'infrastruttura su cui vengono eseguite le attività o i servizi. È possibile usare una strategia per il provider di capacità o un tipo di avvio.

Per Fargate, i provider 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 per i 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`.

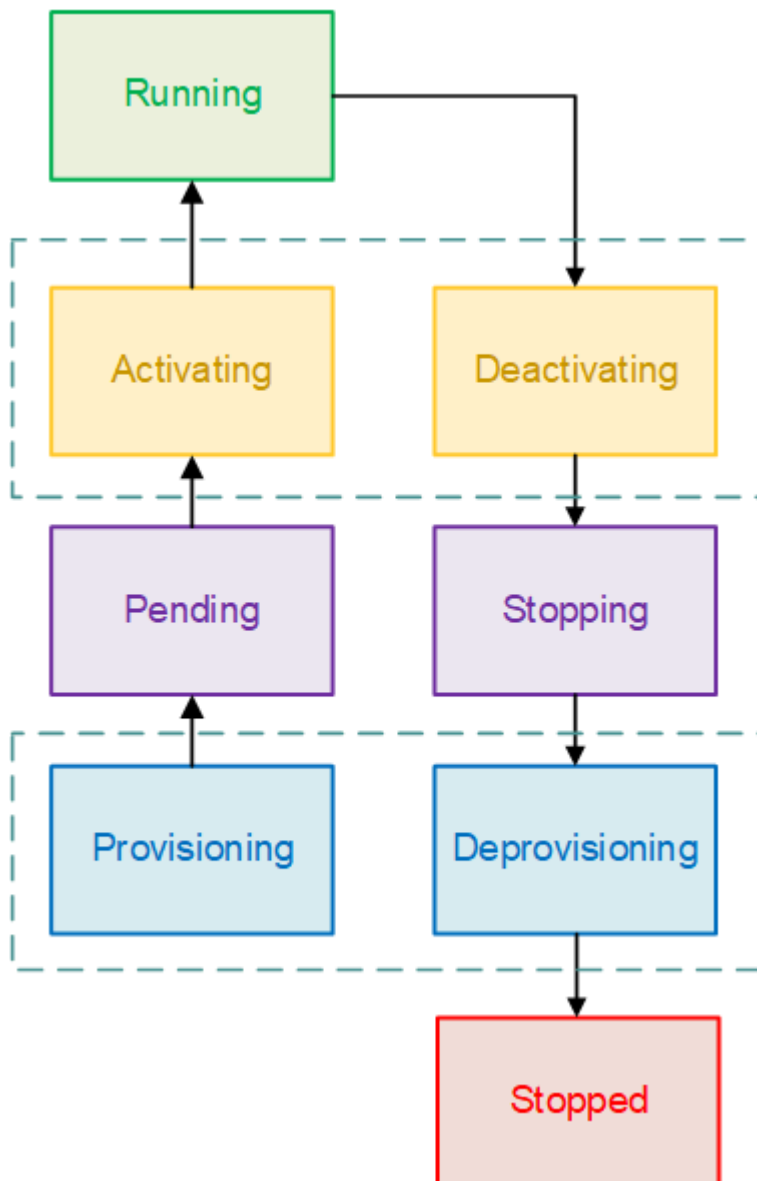
L'opzione di calcolo avvia le tue attività direttamente su Fargate o sulle EC2 istanze Amazon che hai registrato manualmente nei tuoi cluster.

Ciclo di vita dei processi 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.

IN ATTESA

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, crea i container, configura il networking delle attività, registra i gruppi di destinazione dei bilanciatori del carico e configura il rilevamento 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 processi che fanno parte di un servizio che è configurato per l'utilizzo di gruppi di destinazione di Elastic Load Balancing, l'annullamento della registrazione del gruppo di destinazione si verifica 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 segnale di arresto definito nell'immagine del contenitore per notificare che l'applicazione deve terminare e chiudere utilizzando le istruzioni. `STOPSIGNAL` Questo è `SIGTERM` impostazione predefinita. 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'attività è stata interrotta a causa di un errore, consultare [Visualizza gli errori relativi alle attività interrotte in 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 dei container

È possibile 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 istanze di container per il posizionamento delle attività o attività per il termine. Ad esempio, Amazon ECS è in grado di selezionare istanze di container a caso oppure in modo tale che i processi vengano distribuiti in modo uniforme all'interno di 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 di container. 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 attività solo su un particolare tipo di istanza.

Amazon ECS dispone di algoritmi differenti per le diverse opzioni di capacità.

Istanze gestite da Amazon ECS

Per le attività eseguite su Istanze gestite da Amazon, Amazon ECS deve determinare dove posizionare l'attività e, quando si riduce verticalmente il numero di attività, quali attività terminare. Amazon ECS effettua questa determinazione in base ai requisiti dell'istanza specificati nel modello di avvio del provider di capacità, ai requisiti specificati nella definizione dell'attività come CPU e memoria e ai vincoli di posizionamento delle attività.

 Note

Le istanze gestite da Amazon ECS non supportano strategie di posizionamento delle attività. Amazon ECS farà del suo meglio per distribuire le attività tra zone di disponibilità accessibili.

Quando Amazon ECS posiziona i processi, utilizza la seguente procedura per selezionare le istanze di container:

1. Identificazione delle istanze di container che soddisfano i requisiti di memoria, CPU, GPU e porta nella definizione di attività.
2. Identificazione delle istanze di container che soddisfano i vincoli di posizionamento delle attività.
3. Identificazione delle istanze di container che soddisfano i requisiti di istanza specificati nel modello di lancio del provider di capacità.
4. Selezione delle istanze di container per il posizionamento delle attività.

EC2

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 dei processi di default dipendono dal fatto che i processi vengano eseguiti manualmente (attività indipendenti) 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 è presente un vincolo predefinito per il posizionamento delle attività non nei servizi. Per ulteriori informazioni, consultare [Pianificare i container 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. Identificazione delle istanze di container che soddisfano i requisiti di memoria, CPU, GPU e porta nella definizione di attività.
2. Identificazione delle istanze di container che soddisfano i vincoli di posizionamento delle attività.
3. Identificazione delle istanze di container che soddisfano le strategie di posizionamento delle attività.
4. Selezione delle istanze di container per il posizionamento delle attività.

Fargate

Per attività con 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.

Dimensionamento automatico di Istanze gestite da Amazon ECS e posizionamento delle attività

Le istanze gestite da Amazon ECS utilizzano algoritmi intelligenti per scalare automaticamente la capacità del cluster e posizionare le attività in modo efficiente nell'infrastruttura. Capire come funzionano questi algoritmi aiuta a ottimizzare le configurazioni dei servizi e a risolvere i problemi di posizionamento.

Algoritmo di posizionamento delle attività

Le istanze gestite da Amazon ECS utilizzano un sofisticato algoritmo di posizionamento che bilancia disponibilità, utilizzo delle risorse e requisiti di rete durante la pianificazione delle attività.

Diffusione della zona di disponibilità

Per impostazione predefinita, le istanze gestite da Amazon ECS danno priorità alla disponibilità distribuendo le attività su più zone di disponibilità:

- Per i servizi con più attività, le istanze gestite da Amazon ECS garantiscono la distribuzione su almeno 3 istanze in diverse zone di disponibilità, quando possibile.
- Questo comportamento fornisce tolleranza ai guasti, ma può comportare un minore utilizzo delle risorse per istanza
- La diffusione della zona di disponibilità ha la precedenza sull'ottimizzazione del bin packing

Comportamento di bin packing

Sebbene le istanze gestite da Amazon ECS siano in grado di eseguire il bin packing per massimizzare l'utilizzo delle risorse, questo comportamento è influenzato dalla configurazione di rete:

- Per eseguire il bin packing, configurare il servizio in modo che utilizzi una singola sottorete
- Le configurazioni a più sottoreti danno priorità alla distribuzione delle zone di disponibilità rispetto alla densità delle risorse
- Il bin packing è più probabile durante l'avvio iniziale del servizio che durante gli eventi di dimensionamento

Considerazioni sulla densità dell'ENI

Per i servizi che utilizzano la modalità di rete `awsvpc`, le istanze gestite da Amazon ECS considerano la densità dell'interfaccia di rete elastica (ENI) quando prendono decisioni di posizionamento:

- Ogni attività in modalità `awsvpc` richiede un ENI dedicato
- I tipi di istanza hanno limiti ENI diversi che influiscono sulla densità delle attività
- Le istanze gestite da Amazon ECS tengono conto della disponibilità ENI durante la selezione delle istanze di destinazione

Note

Vengono continuamente apportati miglioramenti ai calcoli della densità ENI per ottimizzare le decisioni di posizionamento.

Logica decisionale del provider di capacità

I provider di capacità delle istanze gestite da Amazon ECS prendono decisioni di scalabilità e posizionamento in base a diversi fattori:

Requisiti per le risorse

- Requisiti di CPU, memoria e rete per le attività in sospeso

Disponibilità delle istanze

- Capacità e utilizzo attuali tra le istanze esistenti

Vincoli di rete

- Configurazione della sottorete e disponibilità ENI

Distribuzione delle zone di disponibilità

- Mantenimento della tolleranza ai guasti in più zone di disponibilità

Opzioni di configurazione

Strategia di selezione delle sottoreti

La configurazione della sottorete influisce in modo significativo sul comportamento di posizionamento delle attività:

Più sottoreti (impostazione predefinita)

- Assegna la priorità alla distribuzione della zona di disponibilità per un'elevata disponibilità

- Può comportare un minore utilizzo di risorse per istanza

- Consigliato per carichi di lavoro di produzione che richiedono tolleranza ai guasti

Sottorete singola

- Consente il bin packing per un maggiore utilizzo delle risorse

- Riduce la tolleranza ai guasti concentrando le attività in un'unica zona di disponibilità

- Adatto a carichi di lavoro di sviluppo o ottimizzati in termini di costi

Considerazioni sulla modalità di rete

La modalità di rete selezionata influisce sulle decisioni di posizionamento:

- Modalità `awsvpc`: ogni attività richiede un ENI dedicato, limitando la densità delle attività per istanza
- Modalità `host`: le attività utilizzano direttamente la rete dell'host, con il posizionamento basato principalmente sulla disponibilità delle risorse

Considerazioni sull'architettura della CPU

Quanto `cpuArchitecture` specificato nella definizione dell'attività viene utilizzato per collocare le attività su un'architettura specifica. Se non specifichi `cpuArchitecture`, Amazon ECS proverà a collocare le attività su qualsiasi architettura CPU disponibile in base alla configurazione del provider di capacità. È possibile specificare `X86_64` o `ARM64`.

Risoluzione dei problemi di posizionamento delle attività

Schemi comuni di posizionamento

La comprensione degli schemi di posizionamento previsti aiuta a distinguere il comportamento normale dai potenziali problemi:

Distribuzione sparsa

Attività distribuite su più istanze con utilizzo parziale

Comportamento normale quando si utilizzano più sottoreti

Indica la priorità della disponibilità rispetto all'efficienza delle risorse

Posizionamento concentrato

Attività multiple assegnate a un minor numero di istanze con un utilizzo più elevato

Previsto quando si utilizza la configurazione a sottorete singola

Può verificarsi durante l'avvio iniziale del servizio

Distribuzione irregolare

Alcune istanze sono molto utilizzate mentre altre rimangono sottoutilizzate

Può indicare limiti ENI o vincoli di risorse

Valutare la possibilità di rivedere i tipi di istanza e la configurazione

Ottimizzazione del comportamento di posizionamento

Per ottimizzare il posizionamento delle attività in base a esigenze specifiche:

1. Valutare i requisiti di disponibilità rispetto alle esigenze di ottimizzazione dei costi
2. Scegliere la configurazione di sottorete appropriata in base alle priorità
3. Selezionare i tipi di istanza con una capacità ENI adeguata per la modalità di rete
4. Monitorare i modelli di posizionamento e regolare la configurazione secondo necessità

Best practice

- Per carichi di lavoro di produzione: utilizzare più sottoreti in diverse zone di disponibilità per garantire un'elevata disponibilità, accettando il compromesso in termini di utilizzo delle risorse
- Per lo sviluppo o il test: prendere in considerazione la configurazione di una singola sottorete per massimizzare l'utilizzo delle risorse e ridurre i costi
- Per la modalità **awsvpc**: scegliere i tipi di istanza con una capacità ENI sufficiente per evitare vincoli di posizionamento
- Per l'ottimizzazione dei costi: monitorare i modelli di utilizzo e regolare la configurazione del servizio per bilanciare disponibilità ed efficienza
- Per la risoluzione dei problemi: esaminare la configurazione della sottorete e la modalità di rete quando si esaminano schemi di posizionamento imprevisti

Usare le strategie per definire il posizionamento delle attività su 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 dei processi di default dipendono dal fatto che i processi vengano eseguiti manualmente (attività indipendenti) 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 è presente un vincolo predefinito per il posizionamento delle attività non nei servizi. Per ulteriori informazioni, consultare [Pianificare i container 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. Identificazione delle istanze di container che soddisfano i requisiti di memoria, CPU, GPU e porta nella definizione di attività.
2. Identificazione delle istanze di container che soddisfano i vincoli di posizionamento delle attività.
3. Identificazione delle istanze di container che soddisfano le strategie di posizionamento delle attività.
4. Selezione delle istanze di container per il posizionamento delle attività.

È possibile specificare le strategie di posizionamento delle attività nella definizione del servizio o la definizione delle attività utilizzando il parametro `placementStrategy`.

```
"placementStrategy": [  
  {  
    "field": "The field to apply the placement strategy against",  
    "type": "The placement strategy to use"  
  }  
]
```

Puoi specificare le strategie quando esegui un'attività ([RunTask](#)), crei un nuovo servizio ([CreateService](#)) o aggiorni un servizio esistente ([UpdateService](#)).

Nella tabella seguente vengono descritti 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 rimanenti dopo la fine del processo avrà il processo terminato.	• cpu • memory	
random Le attività vengono posizionate in modo casuale.	Non utilizzato	

tipo	Valori di campo validi	
spread 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 spread 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 di container, 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, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei 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 [Strategie di collocazione di esempio delle attività di Amazon ECS](#).

Strategie di collocazione di esempio delle attività di Amazon ECS

È possibile specificare le strategie di posizionamento delle attività con le seguenti azioni: [CreateServiceUpdateService](#), 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 zone 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

È possibile 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, consultare [Usare le strategie per definire il posizionamento delle attività su 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 [Vincoli di posizionamento di esempio 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 è possibile modificarne il gruppo di attività.

Definisci quali istanze di container utilizza Amazon ECS per le attività

Un vincolo di posizionamento delle attività è una regola relativa a un'istanza di container 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 delle attività o nelle attività utilizzando il parametro `placementConstraint`.

```
"placementConstraints": [  
  {  
    "expression": "The expression that defines the task placement constraints",  
    "type": "The placement constraint type to use"  
  }  
]
```

Nella tabella seguente viene descritta la modalità di utilizzo di questi parametri.

Tipo di vincolo	Può essere specificato quando	
<code>distinctInstance</code> Posizionamento di ciascuna attività attiva in una istanza di container differente. Amazon ECS esamina lo stato desiderato delle attività per il posizionamento delle stesse. 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 posizionata nella stessa istanza nonostante il vincolo di posizionamento <code>distinctInstance</code>. Pertanto, è possibile che vengano visualizzate 2 attività con l'ultimo stato di RUNNING nella stessa istanza.  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.	• Esecuzione di un'attività RunTask • Creando un nuovo servizio CreateService,	

Tipo di vincolo	Può essere specificato quando	
Ciò garantisce che tali carichi di lavoro containerizzati non condividono le interfacce di rete, lo spazio di archiviazione temporaneo, la CPU o la memoria di Fargate con altre attività. Per ulteriori informazioni, vedere Panoramica sulla sicurezza di AWS Fargate.		
<code>memberOf</code> Posizionamento delle attività in istanze di container che soddisfano un'espressione.	• Esecuzione di un'attività RunTask • Creando 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 si utilizza il tipo di vincolo `memberOf`, è possibile creare un'espressione utilizzando il linguaggio di interrogazione del cluster che definisce le istanze di container in cui Amazon ECS può posizionare le attività. L'espressione è un modo per raggruppare le istanze di container in base agli attributi. L'espressione rientra nel parametro `expression` di `placementConstraint`.

Attributi delle istanze di container di 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, consultare [Compatibilità delle versioni dei 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 dei metadati dell'AMI Windows ottimizzata 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`.

Note

Le istanze gestite da Amazon ECS supportano il seguente sottoinsieme di attributi:

- `ecs.subnet-id`
- `ecs.availability-zone`
- `ecs.instance-type`
- `ecs.cpu-architecture`

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, consultare [Aumento delle interfacce di rete delle istanze di container Linux di Amazon ECS](#).

`ecs.outpost-arn`

Se questo attributo esiste, contiene l'Amazon Resource Name (ARN) dell'Outpost. Per ulteriori informazioni, consultare [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, consultare [Cluster Amazon ECS per istanze esterne](#).

Attributi personalizzati

È possibile applicare attributi personalizzati alle istanze di container. Ad esempio, è possibile 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.

Creare 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, consultare [Attributi delle istanze di container di 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, consultare [Esecuzione di un'applicazione come attività Amazon ECS](#) e [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#). 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 delle istanze di container di 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, consultare [Attività Amazon ECS relative al gruppo](#).

Operatore

L'operatore di confronto. Sono supportati i seguenti operatori.

Operatore	Description
==, equals	Uguaglianza stringhe
!=, not_equals	Disuguaglianza stringhe
>, greater_than	Maggiore di
>=, greater_than_equal	Maggiore di o uguale a
<, less_than	Minore di
<=, less_than_equal	Minore di o uguale a
exists	Il soggetto esiste
!exists, not_exists	Il soggetto non esiste
in	Il valore è nell'elenco di argomenti
!in, not_in	Il valore non è nell'elenco di argomenti
=~, matches	Corrispondenza modelli

Operatore	Description
!~, not_matches	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, consultare [java.util.regex.Pattern](#).

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']
```

Vincoli di posizionamento di esempio delle attività di Amazon ECS

Di seguito sono elencati esempi di vincoli del posizionamento delle attività.

In questo esempio viene utilizzata la restrizione `memberOf` per posizionare attività sulle istanze t2. Può essere specificata con le seguenti azioni: [CreateService](#), [UpdateServiceRegisterTaskDefinition](#), e [RunTask](#)

```
"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 stesse. 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 posizionata nella stessa istanza nonostante il vincolo di posizionamento `distinctInstance`. Pertanto, è possibile che vengano visualizzate 2 attività con l'ultimo stato di `RUNNING` nella stessa istanza.

```
"placementConstraints": [  
  {  
    "type": "distinctInstance"  
  }  
]
```

Attività autonome di Amazon ECS

È possibile eseguire l'applicazione come attività quando si dispone di un'applicazione che esegue alcune operazioni e quindi 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

Flusso di lavoro delle attività

Quando si avviano attività di Amazon ECS (autonome o tramite servizi Amazon ECS), un'attività viene creata e inizialmente spostata nello stato `PROVISIONING`. Quando un'attività è nello stato `PROVISIONING`, non esistono né l'attività né i container perché Amazon ECS deve trovare la capacità di calcolo per posizionare 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. È possibile utilizzare i provider di capacità e le strategie dei provider di capacità sia con Fargate che EC2s 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. Infatti EC2, puoi gestire la capacità del cluster registrando EC2 le istanze Amazon nel cluster oppure puoi utilizzare l'auto scaling del cluster per semplificare la gestione della capacità di calcolo. Il dimensionamento automatico del cluster si occupa del dimensionamento dinamico della capacità del cluster, per concentrarsi sulle attività in esecuzione. Amazon ECS determina dove posizionare l'attività in base ai requisiti specificati nella definizione dell'attività, ad esempio CPU e memoria, oltre ai vincoli e alle strategie di posizionamento. Per ulteriori informazioni, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei container](#).

Se si usa un provider di capacità con dimensionamento gestito abilitato, le attività che non possono essere avviate a causa della mancanza di capacità di elaborazione vengono spostate nello stato `PROVISIONING` anziché avere immediatamente un esito negativo. 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 container e quindi avviare i container. Dopo il completamento del provisioning e l'avvio dei container pertinenti, Amazon ECS sposta l'attività nello stato `RUNNING`. Per ulteriori informazioni sugli stati delle attività, consultare [Ciclo di vita dei processi di Amazon ECS](#).

Esecuzione di un'applicazione come attività Amazon ECS

È possibile creare un'attività per un processo singolo utilizzando Console di gestione AWS.

Per creare un'attività autonoma (Console di gestione AWS)

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. La console Amazon ECS consente di creare un'attività autonoma dalla pagina dei dettagli del cluster o dall'elenco di revisione delle definizioni delle attività. Utilizzare la procedura seguente per creare un'attività autonoma in base alla pagina di 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, scegliere Esegui attività.	
La pagina di revisione di definizione dell'attività	a. Nella pagina Definizioni delle attività, scegliere la famiglia di definizioni delle attività per visualizzare le revisioni per quella famiglia. b. Selezionare la revisione da utilizzare. c. Nel menu Distribuisci, scegliere Esegui attività.	

3. Per Cluster esistente, scegliere il cluster.

Scegliere Crea cluster per eseguire l'attività su un nuovo cluster

4. Scegliere come vengono distribuite le attività nell'infrastruttura cluster. In Configurazione di calcolo, scegliere l'opzione desiderata. Per utilizzare una strategia di provider di capacità, è necessario configurare i provider di capacità a livello di cluster.

Se il cluster non è stato configurato per utilizzare un provider di capacità, utilizzare invece un tipo di avvio.

Se si desidera eseguire i carichi di lavoro su istanze gestite da Amazon ECS, è necessario utilizzare l'opzione di strategia del provider di capacità.

Metodo di distribuzione	Fasi	
Strategia del fornitore di capacità	a. Nella sezione Compute options (Opzioni di	

Metodo di distribuzione	Fasi	
	calcolo), seleziona Capacity provider strategy (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 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).	

Metodo di distribuzione	Fasi	
	 Note Per utilizzare un provider di capacità in una strategia , il provider di capacità deve essere associato al cluster.	
Tipo di lancio	- Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). - Per Tipo di avvio, selezionare un tipo di avvio. (Facoltativo) Quando si usa Fargate, per Versione della piattaforma specifica re la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

5. Nella sezione Configurazione implementazione, procedere come segue.

- Per Definizione dell'attività, inserire 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, inserire 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.
 - 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, consultare [Amazon ECS task networking options for Fargate](#) e [Allocate a network interface for an Amazon ECS task](#).

7. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento dell'implementazione, è possibile configurarlo 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 di attività ed eseguire un'attività utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per Tipo di volume EBS, scegliere il tipo di volume EBS da collegare all'attività. b. Per Dimensione (GiB), inserire 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 snapshot. c. Per IOPS, inserisci il numero massimo di input/output operazioni (IOPS) che il volume deve fornire. Questo valore è configurabile solo per i tipi di volume io1, io2 e 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.	

Per configurare questo tipo di volume	Esegui questa operazione	
	e. Per ID Snapshot, scegliere uno snapshot del volume Amazon EBS esistente o inserire l'ARN di uno snapshot se si desidera creare un volume da uno snapshot. È possibile creare un nuovo volume vuoto anche non scegliendo o inserendo un ID di snapshot. f. Se si specifica un ID di snapshot, è possibile specificare una Velocità di inizializzazione del volume (MiB/s). Inserire un valore compreso tra 100 e 300 in MiB/s, che determinerà la velocità di caricamento dei dati dallo snapshot specificato utilizzando ID Snapshot per la creazione del volume. g. Per la Policy di terminazione, deselezionare la casella di spunta se si desidera che il volume configurato per l'allegato all'attività venga preservato o dopo il termine dell'attività. Per impostazione predefinita, i volumi EBS collegati alle attività	

Per configurare questo tipo di volume	Esegui questa operazione	
	vengono eliminati quando l'attività viene terminata. h. Per Tipo di file system, scegliere 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 una snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione della snapshot. Se c'è una mancata corrispondenza del tipo di file system, l'attività non verrà avviata. i. Per il Ruolo di infrastruttura, scegliere un ruolo IAM con le autorizzazioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. È possibile allegare la policy gestita da AmazonECSInfrastru	

Per configurare questo tipo di volume	Esegui questa operazione	
	<code>ctureRole</code> <code>PolicyForVolumes</code> al ruolo oppure utilizzar e la policy come guida per creare e allegare la policy con autorizzazioni che soddisfino le proprie esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessari e, consultare Ruolo IAM dell'infrastruttura Amazon ECS. j. Per Crittografia, scegliere Predefinita se si desidera utilizzare le impostazi oni della crittografia Amazon EBS predefini ta. Se sul tuo account è configurata la crittografia per impostazione predefini ta, il volume verrà crittogra fato con la chiave AWS Key Management Service ()AWS KMS specifica ta nell'impostazione. Se si sceglie Predefinita e la crittografia predefini ta di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a	

Per configurare questo tipo di volume	Esegui questa operazione	
	tua scelta per la crittografia AWS KMS key dei volumi. Se si sceglie Nessuna, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se si crea un volume da uno snapshot crittografato. k. Se hai scelto Personalizzato per la crittografia, devi specificare AWS KMS key quello che desideri utilizzare. Per la Chiave KMS, scegliere una AWS KMS key o inserire un ARN della chiave. Se scegli di crittografare il volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazioni, consultare Data encryption for Amazon EBS volumes. l. (Facoltativo) In Tag, è possibile aggiungere tag	

Per configurare questo tipo di volume	Esegui questa operazione	
	al volume di Amazon EBS propagandoli dalla definizione dell'attività o fornendone di personali zzati. Se si desidera propagare i tag dalla definizione dell'attività, scegliere Definizione dell'attività per Propaga tag da. Se si sceglie Non propagare , o se non si sceglie un valore, i tag non vengono propagati. Se si desidera fornire i propri tag, scegliere Aggiungi tag, e poi fornire la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sulla menzione di volumi Amazon EBS, consultare e Tagging Amazon EBS volumes.	

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, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei container](#).

- Distribuzione con bilanciamento AZ: consente di distribuire le attività tra zone di disponibilità e istanze di container nella zona di disponibilità.
- AZ Balanced BinPack: distribuisce le attività tra zone di disponibilità e tra istanze di container con la minor quantità di memoria disponibile.
- BinPack— Distribuisce le attività in base alla quantità minima disponibile di CPU o memoria.
- One Task Per Host (Un'attività per host): consente di posizionare al massimo un'attività dal servizio in ogni istanza di container.
- Personalizzato: consente di definire una strategia personalizzata 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, consultare [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, consultare [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 strumento di pianificazione 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 di destinazione. 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

È possibile 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 si desidera utilizzare una strategia di provider di capacità, è necessario disporre di 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 Gruppo di pianificazioni, scegliere un gruppo di pianificazioni. 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 Una pianificazione unica richiama una destinazione solo una volta alla data e all'ora specificate.	Per Data e ora, effettua le seguenti operazioni: • 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.	
Pianificazione ricorrente Una pianificazione ricorrent e richiama una destinazione con una frequenza specifica utilizzando un'espressione cron o un'espressione della frequenza.	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	

Ricorrenza	Esegui questa operazione...	
	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 per disattivare l'opzione o scegli una delle finestre temporali predefinite. Ad esempio, se scegli 15 minuti e imposti una pianificazione ricorrente e per il richiamo della 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. Scegliere come vengono distribuite le attività nell'infrastruttura cluster. Espandere le Opzioni di calcolo, poi scegliere una delle seguenti opzioni

Opzione di calcolo	Fasi	
Strategia del fornitore di capacità	a. Scegliere Strategia del provider di capacità. b. Scegli una strategia: • Per utilizzare una strategia del provider di capacità predefinita, scegliere Usa impostazione predefinita del cluster. • Per utilizzare una strategia personalizzata, scegliere Usa personalizzata. Quindi, inserire la Base, il Provider di capacità e il Peso. Infatti EC2, il fornitore di capacità è il gruppo Auto Scaling.	

Opzione di calcolo	Fasi	
Tipo di lancio	a. Scegliere Tipo di avvio. b. Per Tipo di avvio, selezionare un tipo di avvio. c. Quando viene specificato Fargate, per Versione della piattaforma specificare la versione della piattaforma da utilizzare.	

- g. Per le sottoreti, immettere la sottorete in IDs cui eseguire l'attività.
- 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, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei 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 e...	
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) 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 si sceglie questa opzione, inserire l'ARN di una chiave KMS esistente o scegliere 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.

Fasi successive

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 di Amazon ECS

Se non è più necessario mantenere in esecuzione un'attività autonoma, è possibile interrompere l'attività. La console Amazon ECS semplifica l'interruzione di una o più attività.

Non è possibile riavviare un'attività autonoma interrotta.

Se si desidera interrompere un servizio, consultare [Eliminazione di un servizio Amazon ECS utilizzando la console](#).

Per interrompere un'attività autonoma (Console di gestione AWS)

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nel pannello di navigazione scegliere Cluster.

3. Nella pagina Cluster, scegliere il cluster per accedere alla pagina dei dettagli del cluster.
4. Nella pagina dei dettagli del cluster, scegliere la scheda Attività.
5. È possibile filtrare le attività per tipo di avvio utilizzando l'elenco Filtra tipo di avvio.

Attività da interrompere	Fasi	
Una o più	a. Selezionare le attività, poi scegliere Interrompi, Interrompi selezionate. b. Nella pagina di conferma dell'interruzione dell'attività, scegliere Interrompi	
Tutti	 Important Se si sceglie di interrompere tutte le attività utilizzando la console, Amazon ECS interrompe tutte le attività autonome e le attività che fanno parte di un servizio. Pertanto, consigliamo cautela quando si utilizza questa opzione. a. Seleziona Interrompi, Interrompi tutto. b. Nella pagina di conferma dell'interruzione dell'attività, inserire Interrompi tutte le attività, poi scegliere 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.

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 gruppo di destinazione di un bilanciatore del carico. 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. Poi, si verificano le seguenti situazioni.

- Se lo stato di integrità dell'attività di sostituzione è `HEALTHY`, il pianificatore di servizi 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 limitazione del servizio Amazon ECS](#) e [Visualizzare i messaggi relativi agli eventi del servizio Amazon ECS](#).

Opzione di calcolo dell'infrastruttura

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.

Se si desidera eseguire i carichi di lavoro su istanze gestite da Amazon ECS, è necessario utilizzare l'opzione di strategia del provider di capacità.

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 presente una strategia di provider di capacità di default definita, ma sono presenti i provider di capacità di Fargate aggiunti al cluster, è selezionata una strategia di provider di capacità personalizzata che utilizza il provider di capacità `FARGATE`.
- Se nel cluster non è presente una strategia di provider di capacità di default definita, ma sono presenti uno o più provider di capacità del gruppo Auto Scaling aggiunti al cluster, l'opzione Usa personalizzato (avanzate) è selezionata e sarà 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.
- Un tipo di avvio fa sì che Amazon ECS avvii le nostre attività direttamente su Fargate o sulle istanze registrate EC2 nei tuoi cluster.

Se si desidera eseguire i carichi di lavoro su istanze gestite da Amazon ECS, occorre utilizzare l'opzione di strategia del provider di capacità.

Per impostazione predefinita, il servizio viene avviato nelle sottoreti del cluster VPC.

Scalabilità automatica del servizio

Il dimensionamento automatico dei servizi è la capacità di aumentare o diminuire automaticamente il numero desiderato di processi nel servizio Amazon ECS. Amazon ECS utilizza il servizio di Application Auto Scaling per fornire questa funzionalità.

Per ulteriori informazioni, consultare [Scalabilità automatica del servizio Amazon ECS](#).

Bilanciamento del carico nel servizio

I servizi Amazon ECS ospitati su AWS Fargate supportano Application Load Balancer, Network Load Balancer e Gateway Load Balancer. Utilizzare la tabella seguente per ottenere maggiori informazioni sul tipo di bilanciamento del carico da utilizzare.

Tipo di bilanciamento del carico	Utilizzare in questi casi	
Application Load Balancer	Traffico di routing HTTP/HTTPS (o livello 7). Gli Application Load Balancer offrono diverse funzionalità che li rendono particolarmente 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.	

Tipo di bilanciatore del carico	Utilizzare in questi casi	
	• 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	Instradare il traffico TCP o UDP (o livello 4).	
Gateway Load Balancer	Instradare il traffico TCP o UDP (o livello 4). Usare le appliance virtuali, come firewall, sistemi di prevenzione e rilevamento delle intrusioni, e sistemi di ispezione approfondita dei pacchetti.	

Per ulteriori informazioni, consultare [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

Servizi di interconnessione

Se è necessaria un'applicazione per connettersi ad altre applicazioni eseguite nei servizi Amazon ECS, Amazon ECS offre i modi seguenti per farlo senza un bilanciatore del carico:

- **Service Connect:** consente service-to-service comunicazioni con rilevamento automatico utilizzando nomi brevi e porte standard.
- **Rilevamento servizi:** il rilevamento servizi utilizza Route 53 per creare un namespace per il servizio, che ne consente il rilevamento tramite DNS.
- **Amazon VPC Lattice - VPC Lattice** è un servizio di rete di applicazioni completamente gestito per connettere, proteggere e monitorare i tuoi servizi su più account e VPCs. È previsto un costo associato al suo utilizzo.

Per ulteriori informazioni, consultare [Interconnessione dei servizi Amazon ECS](#).

Controller e strategie di implementazione dei servizi Amazon ECS

Prima di implementare il servizio, stabilire le opzioni per l'implementazione e le funzionalità utilizzate dal servizio.

Strategia di pianificazione

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, consultare [Strategia di pianificazione delle repliche](#).
- **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, consultare [Strategia di pianificazione dei daemon](#).

 Note

I processi Fargate non supportano la strategia di pianificazione DAEMON

Strategia di pianificazione delle repliche

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 [Vincoli di posizionamento di esempio 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.

Si consiglia di utilizzare la funzionalità di ribilanciamento del servizio quando si utilizza la strategia REPLICA perché contribuisce a garantire un'elevata disponibilità del servizio.

Strategia di pianificazione dei 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 pianificatore del servizio valuta i vincoli di posizionamento delle attività per le attività in esecuzione e interrompe quelle che non rispondono a tali vincoli. Con questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy 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. Questo significa che l'attività del daemon è la prima da avviare sulle istanze e l'ultima da interrompere dopo aver interrotto tutte le attività delle repliche. La strategia garantisce che le risorse non vengano utilizzate da attività di replica in sospenso e che siano disponibili per le attività del daemon.

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 implementazione, il valore del parametro `maximumPercent` deve essere 100 (specificato in percentuale), che è il valore predefinito utilizzato se non è impostato. Il valore predefinito per il parametro `minimumHealthyPercent` 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 implementazione 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.
- CloudWatch Viene generato un 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 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.

Controller di implementazione

Il controller di implementazione è il meccanismo che determina il modo in cui le attività vengono implementate per il servizio. Le opzioni valide sono:

- ECS

Quando si crea un servizio che utilizza il controller di implementazione ECS, è possibile scegliere tra le seguenti strategie di implementazione:

- ROLLING: quando si crea un servizio che utilizza la strategia di implementazione con aggiornamento in sequenza (ROLLING), il pianificatore del servizio Amazon ECS sostituisce le attività correntemente in esecuzione con le nuove. Il numero di attività che Amazon ECS aggiunge o rimuove dal servizio durante un aggiornamento continuo è controllato dalla configurazione di distribuzione del servizio.

Le distribuzioni di aggiornamento continuo sono più adatte per i seguenti scenari:

- Aggiornamenti graduali ai servizi: è necessario aggiornare il servizio in modo incrementale senza mettere offline l'intero servizio contemporaneamente.
- Requisiti di risorse limitati: è necessario evitare i costi aggiuntivi in termini di risorse derivanti dall'esecuzione simultanea di due ambienti completi (come richiesto dalle implementazioni blu/verdi).
- Tempo di implementazione accettabile: l'applicazione può tollerare un processo di implementazione più lungo, poiché gli aggiornamenti continui sostituiscono le attività una per una.
- Rollback istantaneo non necessario: il servizio può tollerare un processo di rollback che richiede minuti anziché secondi.
- Processo di implementazione semplice: si preferisce un approccio di implementazione semplice senza la complessità della gestione di più ambienti, gruppi di destinazione e listener.
- Nessun requisito di bilanciamento del carico: il servizio non utilizza né richiede un load balancer, Application Load Balancer, Network Load Balancer o Service Connect (necessari per le implementazioni). blue/green

- Applicazioni stateful: l'applicazione mantiene uno stato che rende difficile l'esecuzione di due ambienti paralleli.
- Attenzione ai costi: si desidera ridurre al minimo i costi di implementazione evitando l'esecuzione di ambienti duplicati durante l'implementazione.

Gli aggiornamenti continui sono la strategia di implementazione predefinita per i servizi e forniscono un equilibrio tra sicurezza dell'implementazione ed efficienza delle risorse per molti scenari applicativi comuni.

- BLUE_GREEN: una strategia di implementazione blu/verde (BLUE_GREEN) è una metodologia di rilascio che riduce i tempi di inattività e i rischi eseguendo due ambienti di produzione identici denominati blu e verde. Con le blue/green implementazioni di Amazon ECS, puoi convalidare nuove revisioni dei servizi prima di indirizzare il traffico di produzione verso di esse. Questo approccio offre un modo più sicuro per implementare le modifiche con la possibilità di ripristinarle rapidamente, se necessario.

Le blue/green implementazioni di Amazon ECS sono più adatte per i seguenti scenari:

- Convalida dei servizi: quando è necessario convalidare le nuove revisioni del servizio prima di indirizzare il traffico di produzione verso di esse
- Azzeramento del tempo di inattività: quando il servizio richiede implementazioni senza tempi di inattività
- Rollback istantaneo: quando è necessaria la possibilità di eseguire rapidamente il rollback se vengono rilevati problemi
- Requisito di bilanciamento del carico: quando il servizio utilizza Application Load Balancer, Network Load Balancer o Service Connect
- LINEAR: Una strategia di implementazione lineare (LINEAR) sposta gradualmente il traffico dall'ambiente di produzione corrente a un nuovo ambiente con incrementi percentuali uguali in un periodo di tempo specificato. Con le implementazioni lineari di Amazon ECS, puoi controllare il ritmo dello spostamento del traffico e convalidare nuove revisioni dei servizi con quantità crescenti di traffico di produzione.

Le implementazioni lineari di Amazon ECS sono più adatte per i seguenti scenari:

- Convalida graduale: quando desideri convalidare gradualmente la tua nuova versione del servizio con l'aumento del traffico
- Monitoraggio delle prestazioni: quando hai bisogno di tempo per monitorare metriche e prestazioni durante l'implementazione

- Riduzione al minimo del rischio: quando si desidera ridurre al minimo il rischio esponendo la nuova versione al traffico di produzione in modo incrementale
- Requisito di bilanciatore del carico: quando il servizio utilizza Application Load Balancer, Network Load Balancer o Service Connect
- CANARY: Una strategia di implementazione di Canary (CANARY) sposta prima una piccola percentuale di traffico verso la nuova revisione del servizio, quindi sposta il traffico rimanente tutto in una volta dopo un periodo di tempo specificato. Ciò consente di testare la nuova versione con un sottoinsieme di utenti prima della distribuzione completa.

Le implementazioni di Amazon ECS Canary sono più adatte per i seguenti scenari:

- Test delle funzionalità: quando desideri testare nuove funzionalità con un piccolo sottoinsieme di utenti prima dell'implementazione completa
 - Convalida della produzione: quando è necessario convalidare prestazioni e funzionalità con un traffico di produzione reale
 - Controllo del raggio di esplosione: quando si desidera ridurre al minimo il raggio di esplosione se vengono rilevati problemi nella nuova versione
 - Requisito di bilanciatore del carico: quando il servizio utilizza Application Load Balancer, Network Load Balancer o Service Connect
- Esterno

Utilizzare un controller di implementazione di terza parte.

- Implementazione blu/verde (fornita da) AWS CodeDeploy

CodeDeploy installa una versione aggiornata dell'applicazione come nuovo set di attività sostitutivo e reindirizza il traffico di produzione dal set di attività dell'applicazione originale al set di attività sostitutivo. Il set di attività originale viene terminato una volta completata l'implementazione. Usare questo controller di implementazione per verificare una nuova implementazione di un servizio prima che questo riceva traffico di produzione.

Rilevamento degli errori di implementazione di Amazon ECS

Amazon ECS offre due metodi per rilevare gli errori di implementazione:

- Interruttore automatico di implementazione
- CloudWatch Allarmi

Entrambi i metodi possono essere configurati per ripristinare automaticamente le implementazioni non riuscite all'ultimo stato valido conosciuto.

Considera i seguenti aspetti:

- Entrambi i metodi supportano solo i tipi di distribuzione e blue/green distribuzione degli aggiornamenti in sequenza.
- Quando vengono utilizzati entrambi i metodi, entrambi possono causare un errore di implementazione.
- Il metodo rollback richiede un'implementazione precedente nello stato COMPLETED.
- EventBridge gli eventi vengono generati per le modifiche dello stato di distribuzione.

In che modo l'interruttore di implementazione 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 di 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 l'implementazione non è idonea per un altro rollback fino a quando non raggiunge uno stato COMPLETED. 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, il pianificatore tiene traccia delle attività che non sono state avviate in due fasi.

- Fase 1: il pianificatore monitora le attività per vedere se passano allo stato RUNNING.

- Operazione riuscita: l'implementazione 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 l'implementazione potrebbe passare allo stato FAILED.
- Fase 2: l'implementazione entra in questa fase quando è presente almeno un'attività nello stato RUNNING. L'interruttore automatico verifica i controlli dello stato delle attività nell'implementazione attuale in fase 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 dell'integrità sono stati superati.
- Operazione non riuscita: le attività sostituite a causa di errori nei controlli dell'integrità hanno raggiunto la soglia di errore.

Considera quanto segue quando utilizzi 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, consultare [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, consultare [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 quindi lo utilizza per determinare quando modificare lo stato dell'implementazione 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.

```
Minimum threshold <= 0.5 * desired task count => 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 in presenza di uno o più processi nello stato RUNNING. 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)

Conteggio attività desiderato	Calcolo	Threshold
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 numero di errori impostato su 0. Quando un'attività non riesce a raggiungere lo stato RUNNING, l'interruttore automatico di implementazione aumenta di uno il numero di errori. Quando il numero di errori è uguale a 3, l'implementazione 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=EN
```

Considera quanto segue quando utilizzi il metodo Amazon CloudWatch alarms su un servizio.

- La durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione. Amazon ECS calcola questo periodo di tempo in base alla configurazione degli allarmi associata all'implementazione. Non è possibile 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 `Rollback` 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, consultare [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).
- È possibile configurare questa opzione utilizzando la console di Amazon ECS o la AWS CLI. Per ulteriori informazioni, consultare [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, consultare [Creating wait conditions in a template](#) 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 chiamate `DescribeAlarms`, Amazon ECS potrebbe avere un impatto sulla necessità di interrogare gli allarmi. Ad esempio, se un altro servizio effettua un numero sufficiente di chiamate a `DescribeAlarms` per raggiungere la quota, tale servizio viene limitato e Amazon ECS non sarà in grado di eseguire il polling degli allarmi. Se viene generato un allarme durante il periodo di limitazione, Amazon ECS potrebbe non rilevare 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 implementazione per correggere un errore di implementazione 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](#).

Tempo di incorporamento

Quando utilizzi l'opzione di rollback per le distribuzioni dei tuoi servizi, Amazon ECS attende un ulteriore periodo di tempo dopo la distribuzione della revisione del servizio di destinazione prima di inviare un allarme. CloudWatch Questo è indicato come tempo di incorporamento, che inizia dopo che:

- Tutte le attività per una revisione del servizio di destinazione sono in esecuzione e in buono stato
- Le revisioni del servizio di origine vengono ridotte allo 0%

Il tempo di incorporamento predefinito è inferiore a 5 minuti. L'implementazione del servizio viene contrassegnata come completa dopo la scadenza del tempo di incorporamento.

È possibile configurare il tempo di incorporamento per un'implementazione continua. Quando utilizzi gli CloudWatch allarmi per rilevare un guasto, se modifichi il tempo di cottura e poi decidi di utilizzare l'impostazione predefinita di Amazon ECS, devi impostare manualmente il tempo di cottura.

Hook del ciclo di vita per le implementazioni di servizi Amazon ECS

Quando inizia un'implementazione, passa attraverso le fasi del ciclo di vita. Queste fasi possono trovarsi in stati come IN_PROGRESS o SUCCESSFUL. È possibile utilizzare gli hook del ciclo di vita, che sono funzioni Lambda che Amazon ECS esegue in fasi del ciclo di vita specifiche. Le funzioni possono essere le seguenti:

- Un'API asincrona che convalida il controllo dell'integrità entro 15 minuti.
- Un'API di sondaggio che avvia un altro processo asincrono che valuta il completamento dell'hook del ciclo di vita.

Al termine dell'esecuzione della funzione, essa deve restituire un `hookStatus` affinché l'implementazione continui. Se non viene restituito un `hookStatus` o se la funzione ha esito negativo, l'implementazione viene ripristinata. Di seguito sono riportati i valori `hookStatus`:

- **SUCCEEDED**: l'implementazione continua fino alla fase successiva del ciclo di vita

- **FAILED:** l'implementazione torna all'ultima riuscita.
- **IN_PROGRESS:** Amazon ECS esegue nuovamente la funzione dopo un breve periodo di tempo. Per impostazione predefinita, si tratta di un intervallo di 30 secondi, tuttavia questo valore è personalizzabile restituendo un `callbackDelay` insieme a `hookStatus`.

Nell'esempio seguente viene illustrato come restituire un `hookStatus` con un ritardo di callback personalizzato. In questo esempio, Amazon ECS riprovarebbe questo hook dopo 60 secondi invece dei 30 secondi predefiniti:

```
{
  "hookStatus": "IN_PROGRESS",
  "callbackDelay": 60
}
```

Quando si verifica un rollback, Amazon ECS esegue gli hook del ciclo di vita per le seguenti fasi del ciclo di vita:

- **PRODUCTION_TRAFFIC_SHIFT**
- **TEST_TRAFFIC_SHIFT**

Payload del ciclo di vita

Quando si configurano gli hook del ciclo di vita per le implementazioni di servizi ECS, Amazon ECS richiama questi hook in fasi specifiche del processo di implementazione. Ogni fase del ciclo di vita fornisce un payload JSON con informazioni sullo stato attuale dell'implementazione. Questo documento descrive la struttura del payload per ogni fase del ciclo di vita.

Struttura comune del payload

I payload in tutte le fasi del ciclo di vita includono i seguenti campi comuni:

- **serviceArn:** il nome della risorsa Amazon (ARN) del servizio.
- **targetServiceRevisionArn:** l'ARN della revisione del servizio target in fase di implementazione.
- **testTrafficWeights-** Una mappa delle revisioni dei servizi in base ARNs alle corrispondenti percentuali di peso del traffico di test.

- **productionTrafficWeights**- Una mappa della revisione dei servizi in base ARNs alle corrispondenti percentuali di peso del traffico di produzione.

Payload delle fasi del ciclo di vita

RECONCILE_SERVICE

Questa fase si verifica all'inizio del processo di implementazione quando il servizio viene riconciliato. Di seguito viene riportato un esempio di payload per questa fase del ciclo di vita.

```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
  "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
  "testTrafficWeights": {
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/01275892":
100,
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/78652123": 0
  },
  "productionTrafficWeights": {
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/01275892":
100,
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/78652123": 0
  }
}
```

Aspettative in questa fase:

- Il set di attività principale è su scala 0%

PRE_SCALE_UP

Questa fase avviene prima che le nuove attività vengano aumentate. Di seguito viene riportato un esempio di payload per questa fase del ciclo di vita.

```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
```

```
"targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
"testTrafficWeights": {},
"productionTrafficWeights": {}
}
```

Aspettative in questa fase:

- Le attività di revisione del servizio verde sono su una scala dello 0%

POST_SCALE_UP

Questa fase avviene dopo che le nuove attività sono state aumentate e sono integre. Di seguito viene riportato un esempio di payload per questa fase del ciclo di vita.

```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
  "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
  "testTrafficWeights": {},
  "productionTrafficWeights": {}
}
```

Aspettative in questa fase:

- Le attività di revisione del servizio verde sono su una scala dello 100%
- Le attività di revisione del servizio verde sono integre

TEST_TRAFFIC_SHIFT

Questa fase avviene quando il traffico di test viene spostato sulle attività di revisione del servizio verde.

Di seguito viene riportato un esempio di payload per questa fase del ciclo di vita.

```
{
```

```
"serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
"targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
"testTrafficWeights": {
  "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/01275892":
100,
  "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/78652123": 0
},
"productionTrafficWeights": {}
}
```

Aspettative in questa fase:

- Il traffico di test si sta avviando verso le attività di revisione del servizio verde.

POST_TEST_TRAFFIC_SHIFT

Questa fase avviene dopo che il traffico di test è stato completamente spostato verso le nuove attività.

Di seguito viene riportato un esempio di payload per questa fase del ciclo di vita.

```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
  "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
  "testTrafficWeights": {},
  "productionTrafficWeights": {}
}
```

Aspettative in questa fase:

- Il 100% del traffico di test è stato spostato verso le attività di revisione del servizio verde.

PRODUCTION_TRAFFIC_SHIFT

Questa fase avviene quando il traffico di produzione viene spostato sulle attività di revisione del servizio verde.


```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
  "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
  "testTrafficWeights": {},
  "productionTrafficWeights": {
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/01275892":
100,
    "arn:aws:ecs:us-west-2:1234567890:service-revision/myCluster/myService/78652123": 0
  }
}
```

Aspettative in questa fase:

- Il traffico di produzione si sta spostando verso la revisione del servizio verde.

POST_PRODUCTION_TRAFFIC_SHIFT

Questa fase avviene dopo che il traffico di produzione è stato spostato completamente alle attività di revisione del servizio verde.

```
{
  "serviceArn": "arn:aws:ecs:us-west-2:1234567890:service/myCluster/myService",
  "targetServiceRevisionArn": "arn:aws:ecs:us-west-2:1234567890:service-revision/
myCluster/myService/01275892",
  "testTrafficWeights": {},
  "productionTrafficWeights": {}
}
```

Aspettative in questa fase:

- Il 100% del traffico di produzione è stato spostato verso le attività di revisione del servizio verde.

Categorie della fase del ciclo di vita

Le fasi del ciclo di vita rientrano in due categorie:

1. Fasi di invocazione singole: queste fasi vengono invocate una sola volta durante l'implementazione di un servizio:
 - PRE_SCALE_UP
 - POST_SCALE_UP
 - POST_TEST_TRAFFIC_SHIFT
 - POST_PRODUCTION_TRAFFIC_SHIFT
2. Fasi di invocazione ricorrenti: queste fasi possono essere invocate più volte durante l'implementazione di un servizio, ad esempio quando si verifica un'operazione di rollback:
 - TEST_TRAFFIC_SHIFT
 - PRODUCTION_TRAFFIC_SHIFT

Stato dell'implementazione durante gli hook del ciclo di vita

Mentre gli hook del ciclo di vita sono in esecuzione, lo stato di implementazione sarà `IN_PROGRESS` per tutte le fasi del ciclo di vita.

Stadio del ciclo di vita	Stato dell'implementazione
RECONCILE_SERVICE	IN_PROGRESS
PRE_SCALE_UP	IN_PROGRESS
POST_SCALE_UP	IN_PROGRESS
TEST_TRAFFIC_SHIFT	IN_PROGRESS
POST_TEST_TRAFFIC_SHIFT	IN_PROGRESS
PRODUCTION_TRAFFIC_SHIFT	IN_PROGRESS
POST_PRODUCTION_TRAFFIC_SHIFT	IN_PROGRESS

Interruzione delle implementazioni dei servizi Amazon ECS

È possibile interrompere manualmente una distribuzione quando l'interruttore automatico o gli allarmi non hanno rilevato un'installazione difettosa. CloudWatch Sono disponibili i seguenti tipi di interruzione:

- **Rollback:** questa opzione ripristina l'implementazione del servizio alla revisione precedente del servizio.

È possibile utilizzare questa opzione anche se non è stata configurata l'implementazione del servizio per l'opzione di rollback.

È possibile interrompere un'implementazione che si trova in uno dei seguenti stati. Per ulteriori informazioni sugli stati di implementazione, consultare [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).

- **PENDING:** l'implementazione del servizio passa allo stato `ROLLBACK_REQUESTED`, quindi viene avviata l'operazione di rollback.
- **IN_PROGRESS:** l'implementazione del servizio passa allo stato `ROLLBACK_REQUESTED`, quindi viene avviata l'operazione di rollback.
- **STOP_REQUESTED:** l'implementazione del servizio continua fino all'interruzione.
- **ROLLBACK_REQUESTED:** l'implementazione del servizio continua l'operazione di rollback.
- **ROLLBACK_IN_PROGRESS:** l'implementazione del servizio continua l'operazione di rollback.

Procedura

Prima di iniziare, configurare le autorizzazioni richieste per visualizzare le implementazioni dei servizi. Per ulteriori informazioni, consulta [Autorizzazioni richieste per visualizzare le implementazioni del servizio di 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 Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegliere Implementazioni.

Viene visualizzata la pagina delle implementazioni.

5. In Implementazione in corso, scegliere Esegui il rollback. Quindi, nella finestra di conferma, scegliere Rollback.

AWS CLI

1. Eseguire `list-service-deployments` per recuperare l'ARN di implementazione del servizio.

Sostituiscili *user-input* con i tuoi valori.

```
aws ecs list-service-deployments --cluster cluster-name --service service-name
```

Annotare il `serviceDeploymentArn` per l'implementazione che si desidera 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`. Usare il `serviceDeploymentArn` restituito da `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
```

Fasi successive

Decidere quali modifiche devono essere apportate al servizio, quindi aggiornare il servizio. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).

Distribuisci i servizi Amazon ECS sostituendo le attività

Quando viene creato un servizio che utilizza il tipo di implementazione con aggiornamento in sequenza (ECS), il pianificatore del servizio Amazon ECS sostituisce le attività correntemente in esecuzione con nuove attività. Il numero di attività che Amazon ECS aggiunge o rimuove dal servizio durante un aggiornamento continuo è controllato dalla configurazione di distribuzione del servizio.

Amazon ECS utilizza i seguenti parametri per determinare il numero di attività:

- `minimumHealthyPercent` Rappresenta il limite inferiore al numero di attività che devono essere eseguite e funzionanti per un servizio durante una distribuzione continua o quando un'istanza del contenitore si sta esaurendo, come 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, 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.
- `maximumPercent` Rappresenta il limite massimo al numero di attività che devono essere eseguite per un servizio durante una distribuzione continua o quando un'istanza del contenitore si sta esaurendo, come 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 avviare 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.

Durante una distribuzione continua, quando le attività non funzionano correttamente, Amazon ECS le sostituisce per mantenere il servizio `minimumHealthyPercent` e proteggere la disponibilità. Le attività non corrette vengono sostituite utilizzando la stessa revisione del servizio a cui appartengono. Ciò garantisce che la sostituzione di attività non corrette nella revisione di origine sia indipendente dagli errori delle attività nella revisione di destinazione. Quando l'`maximumPercent` impostazione lo consente, lo scheduler avvia le attività sostitutive prima di interrompere quelle non sane. Se il `maximumPercent` parametro impedisce allo scheduler di avviare prima un'attività sostitutiva, lo scheduler interrompe un'attività non integra alla volta per liberare capacità prima di avviare un'attività sostitutiva.

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 implementazione non valida, verrà inviato un messaggio di evento del servizio. Per ulteriori informazioni, consultare [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 implementazioni in sequenza dispongono di 2 metodi che forniscono un sistema per identificare rapidamente quando un'implementazione di servizi ha avuto esito negativo:

- [the section called “In che modo l'interruttore di implementazione 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 si utilizzano entrambi i metodi, l'implementazione 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 revisione del servizio precedente.

Risoluzione dell'immagine del container

Per impostazione predefinita, Amazon ECS risolve i tag di immagine dei container specificati nella definizione dell'attività in digest delle immagini del container. Se si crea un servizio che esegue e gestisce una singola attività, tale attività viene utilizzata per stabilire digest di immagini per i container inclusi nell'operazione. Se viene creato un servizio che esegue e gestisce più attività, la prima attività

avviata dal pianificatore di servizi durante l'implementazione viene utilizzata per stabilire i digest delle immagini per i container nelle attività.

Se tre o più tentativi di stabilire i digest delle immagini dei container hanno esito negativo, l'implementazione continua senza la risoluzione dei digest delle immagini. Se l'interruttore automatico di implementazione è abilitato, quest'ultima ha esito negativo e viene sottoposta a rollback.

Dopo aver stabilito i digest delle immagini del container, Amazon ECS utilizza i digest per avviare qualsiasi altra attività desiderata e per eventuali futuri aggiornamenti del servizio. Ciò comporta che tutte le attività di un servizio eseguono sempre immagini di container identiche, con conseguente coerenza delle versioni del software.

È possibile configurare questo comportamento per ogni container nell'attività utilizzando il parametro `versionConsistency` nella definizione del container. Per ulteriori informazioni, consultare [versionConsistency](#).

Note

- Le versioni di Amazon ECS Agent precedenti a 1.31.0 non supportano la risoluzione del digest delle immagini. Le versioni Agent da 1.31.0 a 1.69.0 supportano la risoluzione del digest delle immagini solo per le immagini inviate ai repository Amazon ECR. Le versioni Agent 1.70.0 o successive supportano la risoluzione del digest delle immagini per tutte le immagini.
- La versione della piattaforma minima di Fargate Linux per la risoluzione del digest delle immagini è 1.3.0. La versione della piattaforma minima di Fargate Windows per la risoluzione del digest delle immagini 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, specificare i digest delle immagini del container nella definizione dell'attività.
- Se viene creato un servizio con un numero di attività desiderate pari a zero, Amazon ECS non può stabilire i digest dei container finché non si attiva un'altra implementazione del servizio con un numero di attività desiderate maggiore di zero.
- Per stabilire digest di immagini aggiornati, è possibile forzare una nuova implementazione. 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.

- Quando si utilizzano fornitori di EC2 capacità, se la capacità di avvio di un'attività non è sufficiente durante la distribuzione iniziale, la coerenza della versione del software potrebbe fallire. Per garantire la coerenza delle versioni anche in caso di capacità limitata, impostare esplicitamente `versionConsistency`: "enabled" nella configurazione del container di definizione delle attività anziché fare affidamento sul comportamento predefinito. Ciò fa sì che Amazon ECS attenda che la capacità diventi disponibile prima di procedere con l'implementazione.

Best practice per i parametri del servizio di Amazon ECS

Per garantire che non si verifichino tempi di inattività delle applicazioni, il processo di implementazione è il seguente:

1. Avviare i nuovi container di applicazioni mantenendo attivi quelli esistenti.
2. Verificare che i nuovi container siano integri.
3. Arrestare i vecchi container.

A seconda della configurazione di implementazione e della quantità di spazio libero e non riservato nel cluster, potrebbero essere necessari più cicli per completare la procedura per sostituire tutte le vecchie attività con nuove attività.

Vi sono due opzioni di configurazione del servizio che è possibile utilizzare per modificare il numero:

- `minimumHealthyPercent`: 100% (impostazione predefinita)

Il limite inferiore sul numero di attività per cui il servizio deve rimanere nello stato `RUNNING` durante un'implementazione. Si tratta di una percentuale del `desiredCount` arrotondata per eccesso al valore intero più vicino. Questo parametro consente di eseguire l'implementazione senza utilizzare la capacità aggiuntiva del cluster.

- `maximumPercent`: 200% (impostazione predefinita)

Il limite superiore sul numero di attività per il servizio che sono consentite nello stato `RUNNING` o `PENDING` durante un'implementazione. Si tratta di una percentuale del `desiredCount` arrotondata per difetto al valore intero più vicino.

Esempio: opzioni di configurazione predefinite

Prendiamo in considerazione il seguente servizio con sei attività, implementato in un cluster che può ospitare otto attività in totale. Le opzioni di configurazione predefinite del servizio non consentono all'implementazione di scendere al di sotto del 100% delle sei attività desiderate.

Di seguito è riportato il procedimento di implementazione:

1. L'obiettivo è sostituire le sei attività.
2. Il pianificatore 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. Il pianificatore interrompe due delle attività esistenti.

Ora ci sono quattro attività esistenti e due nuove attività.

4. Il pianificatore avvia due nuove attività aggiuntive.

Ora ci sono quattro attività esistenti e quattro nuove attività.

5. Il pianificatore arresta due delle attività esistenti.

Ora ci sono due attività esistenti e quattro attività nuove.

6. Il pianificatore avvia due nuove attività aggiuntive.

Ora ci sono due attività esistenti e sei nuove attività.

7. Il pianificatore arresta 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 bilanciatore del carico potrebbe dover attendere 5 minuti prima che la vecchia attività si interrompa.

Esempio: Modifica **minimumHealthyPercent**

È possibile velocizzare l'implementazione impostando il valore `minimumHealthyPercent` al 50%.

Prendiamo in considerazione il seguente servizio con sei attività, implementato in un cluster che può ospitare otto attività in totale. Di seguito è riportato il procedimento di implementazione:

1. L'obiettivo è sostituire sei attività.
2. Il pianificatore interrompe tre delle attività esistenti.

Sono ancora in esecuzione tre attività esistenti che soddisfano il valore `minimumHealthyPercent`.

3. Il pianificatore avvia cinque nuove attività.

Ci sono tre attività esistenti e cinque nuove attività.

4. Il pianificatore interrompe tre attività esistenti rimanenti.

Ci sono cinque nuove attività

5. Il pianificatore avvia le nuove attività finali.

Ci sono 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, implementato in un cluster che può ospitare dieci attività in totale. Di seguito è riportato il procedimento di implementazione:

1. L'obiettivo è sostituire le attività esistenti.
2. Il pianificatore interrompe tre delle attività esistenti.

Ci sono tre attività esistenti.

3. Il pianificatore avvia sei nuove attività.

Ci sono attività esistenti e sei nuove attività

4. Il pianificatore interrompe le tre attività esistenti.

Ci sono sei nuove attività.

Raccomandazioni

Utilizzare 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%

Creazione di un'implementazione di aggiornamenti continui di Amazon ECS

Creare un servizio per eseguire e mantenere simultaneamente un numero specificato 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.

Decidere 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.

Se si desidera eseguire i carichi di lavoro su istanze gestite da Amazon ECS, è necessario utilizzare l'opzione di strategia del 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.

Se si desidera eseguire i carichi di lavoro su istanze gestite da Amazon ECS, occorre utilizzare l'opzione di strategia del provider di capacità.

- 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, ossia la strategia di posizionamento delle attività predefinita, distribuisce le attività in modo uniforme tra le zone di disponibilità.

Consigliamo di utilizzare il ribilanciamento delle zone di disponibilità per garantire un'elevata disponibilità del servizio. Per ulteriori informazioni, consultare [Bilanciamento di un servizio Amazon ECS tra 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 presente una strategia di provider di capacità di default definita, ma sono presenti i provider di capacità di Fargate aggiunti al cluster, è selezionata una strategia di provider di capacità personalizzata che utilizza il provider di capacità FARGATE.
- Se nel cluster non è presente una strategia di provider di capacità di default definita, ma sono presenti uno o più provider di capacità del gruppo Auto Scaling aggiunti al cluster, l'opzione Usa personalizzato (avanzate) è selezionata e sarà 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 per il rilevamento degli errori di implementazione prevedono l'utilizzo dell'opzione Utilizza l'interruttore di implementazione di Amazon ECS con l'opzione di Rollback in caso di errori.

Per ulteriori informazioni, consultare [In che modo l'interruttore di implementazione Amazon ECS rileva i guasti](#).

- Decidere se si desidera che Amazon ECS aumenti o riduca automaticamente il numero desiderato di attività nel servizio. Per informazioni, consulta [Scalabilità automatica del 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, consultare [Interconnessione dei servizi Amazon ECS](#).
- Quando viene creato un servizio che utilizza l'interruttore automatico Amazon ECS, Amazon ECS crea un'implementazione 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 dei servizi Amazon ECS](#).

Per informazioni su come creare un servizio utilizzando il AWS CLI, consulta [create-servicela](#) sezione Reference.AWS Command Line Interface

Per informazioni su come creare un servizio utilizzando AWS CloudFormation, consulta [AWS::ECS::Service](#) la Guida per l'AWS CloudFormation utente.

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, scegliere il cluster in cui creare il servizio.
4. Nella scheda Servizi, scegliere Crea.

Viene visualizzata la pagina Crea servizio.

5. In Dettagli del servizio, effettuare le seguenti operazioni:
 - a. Per Definizione di processo, inserire la famiglia di definizioni di processi e la revisione da utilizzare.
 - b. In Nome servizio, specificare un nome per il servizio.
6. Per utilizzare ECS Exec per il debug del servizio, selezionare Attiva ECS Exec in Risoluzione dei problemi di configurazione.
7. Nella sezione Configurazione implementazione, procedere come segue.
 - Per Attività desiderate, immettere il numero di attività da avviare e gestire nel servizio.
8. (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 i parametri definiti, completare la seguente procedura.

Per creare un servizio (console Amazon ECS)

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
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, selezionare il pulsante di opzione accanto alla definizione dell'attività. b. Nel menu Implementa, scegliere Crea servizio.	

Viene visualizzata la pagina Crea servizio.

3. In Dettagli del servizio, effettuare le seguenti operazioni:
 - a. Per Definizione di processo, inserire la definizione di processi da utilizzare. Quindi, per Revisione, scegliere la revisione da utilizzare.
 - b. In Nome servizio, specificare un nome per il servizio.
4. Per Cluster esistente, scegliere il cluster.

Scegliere Crea cluster per eseguire l'attività su un nuovo cluster

5. Scegliere come vengono distribuite le attività nell'infrastruttura cluster. In Configurazione di calcolo, scegliere l'opzione.

Opzione di calcolo	Fasi	
Strategia del fornitore 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	

Opzione di calcolo	Fasi	
	definisci la strategia personalizzata specificando Base, Provider di capacità e Peso.  Note Per utilizzare un provider di capacità in una strategia, il provider di capacità deve essere associato al cluster.	
Tipo di lancio	- Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). - Per Tipo di avvio, selezionare un tipo di avvio. (Facoltativo) Quando si usa Fargate, per Versione della piattaforma specificare la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

- Per utilizzare ECS Exec per il debug del servizio, selezionare Attiva ECS Exec in Risoluzione dei problemi di configurazione.
- Nella sezione Configurazione implementazione, procedere 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 è stato 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 Ribilanciamento del servizio delle zone di disponibilità, selezionare Ribilanciamento del servizio della zona di disponibilità.
- d. Per Periodo di tolleranza dei controlli di integrità, inserire il periodo di tempo (in secondi) durante il quale il pianificatore di servizi ignora i controlli dell'integrità dei container, di bilanciamento dei carichi elastici e di VPC Lattice non integri dopo che è stata avviata un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0.
- e. Determina il tipo di implementazione per il servizio. Espandere Opzioni di implementazione, quindi specificare i seguenti parametri.

Il tipo di implementazione	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, consultare Deployment configuration. 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).	

- 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 l'implementazione all'ultimo stato di implementazione completata quando l'interruttore automatico di implementazione imposta l'implementazione su uno stato di errore, selezionare Rollback in caso di errore.

- 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.

- 8. Se la definizione dell'attività utilizza la modalità di rete awsvpc, è possibile specificare una configurazione di rete personalizzata espandendo Rete e quindi effettuare le seguenti operazioni:
 - 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.
 - 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, consultare [Amazon ECS task networking options for Fargate](#) e [Allocate a network interface for an Amazon ECS task](#).

9. (Facoltativo) Per connettere il servizio usando Service Connect, espandere Service Connect, quindi specificare quanto segue:
 - a. Selezionare Attiva Service Connect.
 - b. 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 nel namespace, scegliere 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 nel namespace, scegliere Client e server.
 - c. Per utilizzare un namespace differente da quello del cluster predefinito, per Namespace, scegliere il namespace del servizio. Può trattarsi di uno spazio dei nomi creato separatamente Regione AWS nello stesso spazio dell'utente Account AWS o di uno spazio dei nomi nella stessa regione condiviso con il proprio account utilizzando (). AWS Resource Access Manager AWS RAM Per ulteriori informazioni sugli spazi dei AWS Cloud Map nomi condivisi, consulta Condivisione dello spazio dei nomi [tra AWS Cloud Map](#) account nella Guida per gli sviluppatori.AWS Cloud Map
 - d. (Facoltativo) Specificare una configurazione del log. Selezionare Usa la raccolta di log. L'opzione predefinita invia i log dei contenitori a Logs. CloudWatch 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 del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.

- Flusso di dati Amazon Kinesis: configura il processo per inviare log di container a Kinesis Data Streams. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna 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 log di container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del bucket Amazon S3 valido.
- e. (Facoltativo) Per abilitare i log di accesso, segui questi passaggi:
- i. Espandi la configurazione del registro di accesso. Per Format, scegli JSON o TEXT.
 - ii. Per includere i parametri di interrogazione nei log di accesso, selezionate Includi parametri di interrogazione.
10. (Facoltativo) Per connettere il servizio usando il rilevamento servizi, espandere Rilevamento servizi, ed effettuare le seguenti operazioni:
- a. Selezionare Utilizza il rilevamento servizi.
 - b. Per utilizzare un nuovo namespace, scegliere Crea un nuovo namespace in Configura namespace, fornire quindi un nome e una descrizione del namespace. Per utilizzare un namespace esistente, scegliere Seleziona un namespace esistente, scegliere quindi il namespace da utilizzare.
 - c. Fornire informazioni sul servizio di rilevamento servizi come il nome e la descrizione del servizio.
 - d. Per fare in modo che Amazon ECS esegua controlli periodici dello stato di integrità a livello di container, selezionare Abilita la propagazione dell'integrità delle attività Amazon ECS.
 - e. Per Tipo di record DNS seleziona il tipo di record DNS da creare per il servizio. Il rilevamento servizi Amazon ECS supporta solo record A e SRV, a seconda della modalità di rete specificata dalla definizione di 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 si sceglie A, andare al passaggio successivo. Se si sceglie SRV, specificare la porta sulla quale si trova il servizio oppure un nome di container e una combinazione di porte da associare al record.

Per TTL, inserire il tempo in secondi per cui un set di record viene memorizzato nella cache dai resolver DNS e dai browser Web.

11. (Facoltativo) Per collegare il servizio utilizzando VPC Lattice, espandere VPC Lattice, quindi effettuare le seguenti operazioni:

- a. Selezionare Attiva VPC Lattice
- b. Per Ruolo dell'infrastruttura, scegliere il ruolo dell'infrastruttura.

Se non è stato creato un ruolo, scegliere Crea ruolo dell'infrastruttura.

- c. In Gruppi di destinazione, scegliere il gruppo o i gruppi di destinazione. È necessario scegliere un gruppo di destinazione compreso tra 1 e 5. Scegliere Aggiungi gruppo di destinazione per aggiungere altri gruppi di destinazione. Scegliere il Nome della porta, il Protocollo e la Porta per ogni gruppo di destinazione scelto.

Per rimuovere un gruppo di destinazione, scegliere Rimuovi.

Note

- Se si desidera aggiungere gruppi di destinazione esistenti, è necessario usare la AWS CLI. Per istruzioni su come aggiungere gruppi target utilizzando il AWS CLI, consulta [register-targets](#) nella Guida di riferimento. AWS Command Line Interface
- Sebbene un servizio VPC Lattice possa essere collegato a più gruppi di destinazione, un gruppo di destinazione può essere aggiunto solo a un singolo servizio.

- d. Per completare la configurazione VPC Lattice, includendo i nuovi gruppi di destinazione nell'azione predefinita del listener o nelle regole di un servizio VPC Lattice esistente nella console VPC Lattice. Per ulteriori informazioni, consultare [Listener rules for your VPC Lattice service](#).

12. (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	- Per Seleziona tipo di load balancer, scegli Applicati on Load Balancer. - 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. - Per Load balancer name (Nome load balancer), immetti un nome univoco. - Per Choose container to load balance (Scegli il container per il bilanciamento del carico), scegli il container che ospita il servizio. - 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.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	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 Ritardo di annullamento della registrazione, inserire il numero di secondi in cui il bilanciatore del carico può modificare lo stato di destinazione in 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	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	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 annullamento della registrazione, inserire il numero di secondi in cui il bilanciatore del carico può modificare lo stato di destinazione in UNUSED.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	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 (/).	

13. (Facoltativo) Per configurare il servizio di dimensionamento automatico, espandere Dimensionamento automatico del servizio e quindi specificare i seguenti parametri. Per utilizzare il dimensionamento automatico predittivo, che esamina i dati di caricamento precedenti provenienti dai flussi di traffico, è necessario configurarlo dopo aver creato il servizio. Per ulteriori informazioni, consultare [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).

- Per utilizzare il dimensionamento automatico, selezionare Dimensionamento automatico del servizio.
- Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
- In Numero massimo di processi, specificare il limite superiore del numero di processi che devono essere utilizzati dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
- Scegli il tipo di policy. In Tipo di policy di dimensionamento, scegliere una delle opzioni seguenti.

Per utilizzare questo tipo di policy	Esegui questa operazione	
Monitoraggio degli obiettivi	- In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. - In Nome policy, inserire il nome della policy. - Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. - ECSServiceMediaCPUUtilization: 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 Tempo di raffreddamento di aumento	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	orizzontale, immettere la quantità di tempo che deve passare, in secondi, tra un'attività di aumento (aggiungere attività) e l'inizio di un'altra attività di questo tipo. f. Per Tempo di raffreddamento di riduzione orizzontale, immettere la quantità di tempo che deve passare, in secondi, tra un'attività di riduzione (rimuovere attività) e l'inizio di un'altra attività di questo tipo. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Seleziona Disattiva riduzione orizzontale se si desidera che la politica di scalabilità venga ridimensionata in base all'aumento del traffico, ma non è necessario che	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	venga ampliata quando il traffico diminuisce.	

Per utilizzare questo tipo di policy	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 policy	Esegui questa operazione	
	esegui queste operazioni: • In Operazione, scegliere se aggiungere, rimuovere o impostare un determinato numero desiderato per il servizio. • Se si sceglie di aggiungere o rimuovere attività, immettere nel campo Valore il numero di attività (o la percentuale di attività esistenti) da aggiungere o rimuovere quando l'operazione di dimensionamento viene attivata. 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	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	della regolazione di dimensionamento per fasi. Per impostazione 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. Scegliere Aggiungi nuova operazione di dimensionamento, quindi ripetere i passaggi di Operazioni di dimensionamento. • In Tempo di raffreddamento, inserire la quantità di tempo, espressa in secondi, necessaria per rendere effettiva un'attività di dimensionamento precedent	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	e. Per una policy di aggiunta, si tratta del periodo successivo a un'attività di aumento orizzontale in cui la politica di scalabilità blocca le attività con aumento orizzontale e limita il numero di attività che possono essere aumentate orizzontalmente alla volta. Per una policy di rimozione, questo è il tempo che deve passare tra un'attività di riduzione e l'inizio di un'altra attività di questo tipo.	

14. (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, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei container](#).

- Distribuzione con bilanciamento AZ: consente di distribuire le attività tra zone di disponibilità e 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.
- One Task Per Host (Un'attività per host): consente di posizionare al massimo un'attività dal servizio in ogni istanza di container.

- **Personalizzato:** consente di definire una strategia personalizzata 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.

15. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento dell'implementazione, è possibile configurare il volume espandendo Volume.

Il nome e il tipo di volume vengono configurati durante la creazione di una revisione della definizione di attività e non possono essere modificati quando si crea 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 Tipo di volume EBS, scegliere il tipo di volume EBS da collegare all'attività. - Per Dimensione (GiB), inserire un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume	

Per configurare questo tipo di volume	Esegui questa operazione	
	minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di snapshot. c. Per IOPS, immettere il numero massimo di input/output operazioni (IOPS) che il volume deve fornire. Questo valore è configurabile solo per i tipi di volume <code>io1</code>, <code>io2</code> e <code>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 ID Snapshot, scegliere uno snapshot del volume Amazon EBS esistente o inserire l'ARN di uno snapshot se si desidera creare un volume da uno snapshot. È possibile creare un nuovo volume vuoto anche non scegliendo o inserendo un ID di snapshot.	

Per configurare questo tipo di volume	Esegui questa operazione	
	f. Se si specifica un ID di snapshot, è possibile specificare una Velocità di inizializzazione del volume (MiB/s). Inserire un valore compreso tra 100 e 300 in MiB/s, che determinerà la velocità di caricamento dei dati dallo snapshot specificato utilizzando ID Snapshot per la creazione del volume. g. Per Tipo di file system, scegliere 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 una snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione della snapshot. Se c'è una mancata corrispondenza del tipo di file system, l'attività non verrà avviata.	

Per configurare questo tipo di volume	Esegui questa operazione	
	h. Per il Ruolo di infrastruttura, scegliere un ruolo IAM con le autorizzazioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. È possibile allegare la policy gestita da <code>AmazonECSInfrastructureRolePolicyForVolumes</code> al ruolo, oppure utilizzare la policy come guida per creare e allegare la policy con autorizzazioni che soddisfino le esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessari e, consulta Ruolo IAM dell'infrastruttura Amazon ECS. i. Per Crittografia, scegliere Predefinita se si desidera utilizzare le impostazioni della crittografia Amazon EBS predefinita. Se sul tuo account è configurata la crittografia per impostazione predefinita, il volume verrà crittografato con la chiave AWS	

Per configurare questo tipo di volume	Esegui questa operazione	
	Key Management Service (AWS KMS) specificata nell'impostazione. Se si sceglie Predefinita 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 si sceglie Nessuna, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se si crea un volume da uno snapshot crittografato. j. Se hai scelto Personalizzato per la crittografia, devi specificare AWS KMS key quello che desideri utilizzare. Per la Chiave KMS, scegliere una AWS KMS key o inserire un ARN della chiave. Se scegli di crittografare il volume utilizzando una chiave	

Per configurare questo tipo di volume	Esegui questa operazione	
	simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazioni, consultare Data encryption for Amazon EBS volumes. k. (Facoltativo) In Tag, è possibile aggiungere tag al volume di Amazon EBS propagando i tag dalla definizione dell'attività o del servizio, o fornendo tag personalizzati. Se si desidera propagare i tag dalla definizione dell'attività, scegliere Definizione dell'attività per Propaga tag da. Se si desidera propagare i tag dal servizio, scegliere Servizio per Propagare i tag da. Se si sceglie Non propagare, o se non si sceglie un valore, i tag non vengono propagati. Se si desidera fornire i propri tag, scegliere Aggiungi tag, e poi fornire	

Per configurare questo tipo di volume	Esegui questa operazione	
	la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sulla menzione di volumi Amazon EBS, consultare e Tagging Amazon EBS volumes.	

16. Per utilizzare ECS Exec per il debug del servizio, selezionare Attiva ECS Exec in Risoluzione dei problemi di configurazione.
17. (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).

18. Scegli Create (Crea).

Fasi successive

Di seguito sono riportate le operazioni aggiuntive dopo la creazione di un servizio.

- Configurare il dimensionamento automatico predittivo, che analizza i dati di caricamento precedenti provenienti dai flussi di traffico. Per ulteriori informazioni, consultare [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).
- Tracciare l'implementazione e visualizzare la cronologia dei servizi dell'interruttore automatico di Amazon ECS. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).

Implementazioni Amazon ECS blue/green

Una blue/green distribuzione è una metodologia di rilascio che riduce i tempi di inattività e i rischi eseguendo due ambienti di produzione identici denominati blu e verde. Con le blue/green implementazioni di Amazon ECS, puoi convalidare nuove revisioni dei servizi prima di indirizzare il traffico di produzione verso di esse. Questo approccio offre un modo più sicuro per implementare le modifiche con la possibilità di ripristinarle rapidamente, se necessario.

Vantaggi

Di seguito sono riportati i vantaggi dell'utilizzo delle distribuzioni: blue/green

- Riduce i rischi effettuando test con il traffico di produzione prima di cambiare produzione. È possibile convalidare la nuova implementazione con traffico di test prima di indirizzare il traffico di produzione verso di essa.
- Implementazioni prive di tempo di inattività. L'ambiente di produzione rimane disponibile durante tutto il processo di implementazione, garantendo la disponibilità continua del servizio.
- Rollback semplificato in caso di rilevamento di problemi. In caso di problemi con l'implementazione verde, è possibile tornare rapidamente all'implementazione blu senza interruzioni prolungate del servizio.
- Ambiente di test controllato. L'ambiente verde offre uno spazio isolato per testare nuove funzionalità con modelli di traffico reali prima dell'implementazione completa.
- Processo di implementazione prevedibile. L'approccio strutturato con fasi del ciclo di vita definite rende le implementazioni più coerenti e affidabili.
- Convalida automatizzata tramite hook del ciclo di vita. È possibile implementare test automatizzati in varie fasi dell'implementazione per verificare la funzionalità.

Terminologia

Di seguito sono riportati i termini di blue/green distribuzione di Amazon ECS:

- **Tempo di incorporamento:** la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
- **Implementazione blu:** l'attuale revisione del servizio di produzione che si desidera sostituire.
- **Implementazione verde:** la nuova revisione del servizio che si desidera implementare.
- **Fasi del ciclo di vita:** una serie di eventi nell'operazione di implementazione, ad esempio “dopo lo spostamento del traffico di produzione”.
- **Hook del ciclo di vita:** una funzione Lambda che verifica l'implementazione in una fase specifica del ciclo di vita.
- **Listener:** una risorsa di bilanciamento del carico elastico che controlla le richieste di connessione utilizzando il protocollo e la porta configurata. Le regole definite per un listener determinano il modo in cui Amazon ECS instrada le richieste alle destinazioni registrate.
- **Regola:** una risorsa di bilanciamento del carico elastico associata a un listener. Una regola definisce il modo in cui le richieste vengono instradate e consiste in un'azione, una condizione e una priorità.
- **Gruppo target:** una risorsa Elastic Load Balancing utilizzata per instradare le richieste verso uno o più obiettivi registrati (ad esempio, EC2 istanze). Quando si crea un listener, si specifica un gruppo di destinazione per l'operazione predefinita. Il traffico viene inoltrato al gruppo di destinazione specificato nella regola del listener.
- **Spostamento del traffico:** il processo utilizzato da Amazon ECS per spostare il traffico dall'implementazione blu all'implementazione verde. Per le blue/green implementazioni di Amazon ECS, tutto il traffico viene spostato dal servizio blu al servizio verde contemporaneamente.

Considerazioni

Si tenga in considerazione quanto segue quando si sceglie un tipo di implementazione:

- **Utilizzo delle risorse:** Blue/green le distribuzioni eseguono temporaneamente le revisioni del servizio blu e verde contemporaneamente, il che può raddoppiare l'utilizzo delle risorse durante le distribuzioni.
- **Monitoraggio dell'implementazione:** Blue/green le distribuzioni forniscono informazioni più dettagliate sullo stato dell'implementazione, che consentono di monitorare ogni fase del processo di implementazione.
- **Rollback:** Blue/green le implementazioni semplificano il ripristino alla versione precedente se vengono rilevati problemi, poiché la revisione blu viene mantenuta attiva fino alla scadenza del tempo di cottura.

- Hook del ciclo di vita di Network Load Balancer: se si utilizza un Network Load Balancer per le blue/green distribuzioni, sono disponibili altri 10 minuti per ogni hook del ciclo di vita. Questo perché Amazon ECS garantisce che lo spostamento del traffico sia sicuro.

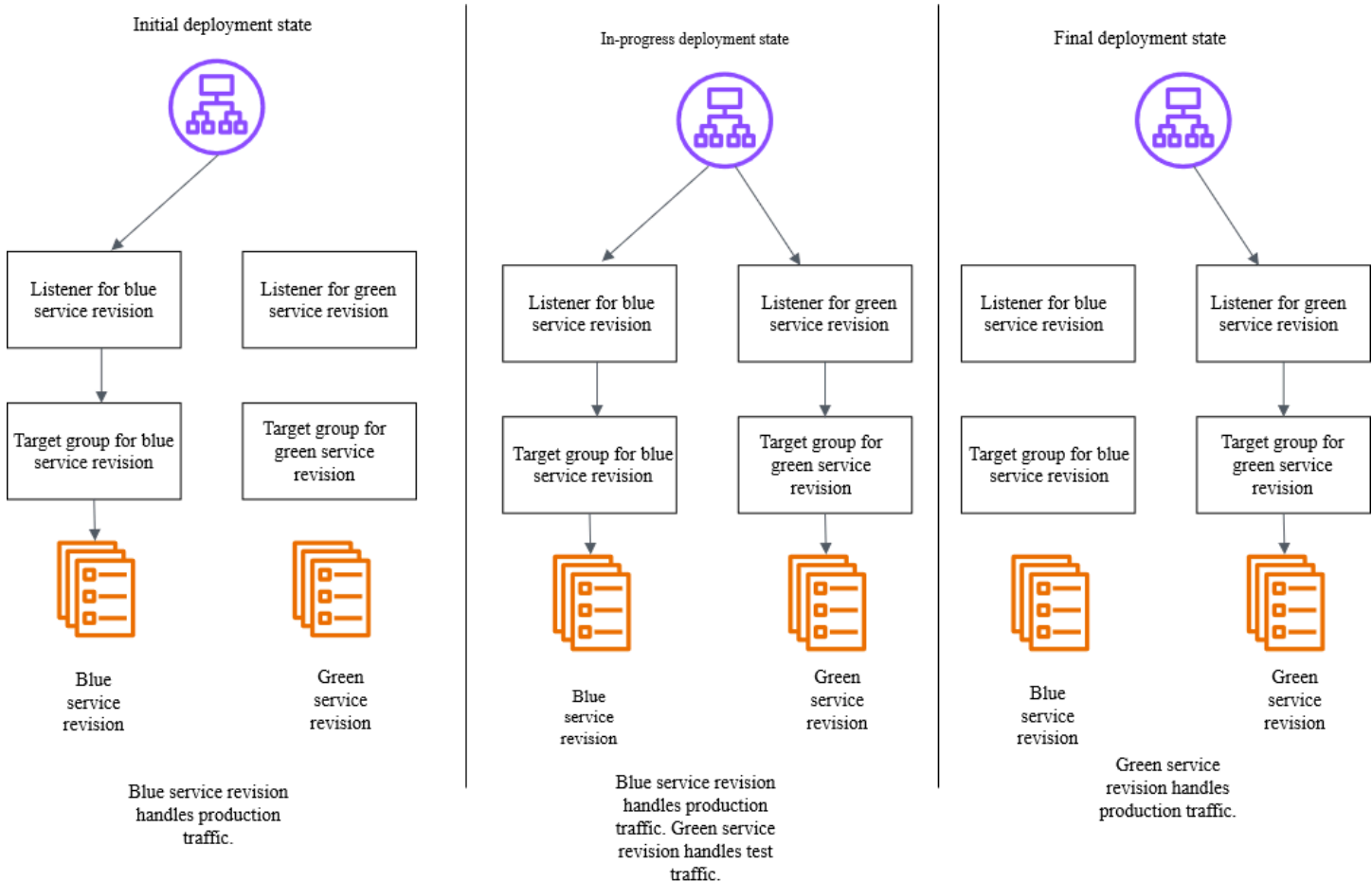
Flusso di lavoro di implementazione dei blue/green servizi Amazon ECS

Il processo di blue/green distribuzione di Amazon ECS segue un approccio strutturato con sei fasi distinte che garantiscono aggiornamenti delle applicazioni sicuri e affidabili. Ogni fase ha uno scopo specifico nella convalida e nella transizione dell'applicazione dalla versione attuale (blu) alla nuova versione (verde).

1. Fase di preparazione: creare l'ambiente verde insieme a quello blu esistente. Ciò include la fornitura di nuove revisioni dei servizi e la preparazione dei gruppi di destinazione.
2. Fase di implementazione: implementazione della nuova revisione del servizio nell'ambiente verde. Amazon ECS avvia nuove attività utilizzando la revisione aggiornata del servizio mentre l'ambiente blu continua a servire il traffico di produzione.
3. Fase di test: convalida dell'ambiente verde utilizzando l'instradamento del traffico di test. L'Application Load Balancer indirizza le richieste di test verso l'ambiente verde mentre il traffico di produzione rimane sul blu.
4. Fase di spostamento del traffico: spostamento del traffico di produzione dal blu al verde in base alla strategia di implementazione configurata. Questa fase include i punti di controllo di monitoraggio e convalida.
5. Fase di monitoraggio: monitoraggio dello stato delle applicazioni, delle metriche delle prestazioni e degli stati di allarme durante il periodo di tempo di incorporamento. Quando vengono rilevati problemi, viene avviata un'operazione di rollback.
6. Fase di completamento: finalizzazione dell'implementazione chiudendo l'ambiente blu o mantenendolo per potenziali scenari di rollback, a seconda della configurazione.

Flusso di lavoro

Il diagramma seguente illustra il flusso di lavoro di blue/green implementazione completo, mostrando l'interazione tra Amazon ECS e Application Load Balancer:



Il flusso di lavoro di implementazione avanzato include i seguenti passaggi dettagliati:

1. Stato iniziale: il servizio blu (produzione attuale) gestisce il 100% del traffico di produzione. L'Application Load Balancer dispone di un unico listener con regole che indirizzano tutte le richieste al gruppo di destinazione blu contenente attività blu integre.
2. Provisioning dell'ambiente verde: Amazon ECS crea nuove attività utilizzando la definizione di quelle aggiornate. Queste attività vengono registrate presso un nuovo gruppo di destinazione verde, ma inizialmente non ricevono traffico.
3. Convalida del controllo dell'integrità: l'Application Load Balancer esegue controlli di integrità sulle attività verdi. Solo quando le attività verdi superano i controlli dell'integrità, l'implementazione passa alla fase successiva.
4. Test di instradamento del traffico: se configurate, le regole del listener di Application Load Balancer indirizzano modelli di traffico specifici (come le richieste con intestazioni di test) verso l'ambiente verde per la convalida, mentre il traffico di produzione rimane sul blu. Questo è controllato dallo stesso listener che gestisce il traffico di produzione, utilizzando regole diverse basate sugli attributi della richiesta.

5. Spostamento del traffico di produzione: in base alla configurazione di implementazione, il traffico passa dal blu al verde. Nelle blue/green implementazioni ECS, si tratta di uno spostamento immediato (all-at-once) in cui il 100% del traffico viene spostato dall'ambiente blu a quello verde. L'Application Load Balancer utilizza un singolo listener con regole del listener che controllano la distribuzione del traffico tra i gruppi di destinazione blu e verdi in base alle ponderazioni.
6. Monitoraggio e convalida: durante l'intero spostamento del traffico, Amazon ECS monitora CloudWatch metriche, stati di allarme e stato di implementazione. I trigger di rollback automatici si attivano se vengono rilevati problemi.
7. Periodo di tempo di incorporamento: la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
8. Terminazione dell'ambiente blu: dopo il corretto spostamento del traffico e la convalida, l'ambiente blu viene terminato per liberare risorse del cluster o mantenuto per consentire una rapida capacità di rollback.
9. Stato finale: l'ambiente verde diventa il nuovo ambiente di produzione, che gestisce il 100% del traffico. L'implementazione è contrassegnata come riuscita.

Fasi del ciclo di vita dell'implementazione

Il processo di blue/green implementazione procede attraverso fasi distinte del ciclo di vita (una serie di eventi durante l'operazione di distribuzione, ad esempio «spostamento del traffico dopo la produzione»), ciascuna con responsabilità e punti di controllo di convalida specifici. La comprensione di queste fasi consente di monitorare l'avanzamento dell'implementazione e risolvere i problemi in modo efficace.

Ogni fase del ciclo di vita può durare fino a 24 ore. Si consiglia di mantenere il valore al di sotto della soglia delle 24 ore. Questo perché i processi asincroni richiedono tempo per attivare gli hook. Il sistema scade, fallisce l'implementazione e quindi avvia un rollback dopo che una fase raggiunge le 24 ore. CloudFormation le distribuzioni hanno restrizioni di timeout aggiuntive. Sebbene il limite delle fasi di 24 ore rimanga in vigore, CloudFormation impone un limite di 36 ore all'intera implementazione. CloudFormation fallisce l'implementazione e quindi avvia un rollback se il processo non viene completato entro 36 ore.

Fasi del ciclo di vita	Description	Usare questa fase per gli hook del ciclo di vita?
RECONCILE_SERVICE	Questa fase si verifica solo quando si avvia una nuova implementazione del servizio con più di 1 revisione del servizio in uno stato ACTIVE.	Sì
PRE_SCALE_UP	La revisione del servizio verde non è stata avviata. La revisione del servizio blu gestisce il 100% del traffico di produzione. Non è previsto alcun traffico di test.	Sì
SCALE_UP	Il momento in cui la revisione del servizio verde aumenta fino al 100% e avvia nuove attività. La revisione del servizio verde non serve alcun traffico a questo punto.	No
POST_SCALE_UP	La revisione del servizio verde è stata avviata. La revisione del servizio blu gestisce il 100% del traffico di produzione. Non è previsto alcun traffico di test.	Sì
TEST_TRAFFIC_SHIFT	Le revisioni del servizio blu e verde sono in esecuzione. La revisione del servizio blu gestisce il 100% del traffico di produzione. La revisione del servizio verde migra dallo 0% al 100% del traffico di test.	Sì
POST_TEST_TRAFFIC_SHIFT	Lo spostamento del traffico di test è completo. La revisione del servizio verde gestisce il 100% del traffico di test.	Sì
PRODUCTION_TRAFFIC_SHIFT	Il traffico di produzione sta passando alla revisione del servizio verde. La revisione	Sì

Fasi del ciclo di vita	Description	Usare questa fase per gli hook del ciclo di vita?
	del servizio verde sta migrando dallo 0% al 100% del traffico di produzione.	
POST_PRODUCTION_TRANSACTION_SHIFT	Lo spostamento del traffico di produzione è completo.	Sì
BAKE_TIME	La durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente.	No
CLEAN_UP	La revisione del servizio blu è stata completamente ridotta a 0 attività in esecuzione. La revisione del servizio verde è ora la revisione del servizio di produzione dopo questa fase.	No

Ogni fase del ciclo di vita include punti di controllo di convalida integrati che devono essere superati prima di passare alla fase successiva. Se una convalida ha esito negativo, l'implementazione può essere ripristinata automaticamente per mantenere la disponibilità e l'affidabilità del servizio.

Quando si utilizza una funzione Lambda, la funzione deve completare il lavoro o ritornare `IN_PROGRESS` entro 15 minuti. È possibile utilizzare il `callbackDelaySeconds` per ritardare la chiamata a Lambda. Per ulteriori informazioni, vedete [la funzione `app.py`](#) in `sample-amazon-ecs-blue-green-deployment-patterns` on GitHub

Risorse necessarie per le implementazioni di Amazon ECS blue/green

Per utilizzare una blue/green distribuzione con trasferimento del traffico gestito, il servizio deve utilizzare una delle seguenti funzionalità:

- Elastic Load Balancing
- Service Connect

Anche i servizi che non utilizzano Service Discovery, Service Connect, VPC Lattice o Elastic Load Balancing possono blue/green utilizzare le implementazioni, ma non ottengono nessuno dei vantaggi dello spostamento del traffico gestito.

L'elenco seguente fornisce una panoramica di alto livello di ciò che è necessario configurare per le distribuzioni di Amazon ECS: blue/green

- Il servizio utilizza Application Load Balancer, Network Load Balancer o Service Connect. Configurare le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 - Network Load Balancer: per ulteriori informazioni, consultare [Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green](#).
 - Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).
- Impostare il controller di implementazione del servizio su ECS.
- Configurare la strategia di implementazione come blue/green nella definizione del servizio.
- Facoltativamente, configurare parametri aggiuntivi come:
 - Tempo di incorporamento per la nuova implementazione
 - CloudWatch allarmi per il rollback automatico
 - Hook del ciclo di vita dell'implementazione per il test (si tratta di funzioni Lambda che vengono eseguite in fasi di implementazione specifiche)

Best practice

Segui queste best practice per blue/green implementazioni Amazon ECS di successo:

- Configurare controlli dell'integrità appropriati che riflettano accuratamente l'integrità dell'applicazione.
- Impostare un tempo di incorporamento che consenta di testare in modo sufficiente l'implementazione verde.
- Implementa CloudWatch allarmi per rilevare automaticamente i problemi e attivare i rollback.
- Utilizzare gli hook del ciclo di vita per eseguire test automatici in ogni fase di implementazione.
- Assicurati che la tua applicazione sia in grado di gestire le revisioni blu e verdi dei servizi in esecuzione simultanea.

- Pianifica una capacità del cluster sufficiente per gestire entrambe le revisioni dei servizi durante la distribuzione.
- Verificare le procedure di rollback prima di implementarle in produzione.

Risorse Application Load Balancer per implementazioni blu/green, lineari e canary

Per utilizzare Application Load Balancers con le blue/green distribuzioni di Amazon ECS, devi configurare risorse specifiche che consentano l'instradamento del traffico tra le revisioni del servizio blu e verde.

Gruppi di destinazione

Per le blue/green implementazioni con Elastic Load Balancing, è necessario creare due gruppi target:

- Un gruppo di destinazione primario per la revisione del servizio blu (traffico di produzione attuale)
- Un gruppo di destinazione alternativo per la revisione del servizio verde (nuova versione)

Entrambi i gruppi di destinazione devono essere configurati con le seguenti impostazioni:

- Tipo di obiettivo: IP (per Fargate o EC2 con modalità di aws vpc rete)
- Protocollo: HTTP (o il protocollo utilizzato dall'applicazione)
- Porta: la porta su cui l'applicazione è in ascolto (in genere 80 per HTTP)
- VPC: lo stesso VPC delle attività di Amazon ECS
- Impostazioni di controllo dell'integrità: configurate per controllare correttamente l'integrità dell'applicazione

Durante una blue/green distribuzione, Amazon ECS registra automaticamente le attività con il gruppo target appropriato in base alla fase di distribuzione.

Example Creazione di gruppi di destinazione per gli Application Load Balancer

I seguenti comandi CLI creano due gruppi target da utilizzare con un Application Load Balancer in una distribuzione: blue/green

```
aws elbv2 create-target-group \  
  --name blue-target-group \  
  --protocol HTTP \  
  --port 80 \  
  --target-type ip
```

```
--vpc-id vpc-abcd1234 \  
--target-type ip \  
--health-check-path / \  
--health-check-protocol HTTP \  
--health-check-interval-seconds 30 \  
--health-check-timeout-seconds 5 \  
--healthy-threshold-count 2 \  
--unhealthy-threshold-count 2  
  
aws elbv2 create-target-group \  
--name green-target-group \  
--protocol HTTP \  
--port 80 \  
--vpc-id vpc-abcd1234 \  
--target-type ip \  
--health-check-path / \  
--health-check-protocol HTTP \  
--health-check-interval-seconds 30 \  
--health-check-timeout-seconds 5 \  
--healthy-threshold-count 2 \  
--unhealthy-threshold-count 2
```

Application Load Balancer

È necessario creare un Application Load Balancer con la seguente configurazione:

- Schema: connesso a Internet o interno, a seconda delle esigenze
- Tipo di indirizzo IP: IPv4
- VPC: lo stesso VPC delle attività di Amazon ECS
- Sottoreti: almeno due sottoreti devono trovarsi in diverse zone di disponibilità
- Gruppi di sicurezza: un gruppo di sicurezza che consente il traffico sulle porte del listener

Il gruppo di sicurezza collegato all'Application Load Balancer deve avere una regola in uscita che consenta il traffico verso il gruppo di sicurezza collegato alle attività di Amazon ECS.

Example Creazione di un Application Load Balancer

Il seguente comando CLI crea un Application Load Balancer da usare in un'implementazione blu/verde:

```
aws elbv2 create-load-balancer \  

```

```
--name my-application-load-balancer \  
--type application \  
--security-groups sg-abcd1234 \  
--subnets subnet-12345678 subnet-87654321
```

Listener e regole

Per le blue/green distribuzioni, è necessario configurare i listener sull'Application Load Balancer:

- Listener di produzione: gestisce il traffico di produzione (in genere sulla porta 80 o 443)
 - Inizialmente inoltra il traffico al gruppo di destinazione primario (revisione blu del servizio)
 - Dopo l'implementazione, inoltra il traffico al gruppo di destinazione alternativo (revisione del servizio verde)
- Listener di test (facoltativo): gestisce il traffico di test per convalidare la revisione del servizio verde prima di spostare il traffico di produzione
 - Può essere configurato su una porta diversa (ad esempio, 8080 o 8443)
 - Inoltra il traffico al gruppo di destinazione alternativo (revisione del servizio verde) durante il test

Durante una blue/green distribuzione, Amazon ECS aggiorna automaticamente le regole del listener per indirizzare il traffico verso il gruppo target appropriato in base alla fase di distribuzione.

Example Creazione di un listener di produzione

Il seguente comando CLI crea un listener di produzione sulla porta 80 che inoltra il traffico al gruppo di destinazione primario (blu):

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/app/my-application-load-balancer/abcdef123456 \  
  --protocol HTTP \  
  --port 80 \  
  --default-actions  
    Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/blue-target-group/abcdef123456
```

Example Creazione di un listener di test

Il seguente comando CLI crea un listener di test sulla porta 8080 che inoltra il traffico al gruppo di destinazione alternativo (verde):

```
aws elbv2 create-listener \
  --load-balancer-arn arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/  

app/my-application-load-balancer/abcdef123456 \
  --protocol HTTP \
  --port 8080 \
  --default-actions  

  Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/  

green-target-group/ghijkl789012
```

Example Creazione di una regola del listener per l'instradamento basato sul percorso

Il seguente comando CLI crea una regola che inoltra il traffico per un percorso specifico al gruppo di destinazione verde per il test:

```
aws elbv2 create-rule \
  --listener-arn arn:aws:elasticloadbalancing:region:123456789012:listener/app/my-  

application-load-balancer/abcdef123456/ghijkl789012 \
  --priority 10 \
  --conditions Field=path-pattern,Values='/test/*' \
  --actions  

  Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/  

green-target-group/ghijkl789012
```

Example Creazione di una regola del listener per l'instradamento basato sull'intestazione

Il seguente comando CLI crea una regola che inoltra il traffico con un'intestazione specifica al gruppo di destinazione verde per il test:

```
aws elbv2 create-rule \
  --listener-arn arn:aws:elasticloadbalancing:region:123456789012:listener/app/my-  

application-load-balancer/abcdef123456/ghijkl789012 \
  --priority 20 \
  --conditions Field=http-header,HTTPHeaderConfig='{Name=X-  

Environment,Values=[test]}' \
  --actions  

  Type=forward,TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/  

green-target-group/ghijkl789012
```

Configurazione del servizio

È necessario disporre delle autorizzazioni per consentire ad Amazon ECS di gestire le risorse del bilanciatore del carico nei cluster. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).

Quando crei o aggiorni un servizio Amazon ECS per blue/green distribuzioni con Elastic Load Balancing, devi specificare la seguente configurazione.

Sostituiscili con i tuoi valori. *user-input*

I componenti chiave di questa configurazione sono:

- `targetGroupArn`: l'ARN del gruppo di destinazione principale (revisione del servizio blu).
- `alternateTargetGroupArn`: l'ARN del gruppo di destinazione alternativo (revisione del servizio verde).
- `productionListenerRule`: l'ARN della regola del listener per il traffico di produzione.
- `roleArn`: l'ARN del ruolo che consente ad Amazon ECS di gestire le risorse del bilanciamento del carico elastico.
- `strategy`: impostato su `BLUE_GREEN` per abilitare implementazioni blu/verdi.
- `bakeTimeInMinutes`: la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
- `TestListenerRule`: l'ARN della regola del listener per il traffico di test. Si tratta di un parametro facoltativo.

```
{
  "loadBalancers": [
    {
      "targetGroupArn":
        "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/primary-target-group/
        abcdef123456",
      "containerName": "container-name",
      "containerPort": 80,
      "advancedConfiguration": {
        "alternateTargetGroupArn":
          "arn:aws:elasticloadbalancing:region:account-id:targetgroup/alternate-target-group/
          ghijkl789012",
      }
    }
  ]
}
```

```

        "productionListenerRule": "arn:aws:elasticloadbalancing:region:account-
id:listener-rule/app/load-balancer-name/abcdef123456/listener/ghijkl789012/rule/
mnopqr345678",
        "roleArn": "arn:aws:iam::123456789012:role/ecs-elb-role"
    }
}
],
"deploymentConfiguration": {
    "strategy": "BLUE_GREEN",
    "maximumPercent": 200,
    "minimumHealthyPercent": 100,
    "bakeTimeInMinutes": 5
}
}

```

Flusso di traffico durante l'implementazione

Durante una blue/green distribuzione con Elastic Load Balancing, il traffico attraversa il sistema nel modo seguente:

1. Stato iniziale: tutto il traffico di produzione viene instradato al gruppo di destinazione principale (revisione blu del servizio).
2. Implementazione della revisione del servizio verde: Amazon ECS implementa le nuove attività e le registra con il gruppo di destinazione alternativo.
3. Traffico di test: se è configurato un listener di test, il traffico di test viene instradato al gruppo di destinazione alternativo per convalidare la revisione del servizio verde.
4. Spostamento del traffico di produzione: Amazon ECS aggiorna la regola del listener di produzione per instradare il traffico verso il gruppo di destinazione alternativo (revisione del servizio verde).
5. Tempo di incorporamento: la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
6. Completamento: dopo una corretta implementazione, la revisione del servizio blu viene interrotta.

Se vengono rilevati problemi durante l'implementazione, Amazon ECS può eseguire automaticamente il rollback instradando il traffico verso il gruppo di destinazione principale (revisione blu del servizio).

Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green

Per utilizzare un Network Load Balancer con blue/green distribuzioni Amazon ECS, devi configurare risorse specifiche che consentano il routing del traffico tra le revisioni del servizio blu e verde. Questa sezione spiega i componenti richiesti e la loro configurazione.

Quando la configurazione include un Network Load Balancer, Amazon ECS aggiunge un ritardo di 10 minuti alle seguenti fasi del ciclo di vita:

- PRE_SCALE_UP
- TEST_TRAFFIC_SHIFT
- PRODUCTION_TRAFFIC_SHIFT

Questo ritardo è dovuto a problemi di temporizzazione del Network Load Balancer che possono causare una mancata corrispondenza tra i pesi del traffico configurati e l'effettivo instradamento del traffico nel piano dati.

Gruppi di destinazione

Per le blue/green implementazioni con un Network Load Balancer, è necessario creare due gruppi target:

- Un gruppo di destinazione primario per la revisione del servizio blu (traffico di produzione attuale)
- Un gruppo di destinazione alternativo per la revisione del servizio verde (nuova revisione del servizio)

Entrambi i gruppi di destinazione devono essere configurati con le seguenti impostazioni:

- Tipo di obiettivo: `ip` (per Fargate o EC2 con modalità di `awsvpc` rete)
- Protocollo: `TCP` (o il protocollo utilizzato dall'applicazione)
- Porta: la porta su cui l'applicazione è in ascolto (in genere `80` per HTTP)
- VPC: lo stesso VPC delle attività di Amazon ECS
- Impostazioni di controllo dell'integrità: configurate per controllare correttamente l'integrità dell'applicazione

Per i controlli dell'integrità del protocollo TCP, il Network Load Balancer stabilisce una connessione TCP con la destinazione. Se la connessione ha esito positivo, la destinazione viene considerata integra.

Per i controlli di HTTP/HTTPS integrità, il Network Load Balancer invia una HTTP/HTTPS richiesta al bersaglio e verifica la risposta.

Durante una blue/green distribuzione, Amazon ECS registra automaticamente le attività con il gruppo target appropriato in base alla fase di distribuzione.

Example Creazione di gruppi di destinazione per un Network Load Balancer

I seguenti comandi AWS CLI creano due gruppi target da utilizzare con un Network Load Balancer in una distribuzione: blue/green

```
aws elbv2 create-target-group \  
  --name blue-target-group \  
  --protocol TCP \  
  --port 80 \  
  --vpc-id vpc-abcd1234 \  
  --target-type ip \  
  --health-check-protocol TCP  
  
aws elbv2 create-target-group \  
  --name green-target-group \  
  --protocol TCP \  
  --port 80 \  
  --vpc-id vpc-abcd1234 \  
  --target-type ip \  
  --health-check-protocol TCP
```

Network Load Balancer

È necessario creare un Network Load Balancer con la seguente configurazione:

- Schema: connesso a Internet o interno, a seconda delle esigenze
- Tipo di indirizzo IP: IPv4
- VPC: lo stesso VPC delle attività di Amazon ECS
- Sottoreti: almeno due sottoreti devono trovarsi in diverse zone di disponibilità

A differenza degli Application Load Balancer, i Network Load Balancer operano a livello di trasporto (Livello 4) e non utilizzano gruppi di sicurezza. Invece, è necessario assicurarsi che i gruppi di

sicurezza associati alle attività di Amazon ECS consentano il traffico dal Network Load Balancer sulle porte del listener.

Example Creazione di un Network Load Balancer

Il seguente comando AWS CLI crea un Network Load Balancer da utilizzare in una distribuzione: blue/green

```
aws elbv2 create-load-balancer \  
  --name my-network-load-balancer \  
  --type network \  
  --subnets subnet-12345678 subnet-87654321
```

Considerazioni sull'utilizzo di NLB con le distribuzioni blue/green

Quando utilizzi un Network Load Balancer per le blue/green distribuzioni, considera quanto segue:

- Funzionamento di livello 4: i Network Load Balancer operano a livello di trasporto (livello 4) e non ispezionano il contenuto a livello di applicazione (livello 7). Ciò significa che non è possibile utilizzare intestazioni o percorsi HTTP per le decisioni di instradamento.
- Controlli dell'integrità: i controlli dell'integrità di Network Load Balancer sono limitati ai protocolli TCP, HTTP o HTTPS. Per i controlli dell'integrità di TCP, il Network Load Balancer verifica solo che la connessione possa essere stabilita.
- Conservazione della connessione: i Network Load Balancer conservano l'indirizzo IP di origine del client, che può essere utile per scopi di sicurezza e registrazione.
- Indirizzi IP statici: i Network Load Balancer forniscono indirizzi IP statici per ogni sottorete, che possono essere utili per la creazione di whitelist o quando i client devono connettersi a un indirizzo IP fisso.
- Traffico di test: poiché i Network Load Balancer non supportano l'instradamento basato sul contenuto, il traffico di test deve essere inviato a una porta diversa rispetto al traffico di produzione.

Listener e regole

Per le blue/green implementazioni con un Network Load Balancer, è necessario configurare i listener:

- Listener di produzione: gestisce il traffico di produzione (in genere sulla porta 80 o 443)
 - Inizialmente inoltra il traffico al gruppo di destinazione primario (revisione blu del servizio)

- Dopo l'implementazione, inoltra il traffico al gruppo di destinazione alternativo (revisione del servizio verde)
- Listener di test (obbligatorio): gestisce il traffico di test per convalidare la revisione del servizio verde prima di spostare il traffico di produzione
- Può essere configurato su una porta diversa (ad es., 8080 o 8443)
- Inoltra il traffico al gruppo di destinazione alternativo (revisione del servizio verde) durante il test

A differenza degli Application Load Balancer, i Network Load Balancer non supportano regole di instradamento basate sul contenuto. Il traffico viene invece instradato in base alla porta e al protocollo del listener.

I seguenti comandi AWS CLI creano listener di produzione e test per un Network Load Balancer:

Sostituiscili *user-input* con i tuoi valori.

```
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/net/my-network-lb/1234567890123456 \  
  --protocol TCP \  
  --port 80 \  
  --default-actions Type=forward,  
  TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/blue-target-group/1234567890123456  
  
aws elbv2 create-listener \  
  --load-balancer-arn arn:aws:elasticloadbalancing:region:123456789012:loadbalancer/net/my-network-lb/1234567890123456 \  
  --protocol TCP \  
  --port 8080 \  
  --default-actions Type=forward,  
  TargetGroupArn=arn:aws:elasticloadbalancing:region:123456789012:targetgroup/green-target-group/1234567890123456
```

Configurazione del servizio

È necessario disporre delle autorizzazioni per consentire ad Amazon ECS di gestire le risorse del bilanciatore del carico nei cluster. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).

Quando crei o aggiorni un servizio Amazon ECS per le blue/green distribuzioni con un Network Load Balancer, devi specificare la seguente configurazione:

Sostituiscili con i tuoi valori. *user-input*

I componenti chiave di questa configurazione sono:

- `targetGroupArn`: l'ARN del gruppo di destinazione principale (revisione del servizio blu)
- `alternateTargetGroupArn`: l'ARN del gruppo di destinazione alternativo (revisione del servizio verde)
- `productionListenerRule`: l'ARN del listener per il traffico di produzione.
- `testListenerRule`: (facoltativo) l'ARN del listener per il traffico di test
- `roleArn`: l'ARN del ruolo che consente ad Amazon ECS di gestire le risorse di Network Load Balancer.
- `strategy`: impostato per `BLUE_GREEN` abilitare le blue/green distribuzioni
- `bakeTimeInMinutes`: la durata dell'attesa dopo l'implementazione della revisione del servizio verde prima di spostare il traffico di produzione

```
{
  "loadBalancers": [
    {
      "targetGroupArn":
        "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/blue-target-
        group/1234567890123456",
      "containerName": "container-name",
      "containerPort": 80,
      "advancedConfiguration": {
        "alternateTargetGroupArn":
          "arn:aws:elasticloadbalancing:region:123456789012:targetgroup/green-target-
          group/1234567890123456",
        "productionListenerRule":
          "arn:aws:elasticloadbalancing:region:123456789012:listener/net/my-network-
          lb/1234567890123456/1234567890123456",
        "testListenerRule":
          "arn:aws:elasticloadbalancing:region:123456789012:listener/net/my-network-
          lb/1234567890123456/2345678901234567",
        "roleArn": "arn:aws:iam::123456789012:role/ecs-nlb-role"
      }
    }
  ]
}
```

```
    ],  
    "deploymentConfiguration": {  
        "strategy": "BLUE_GREEN",  
        "maximumPercent": 200,  
        "minimumHealthyPercent": 100,  
        "bakeTimeInMinutes": 5  
    }  
}
```

Flusso di traffico durante l'implementazione

Durante un' blue/green implementazione con un Network Load Balancer, il traffico attraversa il sistema nel modo seguente:

1. Stato iniziale: tutto il traffico di produzione viene instradato al gruppo di destinazione principale (revisione blu del servizio).
2. Implementazione della revisione del servizio verde: Amazon ECS implementa le nuove attività e le registra con il gruppo di destinazione alternativo.
3. Traffico di test: se è configurato un listener di test, il traffico di test viene instradato al gruppo di destinazione alternativo per convalidare la revisione del servizio verde.
4. Spostamento del traffico di produzione: Amazon ECS aggiorna il listener di produzione per instradare il traffico verso il gruppo di destinazione alternativo (revisione del servizio verde).
5. Tempo di incorporamento: la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
6. Completamento: dopo una corretta implementazione, la revisione del servizio blu viene interrotta.

Se vengono rilevati problemi durante l'implementazione, Amazon ECS può eseguire automaticamente il rollback instradando il traffico verso il gruppo di destinazione principale (revisione blu del servizio).

Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary

Quando si utilizza Service Connect con blue/green le distribuzioni, è necessario configurare componenti specifici per consentire il corretto instradamento del traffico tra le revisioni blu e verde del servizio. Questa sezione spiega i componenti richiesti e la loro configurazione.

Panoramica dell'architettura

Service Connect sviluppa funzionalità di rilevamento servizi e mesh di servizi tramite un proxy sidecar gestito che viene inserito automaticamente nelle attività di Amazon ECS. Questi proxy gestiscono le decisioni di routing, i nuovi tentativi e la raccolta delle metriche, fornendo al contempo il backend del registro dei servizi. AWS Cloud Map Quando si distribuisce un servizio con Service Connect abilitato, il servizio si registra e i servizi client lo scoprono tramite il namespace. AWS Cloud Map

In un'implementazione standard di Service Connect, i servizi client si connettono ai nomi di servizio logici e il proxy sidecar gestisce l'instradamento verso le effettive istanze di servizio. Con le blue/green implementazioni, questo modello viene esteso per includere il routing del traffico di test attraverso la configurazione. `testTrafficRules`

Durante una blue/green distribuzione, i seguenti componenti chiave interagiscono tra loro:

- Proxy di Service Connect: tutto il traffico tra i servizi passa attraverso il proxy di Service Connect, che prende decisioni di instradamento in base alla configurazione.
- AWS Cloud Map Registrazione: sia la distribuzione blu che quella verde si registrano con AWS Cloud Map, ma la distribuzione verde viene inizialmente registrata come endpoint «di test».
- Test dell'instradamento del traffico: la `testTrafficRules` nella configurazione di Service Connect determina come identificare e instradare il traffico di test verso l'implementazione verde. Ciò viene ottenuto tramite l'instradamento basato sulle intestazioni, in cui intestazioni HTTP specifiche nelle richieste indirizzano il traffico verso la revisione di test. Per impostazione predefinita, Service Connect riconosce l'intestazione `x-amzn-ecs-blue-green-test` per i protocolli basati su HTTP quando non vengono specificate regole personalizzate.
- Configurazione del client: tutti i client nel namespace ricevono automaticamente sia i percorsi di produzione che quelli di test, ma solo le richieste che soddisfano le regole di test andranno all'implementazione verde.

Ciò che rende questo approccio efficace è la capacità di gestire la complessità del rilevamento servizi durante le transizioni. Man mano che il traffico si sposta dall'implementazione blu a quella verde, tutti i meccanismi di connettività e rilevamento si aggiornano automaticamente. Non è necessario aggiornare i record DNS, riconfigurare i bilanciatori del carico o implementare le modifiche al rilevamento servizi in modo separato, poiché la mesh di servizi gestisce tutto.

Instradamento e test del traffico

Service Connect offre funzionalità avanzate di routing del traffico per le blue/green implementazioni, tra cui il routing basato su header e la configurazione degli alias client per scenari di test.

Regole di intestazione del traffico di test

Durante le blue/green distribuzioni, è possibile configurare le regole dell'intestazione del traffico di test per indirizzare richieste specifiche alla (nuova) revisione verde del servizio a scopo di test. Ciò consente di convalidare la nuova versione con traffico controllato prima di completare l'implementazione.

Service Connect utilizza l'instradamento basato sulle intestazioni per identificare il traffico di test. Per impostazione predefinita, Service Connect riconosce l'intestazione `x-amzn-ecs-blue-green-test` per i protocolli basati su HTTP quando non vengono specificate regole personalizzate. Quando questa intestazione è presente in una richiesta, il proxy di Service Connect instraderà automaticamente la richiesta nell'implementazione verde per il test.

Le regole di intestazione del traffico di test ti consentono di:

- Indirizzare le richieste con intestazioni specifiche alla revisione del servizio verde
- Testare nuove funzionalità con un sottoinsieme di traffico
- Convalidare il comportamento del servizio prima della conversione completa del traffico
- Implementare strategie di test canary
- Eseguire test di integrazione in un ambiente simile a quello di produzione

Il meccanismo di instradamento basato sulle intestazioni si integra perfettamente con l'architettura applicativa esistente. I servizi client non devono necessariamente essere consapevoli del processo di blue/green distribuzione: includono semplicemente le intestazioni appropriate quando inviano le richieste di test e il proxy Service Connect gestisce automaticamente la logica di routing.

Per ulteriori informazioni sulla configurazione delle regole dell'intestazione del traffico di test, consulta [ServiceConnectTestTrafficHeaderRules](#) la pagina Amazon Elastic Container Service API Reference.

Regole di corrispondenza delle intestazioni

Le regole di header matching definiscono i criteri per il routing del traffico di test durante le distribuzioni. blue/green È possibile configurare più condizioni di corrispondenza per controllare con precisione quali richieste vengono indirizzate alla revisione del servizio verde.

La corrispondenza delle intestazioni supporta:

- Corrispondenza esatta del valore dell'intestazione
- Verifica della presenza dell'intestazione
- Corrispondenza basata su modelli
- Combinazioni multiple di intestazioni

I casi d'uso di esempio includono il routing delle richieste con stringhe di utenti agenti specifiche, versioni API o flag di funzionalità al servizio verde per i test.

Per ulteriori informazioni sulla configurazione dell'header matching, consulta

[ServiceConnectTestTrafficHeaderMatchRules](#) Amazon Elastic Container Service API Reference.

Alias client per le distribuzioni blue/green

Gli alias client forniscono endpoint DNS stabili per i servizi durante le distribuzioni. blue/green Consentono un instradamento fluido del traffico tra le revisioni dei servizi blu e verdi senza richiedere alle applicazioni client di modificare gli endpoint di connessione.

Durante una blue/green distribuzione, gli alias client:

- Mantengono coerenti i nomi DNS per le connessioni client
- Abilitano lo spostamento automatico del traffico tra le revisioni del servizio
- Supportano strategie di migrazione graduale del traffico
- Forniscono funzionalità di rollback reindirizzando il traffico alla revisione blu

È possibile configurare più alias client per porte o protocolli diversi, consentendo ad architetture di servizio complesse di mantenere la connettività durante le implementazioni.

Per ulteriori informazioni sulla configurazione degli alias del client, consulta il riferimento

[ServiceConnectClientAlias](#) all'API di Amazon Elastic Container Service.

Best practice per l'instradamento del traffico

Quando si implementa il routing del traffico per le blue/green distribuzioni con Service Connect, è necessario prendere in considerazione le seguenti best practice:

- Inizia con i test basati sulle intestazioni: utilizza le regole di intestazione del traffico di test per convalidare il servizio verde con traffico controllato prima di cambiare tutto il traffico.

- Configurare i controlli dell'integrità: assicurarsi che sia i servizi blu che quelli verdi abbiano la configurazione dei controlli di integrità appropriati per evitare di instradare il traffico verso istanze non integre.
- Monitora le metriche del servizio: tieni traccia degli indicatori chiave di prestazione per entrambe le revisioni del servizio durante l'implementazione per identificare tempestivamente i problemi.
- Pianificare la strategia di rollback: configurare gli alias dei client e le regole di instradamento per consentire il ripristino rapido del servizio in caso di problemi.
- Verifica la logica di corrispondenza delle intestazioni: convalida le regole di corrispondenza delle intestazioni in un ambiente non di produzione prima di applicarle alle implementazioni di produzione.

Flusso di lavoro di blue/green implementazione di Service Connect

Comprendere come Service Connect gestisce il processo blue/green di distribuzione aiuta a implementare e risolvere i problemi delle distribuzioni in modo efficace. Il seguente flusso di lavoro mostra come interagiscono i diversi componenti durante ogni fase dell'implementazione.

Fasi di implementazione

L' blue/green implementazione di Service Connect procede attraverso diverse fasi distinte:

1. Stato iniziale: il servizio blu gestisce il 100% del traffico di produzione. Tutti i servizi client nel namespace si connettono al servizio blu tramite il nome del servizio logico configurato in Service Connect.
2. Registrazione Green Service: quando la distribuzione verde viene avviata, viene registrata AWS Cloud Map come endpoint di «test». Il proxy di Service Connect nei servizi client riceve automaticamente le configurazioni di instradamento di produzione e test.
3. Test di instradamento del traffico: le richieste contenenti le intestazioni del traffico di test (come `x-amzn-ecs-blue-green-test`) vengono instradate automaticamente al servizio verde dal proxy di Service Connect. Il traffico di produzione continua a fluire verso il servizio blu.
4. Preparazione dello spostamento del traffico: dopo l'esito positivo dei test, il processo di implementazione prepara lo spostamento del traffico di produzione. Sia i servizi blu che quelli verdi rimangono registrati e integri.
5. Spostamento del traffico di produzione: la configurazione di Service Connect si aggiorna per instradare il traffico di produzione verso il servizio verde. Ciò avviene automaticamente senza richiedere aggiornamenti del servizio client o modifiche al DNS.

6. Periodo di tempo di incorporamento: la durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
7. Annullamento della registrazione al servizio blu: dopo aver completato con successo lo spostamento e la convalida del traffico, il servizio blu viene annullato e terminato, completando l'implementazione. AWS Cloud Map

Comportamento del proxy di Service Connect

Il proxy Service Connect svolge un ruolo cruciale nella gestione del traffico durante le blue/green implementazioni. La comprensione del suo comportamento aiuta a progettare strategie di test e implementazione efficaci.

Comportamenti chiave del proxy durante le implementazioni: blue/green

- Rilevamento automatico del percorso: il proxy rileva automaticamente sia i percorsi di produzione che quelli di test AWS Cloud Map senza richiedere riavvii delle applicazioni o modifiche alla configurazione.
- Instradamento basato sulle intestazioni: il proxy esamina le intestazioni di richiesta in entrata e instrada il traffico alla revisione del servizio appropriata in base alle regole del traffico di test configurate.
- Integrazione del controllo dell'integrità: il proxy instrada il traffico solo verso istanze di servizio integre, escludendo automaticamente le attività non integre dal pool di instradamento.
- Nuovo tentativo e utilizzo di interruttori: il proxy offre funzionalità integrate di logica di tentativi e interruzione, migliorando la resilienza durante le implementazioni.
- Raccolta di metriche: il proxy raccoglie metriche dettagliate per i servizi blu e verdi, consentendo un monitoraggio completo durante le implementazioni.

Aggiornamenti del rilevamento servizi

Uno dei principali vantaggi dell'utilizzo di Service Connect per le blue/green implementazioni è la gestione automatica degli aggiornamenti di rilevamento dei servizi. blue/green Le implementazioni tradizionali richiedono spesso aggiornamenti DNS complessi o la riconfigurazione del load balancer, ma Service Connect gestisce queste modifiche in modo trasparente.

Durante un'implementazione, Service Connect gestisce:

- Aggiornamenti del namespace: il namespace di Service Connect include automaticamente gli endpoint di servizio blu e verdi, con regole di instradamento appropriate.
- Configurazione del client: tutti i servizi client nel namespace ricevono automaticamente informazioni di instradamento aggiornate senza richiedere riavvii o nuove implementazioni.
- Transizione graduale: gli aggiornamenti di rilevamento servizi avvengono in modo graduale e sicuro, garantendo l'assenza di interruzioni delle richieste in corso.
- Supporto per il rollback: se è necessario un rollback, Service Connect può ripristinare rapidamente le configurazioni di rilevamento servizi per instradare nuovamente il traffico verso il servizio blu.

Creazione di una distribuzione Amazon ECS blue/green

Utilizzando le blue/green distribuzioni di Amazon ECS, puoi apportare e testare modifiche ai servizi prima di implementarle in un ambiente di produzione.

Prerequisiti

Esegui le seguenti operazioni prima di iniziare una distribuzione. blue/green

1. Configurazione delle autorizzazioni appropriate.
 - Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).
 - Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#)
2. blue/green Le implementazioni di Amazon ECS richiedono che il servizio utilizzi una delle seguenti funzionalità: Configura le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 - Network Load Balancer: per ulteriori informazioni, consultare [Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green](#).
 - Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).
3. Creare una regola per instradare il traffico verso la revisione del servizio verde. Per ulteriori informazioni, consultare [Listener rules](#) nella Guida per l'utente di Network Load Balancer.
4. Creazione di un gruppo di destinazione per la revisione del servizio verde. Quando si usa la modalità di rete aws_vpc per le attività, il tipo di destinazione deve essere ip. Per informazioni

sui gruppi di destinazione, consultare [Target groups](#) nella Guida per l'utente di Network Load Balancer.

5. Decidere se eseguire le funzioni Lambda per gli eventi del ciclo di vita.

- Prima dell'aumento verticale
- Dopo l'aumento verticale
- Spostamento del traffico di test
- Dopo lo spostamento del traffico di test
- Spostamento del traffico di produzione
- Dopo lo spostamento del traffico di produzione

Creare funzioni Lambda per ogni evento del ciclo di vita. Per ulteriori informazioni, consultare [Create a Lambda function with the console](#) nella Guida per gli sviluppatori di AWS Lambda .

Procedura

Puoi utilizzare la console o AWS CLI creare un blue/green servizio Amazon ECS.

Console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
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. Si apre la pagina dei dettagli del cluster. b. Nella scheda Servizi, scegli Crea.	
Definizione di attività	a. Nella pagina Definizioni di attività, selezionare la definizione di attività.	

Per avviare un servizio da	Fasi	
	b. Dal menu Implementa, scegliere Crea servizio.	

Si apre la pagina Crea servizio.

3. In Dettagli del servizio, effettuare le seguenti operazioni:
 - a. Per Famiglia di definizione di attività, scegliere la definizione di attività da utilizzare. Poi, per Revisione della definizione di attività, inserire la revisione da utilizzare.
 - b. In Nome servizio, specificare un nome per il servizio.
4. Per eseguire il servizio in un cluster esistente, per Cluster esistente, scegliere il cluster. Per eseguire il servizio in un nuovo cluster, scegliere Crea cluster
5. Scegliere come vengono distribuite le attività nell'infrastruttura cluster. In Configurazione di calcolo, scegliere l'opzione.

Opzione di calcolo	Fasi	
Strategia del fornitore 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 personali	

Opzione di calcolo	Fasi	
	zzata, 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 capacità deve essere associato al cluster.	

Opzione di calcolo	Fasi	
Tipo di lancio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Tipo di avvio, selezionare un tipo di avvio. c. (Facoltativo) Quando viene specificato Fargate, per Versione della piattaforma, specificare la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

6. Nella sezione Configurazione implementazione, procedere come segue.
 - a. Per Tipo di servizio, scegliere Replica.
 - b. Per Attività desiderate, immettere il numero di attività da avviare e gestire nel servizio.
 - c. Per consentire ad Amazon ECS di monitorare la distribuzione delle attività tra le zone di disponibilità e ridistribuirle in caso di squilibrio, in Ribilanciamento del servizio delle zone di disponibilità, selezionare Ribilanciamento del servizio della zona di disponibilità.
 - d. Per Periodo di tolleranza dei controlli di integrità, inserire il periodo di tempo (in secondi) durante il quale il pianificatore di servizi ignora i controlli dell'integrità dei container, di bilanciamento dei carichi elastici e di VPC Lattice non integri dopo che è stata avviata un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0.

7.
 - a. Per Tempo di incorporamento, inserire il numero di minuti in cui entrambe le revisioni del servizio blu e verde verranno eseguite contemporaneamente prima che la revisione blu venga terminata. Ciò consente di avere tempo per la verifica e il test.
 - b. (Facoltativo) Eseguire le funzioni Lambda in fasi specifiche dell'implementazione. In Hook del ciclo di vita di implementazione, selezionare le fasi per eseguire gli hook del ciclo di vita.

Per aggiungere un hook del ciclo di vita:

- a. Scegliere Aggiungi.
 - b. Per Funzione Lambda, immettere il nome della funzione o ARN.
 - c. Per Ruolo, selezionare il ruolo IAM che ha il permesso di invocare la funzione Lambda.
 - d. Per Fasi del ciclo di vita, selezionare le fasi in cui deve essere eseguita la funzione Lambda.
8. 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 l'implementazione all'ultimo stato di implementazione completata quando l'interruttore automatico di implementazione imposta l'implementazione su uno stato di errore, selezionare Rollback in caso di errore.

- 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.

9. (Facoltativo) Per connettere il servizio usando Service Connect, espandere Service Connect, quindi specificare quanto segue:
 - a. Selezionare Attiva Service Connect.

- b. 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 nel namespace, scegliere 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 nel namespace, scegliere Client e server.
- c. Per utilizzare un namespace differente da quello del cluster predefinito, per Namespace, scegliere il namespace del servizio. Può trattarsi di uno spazio dei nomi creato separatamente Regione AWS nello stesso spazio dell'utente Account AWS o di uno spazio dei nomi nella stessa regione condiviso con il proprio account utilizzando (). AWS Resource Access Manager AWS RAM Per ulteriori informazioni sugli spazi dei AWS Cloud Map nomi condivisi, consulta [Condivisione dello spazio dei nomi tra AWS Cloud Map](#) account nella Guida per gli sviluppatori. AWS Cloud Map
- d. (Facoltativo) Configura le regole di test traffic header per le distribuzioni. blue/green In Instradamento del traffico di test, specificare quanto segue:
 - i. Selezionare Abilita le regole di intestazione del traffico di test per instradare richieste specifiche alla revisione del servizio verde durante il test.
 - ii. Per Regole di corrispondenza di intestazione, configurare i criteri per l'instradamento del traffico di test:
 - Nome dell'intestazione: inserire il nome dell'intestazione HTTP per la corrispondenza (ad esempio, X-Test-Version o User-Agent).
 - Tipo di corrispondenza: scegliere i criteri di corrispondenza:
 - Corrispondenza esatta: richieste di instradamento in cui il valore dell'intestazione corrisponde esattamente al valore specificato
 - Intestazione presente: richieste di instradamento che contengono l'intestazione specificata, indipendentemente dal valore
 - Corrispondenza di schema: richieste di instradamento in cui il valore dell'intestazione corrisponde a uno schema specificato
 - Valore dell'intestazione (se si utilizza la corrispondenza esatta o la corrispondenza di schema): inserire il valore o lo schema a cui far corrispondere l'intestazione.

È possibile aggiungere più regole di corrispondenza dell'intestazione per creare una logica di instradamento complessa. Le richieste che corrispondono a una qualsiasi delle regole configurate verranno instradate alla revisione del servizio verde per essere testate.

- iii. Scegliere Aggiungi regola di intestazione per configurare condizioni aggiuntive di corrispondenza dell'intestazione.

 Note

Le regole di intestazione del traffico di test consentono di convalidare nuove funzionalità con traffico controllato prima di portare a termine l'implementazione completa. Ciò consente di testare la revisione del servizio verde con richieste specifiche (come quelle provenienti da strumenti di test interni o utenti beta) mantenendo al contempo il normale flusso di traffico verso la revisione del servizio blu.

- e. (Facoltativo) Specificare una configurazione del log. Selezionare Usa la raccolta di log. L'opzione predefinita invia i log dei contenitori a Logs. CloudWatch 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 del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
- Flusso di dati Amazon Kinesis: configura il processo per inviare log di container a Kinesis Data Streams. Vengono fornite le opzioni di driver di log predefinite che inviano

i log a un flusso di consegna 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 log di container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del bucket Amazon S3 valido.

10. (Facoltativo) Configurare il Bilanciamento del carico per l'implementazione blu/verde.

Tipo di bilanciamento del carico elastico	Fasi	
Application Load Balancer	- Per Tipo di bilanciatore del carico, scegliere Application Load Balancer. - 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. - Per Container, scegliere il container che ospita il servizio. - Per Nome del bilanciatore del carico, inserire un nome univoco. - Per Listener, specifica una porta e un protocollo per l'Application Load Balancer su cui ascoltare	

Tipo di bilanciamento del carico elastico	Fasi	
	le richieste di connessione. Di default, il load balancer sarà configurato per utilizzare la porta 80 e HTTP. • Per Regola di produzione, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di produzione (blu). • Per Regola di test, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di test (verde). f. Per Gruppo di destinazione, configurare quanto segue: • Per Nome del gruppo di destinazione, specificare un nome e un protocollo per il gruppo di destinazione a cui l'Application Load	

Tipo di bilanciamento del carico elastico	Fasi	
	Balancer instraderà le richieste. • Per Protocollo, scegliere il 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. • Per Ritardo di annullamento della registrazione, inserire il numero di secondi in cui il bilanciatore del carico può modificare lo stato di destinazione in UNUSED. Il valore predefinito è 300 secondi. • 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 periodica	

Tipo di bilanciamento del carico elastico	Fasi	
	mente le richieste per verificare l'integrità della connessione tra l'Application Load Balancer e il container. L'opzione predefinita è la directory root (/). • Per Nome del gruppo alternativo, inserire il nome del gruppo di destinazione per la revisione del servizio di test (verde).	

Tipo di bilanciamento del carico elastico	Fasi	
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 Listener di produzione, scegliere la Porta di listener di produzione e il Protocollo di listener di produzione. Questo è il listener per il traffico di revisione del servizio di produzione (blu). e. Per Listener di test, scegliere la Porta di listener di test e il Protocollo di listener di test. Questo è il listener per il traffico di revisione del servizio di test (verde).	

Tipo di bilanciamento del carico elastico	Fasi	
	f. Per Gruppo di destinazione, configurare quanto segue: • Per Nome del gruppo di destinazione, specificare un nome e un protocollo per il gruppo di destinazione a cui il Network Load Balancer instraderà le richieste. • Per Protocollo, scegliere il 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. • Per Ritardo di annullamento della registrazione, inserire il numero di secondi in cui il bilanciatore del carico può modificare lo stato di destinazione in UNUSED. Il	

Tipo di bilanciamento del carico elastico	Fasi	
	valore predefinito è 300 secondi. • 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 Nome del gruppo alternativo, inserire il nome del gruppo di destinazione per la revisione del servizio di test (verde).	

11. (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).

12. Scegli Create (Crea).

AWS CLI

1. Crea un archivio denominato `service-definition.json` con i seguenti contenuti.

Sostituiscili *user-input* con i tuoi valori.

```
{
  "serviceName": "myBlueGreenService",
  "cluster": "arn:aws:ecs:us-west-2:123456789012:cluster/sample-fargate-
cluster",
  "taskDefinition": "sample-fargate:1",
  "desiredCount": 5,
  "launchType": "FARGATE",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "subnets": [
        "subnet-09ce6e74c116a2299",
        "subnet-00bb3bd7a73526788",
        "subnet-0048a611aaec65477"
      ],
      "securityGroups": [
        "sg-09d45005497daa123"
      ],
      "assignPublicIp": "ENABLED"
    }
  },
  "deploymentController": {
    "type": "ECS"
  },
  "deploymentConfiguration": {
    "strategy": "BLUE_GREEN",
    "maximumPercent": 200,
```

```

    "minimumHealthyPercent": 100,
    "bakeTimeInMinutes": 2,
    "alarms": {
      "alarmNames": [
        "myAlarm"
      ],
      "rollback": true,
      "enable": true
    },
    "lifecycleHooks": [
      {
        "hookTargetArn": "arn:aws:lambda:us-  
west-2:7123456789012:function:checkExample",
        "roleArn": "arn:aws:iam::123456789012:role/ECSLifecycleHookInvoke",
        "lifecycleStages": [
          "PRE_SCALE_UP"
        ]
      }
    ],
    "loadBalancers": [
      {
        "targetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/blue-target-group/54402ff563af1197",
        "containerName": "fargate-app",
        "containerPort": 80,
        "advancedConfiguration": {
          "alternateTargetGroupArn": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:targetgroup/green-target-group/cad10a56f5843199",
          "productionListenerRule": "arn:aws:elasticloadbalancing:us-  
west-2:123456789012:listener-rule/app/my-blue-green-  
demo/32e0e4f946c3c05b/9cfa8c482e204f7d/831dbaf72edb911",
          "roleArn": "arn:aws:iam::123456789012:role/LoadBalancerManagementforECS"
        }
      }
    ]
  }
}

```

2. Esegui create-service.

user-input Sostituiscili con i tuoi valori.

```
aws ecs create-service --cli-input-json file://service-definition.json
```

In alternativa, puoi utilizzare il seguente esempio che crea un servizio di blue/green distribuzione con una configurazione di bilanciamento del carico:

```
aws ecs create-service \
  --cluster "arn:aws:ecs:us-west-2:123456789012:cluster/MyCluster" \
  --service-name "blue-green-example-service" \
  --task-definition "nginxServer:1" \
  --launch-type "FARGATE" \
  --network-configuration
  "awsvpcConfiguration={subnets=[subnet-12345,subnet-67890,subnet-abcdef,subnet-
  fedcba],securityGroups=[sg-12345],assignPublicIp=ENABLED}" \
  --desired-count 3 \
  --deployment-controller "type=ECS" \
  --deployment-configuration
  "strategy=BLUE_GREEN,maximumPercent=200,minimumHealthyPercent=100,bakeTimeInMinutes=0" \
  --load-balancers "targetGroupArn=arn:aws:elasticloadbalancing:us-
  west-2:123456789012:targetgroup/MyBGtg1/
  abcdef1234567890,containerName=nginx,containerPort=80,advancedConfiguration={alternateTa
  west-2:123456789012:targetgroup/
  MyBGtg2/0987654321fedcba,productionListenerRule=arn:aws:elasticloadbalancing:us-
  west-2:123456789012:listener-rule/app/
  MyLB/1234567890abcdef/1234567890abcdef,roleArn=arn:aws:iam::123456789012:role/
  ELBManagementRole}"
```

Fasi successive

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).
- Monitora il processo di distribuzione per assicurarti che segua lo blue/green schema:
 - La revisione del servizio verde viene creata e aumentata verticalmente
 - Il traffico di test viene instradato alla revisione verde (se configurata)
 - Il traffico di produzione viene spostato alla revisione del servizio verde
 - Dopo il tempo di incorporamento, la revisione blu viene interrotta

Risoluzione dei problemi relativi alle implementazioni di Amazon ECS blue/green

Di seguito vengono fornite soluzioni per problemi comuni che potresti riscontrare durante l'utilizzo di blue/green distribuzioni con Amazon ECS. Blue/green gli errori di distribuzione possono verificarsi durante le seguenti fasi:

- Percorso sincrono: errori che appaiono immediatamente in risposta a chiamate API `CreateService` o `UpdateService`.
- Percorso asincrono: errori che appaiono nel campo `statusReason` di `DescribeServiceDeployments` e causano un rollback dell'implementazione

Tip

Puoi utilizzarlo [Server MCP Amazon ECS](#) con gli assistenti AI per monitorare le implementazioni e risolvere i problemi di implementazione utilizzando il linguaggio naturale.

Problemi di configurazione del bilanciatore del carico

La configurazione del bilanciamento del carico è un componente fondamentale delle blue/green distribuzioni in Amazon ECS. La corretta configurazione delle regole dei listener, dei gruppi di destinazione e dei tipi di bilanciatore del carico è essenziale per implementazioni di successo. Questa sezione tratta i problemi comuni di configurazione del load balancer che possono causare il fallimento delle distribuzioni. blue/green

Quando si risolvono i problemi relativi al bilanciatore del carico, è importante comprendere la relazione tra le regole dei listener e i gruppi di destinazione. In una distribuzione: blue/green

- La regola del listener di produzione indirizza il traffico verso la revisione del servizio attualmente attiva (blu)
- La regola del listener di test può essere utilizzata per convalidare la nuova revisione del servizio (verde) prima di spostare il traffico di produzione
- I gruppi di destinazione vengono utilizzati per registrare le istanze di container di ogni revisione del servizio
- Durante l'implementazione, il traffico viene gradualmente spostato dalla revisione del servizio blu alla revisione del servizio verde modificando i pesi dei gruppi di destinazione nelle regole del listener

Errori di configurazione delle regole del listener

I seguenti problemi riguardano la configurazione errata delle regole del listener per le blue/green distribuzioni.

Utilizzo di un ARN del listener di Application Load Balancer anziché di uno della regola del listener

Messaggio di errore: `productionListenerRule` has an invalid ARN format. Must be RuleArn for ALB or ListenerArn for NLB. Got: `arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-alb/abc123/def456`

Soluzione: quando si utilizza un Application Load Balancer, è necessario specificare un ARN della regola del listener per `productionListenerRule` e `testListenerRule`, non un ARN del listener. Per i Network Load Balancer, è necessario utilizzare l'ARN del listener.

Per informazioni su come trova l'ARN del listener, consultare [Listener per Application Load Balancer](#) nella Guida per l'utente di Application Load Balancer. L'ARN per una regola ha il formato *`arn:aws:elasticloadbalancing:region:account-id:listener-rule/app/...`*

Utilizzo della stessa regola per i listener di test e di produzione

Messaggio di errore: The following rules cannot be used as both production and test listener rules: `arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-alb/abc123/def456/ghi789`

Soluzione: è necessario utilizzare regole di listener diverse per il traffico di produzione e quello di test. Creazione di una regola separata del listener per il traffico di test che instrada verso il gruppo di destinazione del test.

Regola del listener di test mancante per un Network Load Balancer

Messaggio di errore: `TestListenerRule` is required for NLB with `arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/nlb-targetgroup/abc123`

Soluzione: quando si utilizza un Network Load Balancer, è necessario specificare sia `productionListenerRule` che `testListenerRule`. Aggiungere un `testListenerRule` con un ARN del listener valido alla configurazione. Per ulteriori informazioni, consultare [Create a listener for your Network Load Balancer](#) nella Guida per l'utente di Network Load Balancer

Gruppo di destinazione non associato alle regole del listener

Messaggio di errore: `Service deployment rolled back because of invalid networking configuration: Target group arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/myAlternateTG/abc123 is not associated with either productionListenerRule or testListenerRule.`

Soluzione: sia il gruppo di destinazione principale che il gruppo di destinazione alternativo devono essere associati alla regola del listener di produzione o alla regola del listener di test. Aggiornare la configurazione del bilanciatore del carico per assicurarsi che entrambi i gruppi di destinazione siano associati correttamente alle regole del listener.

Regola del listener di test mancante con un Application Load Balancer

Messaggio di errore: `For Application LoadBalancer, testListenerRule is required when productionListenerRule is not associated with both targetGroup and alternateTargetGroup`

Soluzione: quando si utilizza un Application Load Balancer, se entrambi i gruppi di destinazione non sono associati alla regola del listener di produzione, è necessario specificare una regola del listener di test. Aggiungere una `testListenerRule` alla configurazione e assicurarsi che entrambi i gruppi di destinazione siano associati alla regola del listener di produzione o di test. Per ulteriori informazioni, consultare [Listeners for your Application Load Balancers](#) nella Guida per l'utente di Application Load Balancer.

Errori di configurazione dei gruppi di destinazione

I seguenti problemi si riferiscono alla configurazione errata del gruppo target per le distribuzioni. blue/green

Gruppi di destinazione multipli con traffico nella regola del listener

Messaggio di errore: `Service deployment rolled back because of invalid networking configuration. productionListenerRule arn:aws:elasticloadbalancing:us-west-2:123456789012:listener-rule/app/my-alb/abc123/def456/ghi789 should have exactly one target group serving traffic but found 2 target groups which are serving traffic`

Soluzione: prima di iniziare una blue/green distribuzione, assicurati che solo un gruppo target riceva traffico (con un peso diverso da zero) secondo la regola del listener. Aggiornare la

configurazione della regola del listener per impostare il peso su zero per qualsiasi gruppo di destinazione che non dovrebbe ricevere traffico.

Duplicare i gruppi di destinazione tra le voci del bilanciatore del carico

Messaggio di errore: Duplicate targetGroupArn found:
arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/myecs-targetgroup/abc123

Soluzione: ogni ARN del gruppo di destinazione deve essere univoco per tutte le voci del bilanciatore del carico nella definizione del servizio. Rivedere la configurazione e assicurarsi di utilizzare gruppi di destinazione diversi per ogni voce del bilanciatore del carico.

Gruppo di destinazione imprevisto nella regola del listener di produzione

Messaggio di errore: Service deployment rolled back because of invalid networking configuration. Production listener rule is forwarding traffic to unexpected target group arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/random-nlb-tg/abc123. Expected traffic to be forwarded to either targetGroupArn: arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/nlb-targetgroup/def456 or alternateTargetGroupArn: arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/nlb-tg-alternate/ghi789

Soluzione: la regola del listener di produzione inoltra il traffico a un gruppo di destinazione non specificato nella definizione del servizio. Assicurarsi che la regola del listener sia configurata per inoltrare il traffico solo ai gruppi di destinazione specificati nella definizione del servizio.

Per ulteriori informazioni, consultare [forward actions](#) nella Guida per l'utente per Application Load Balancer.

Errori di configurazione del tipo di bilanciatore del carico

I seguenti problemi riguardano una configurazione errata del tipo di sistema di bilanciamento del carico per le distribuzioni. blue/green

Combinare configurazioni di Classic Load Balancer, Application Load Balancer o Network Load Balancer.

Messaggio di errore: All loadBalancers must be strictly either ELBv1 (defining loadBalancerName) or ELBv2 (defining targetGroupArn)

 Note

I Classic Load Balancer sono la generazione precedente di bilanciatori del carico dal bilanciamento del carico elastico. Consigliamo di eseguire la migrazione a un bilanciatore del carico di generazione attuale. Per ulteriori informazioni, consultare [Migrate your Classic Load Balancer](#).

Soluzione:. Utilizzare tutti Classic Load Balancer o tutti Application Load Balancer e Network Load Balancer.

Per Application Load Balancer e Network Load Balancer, specificare solo il campo targetGroupArn.

Utilizzo della configurazione avanzata con un Classic Load Balancer

Messaggio di errore: advancedConfiguration field is not allowed with ELBv1 loadBalancers

Soluzione: la configurazione avanzata per le blue/green implementazioni è supportata solo con Application Load Balancer e Network Load Balancer. Se si utilizza un Classic Load Balancer (specificato con loadBalancerName), non è possibile utilizzare il campo advancedConfiguration. Passare a un Application Load Balancer o rimuovere il campo advancedConfiguration.

Configurazione avanzata incoerente tra i bilanciatori del carico

Messaggio di errore: Either all or none of the provided loadBalancers must have advancedConfiguration defined

Soluzione: se si utilizzano più bilanciatori del carico, è necessario definire advancedConfiguration per tutti o nessuno. Aggiornare la configurazione per garantire la coerenza tra tutte le voci del bilanciatore del carico.

Manca blue/green la configurazione avanzata con distribuzione

Messaggio di errore: `advancedConfiguration field is required for all loadBalancers when using a non-ROLLING deployment strategy`

Soluzione: quando si utilizza una strategia di blue/green distribuzione con Application Load Balancers, è necessario specificare il `advancedConfiguration` campo per tutte le voci relative al load balancer. Aggiungere la necessaria `advancedConfiguration` alla configurazione del bilanciatore del carico.

Problemi a livello di autorizzazioni

I seguenti problemi riguardano le autorizzazioni insufficienti per le distribuzioni. blue/green

Policy di attendibilità mancante sul ruolo dell'infrastruttura

Messaggio di errore: `Service deployment rolled back because of invalid networking configuration. ECS was unable to manage the ELB resources due to missing permissions on ECS Infrastructure Role 'arn:aws:iam::123456789012:role/Admin'.`

Soluzione: il ruolo IAM specificato per la gestione delle risorse del bilanciatore del carico non dispone della policy di attendibilità corretta. Aggiornare la policy di attendibilità del ruolo per consentire al servizio di assumere tale ruolo. La policy di attendibilità deve includere:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Autorizzazioni di lettura mancanti per il ruolo di bilanciatore del carico

Messaggio di errore: service myService failed to describe target health on target-group myTargetGroup with (error User: arn:aws:sts::123456789012:assumed-role/myELBRole/ecs-service-scheduler is not authorized to perform: elasticloadbalancing:DescribeTargetHealth because no identity-based policy allows the elasticloadbalancing:DescribeTargetHealth action)

Soluzione: il ruolo IAM utilizzato per la gestione delle risorse del bilanciatore del carico non è autorizzato a leggere le informazioni di integrità della destinazione. Aggiungere l'autorizzazione elasticloadbalancing:DescribeTargetHealth alla policy del ruolo. Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).

Autorizzazioni di scrittura mancanti sul ruolo di bilanciatore del carico

Messaggio di errore: service myService failed to register targets in target-group myTargetGroup with (error User: arn:aws:sts::123456789012:assumed-role/myELBRole/ecs-service-scheduler is not authorized to perform: elasticloadbalancing:RegisterTargets on resource: arn:aws:elasticloadbalancing:us-west-2:123456789012:targetgroup/myTargetGroup/abc123 because no identity-based policy allows the elasticloadbalancing:RegisterTargets action)

Soluzione: il ruolo IAM utilizzato per la gestione delle risorse del bilanciatore del carico non dispone dell'autorizzazione per registrare le destinazioni. Aggiungere l'autorizzazione elasticloadbalancing:RegisterTargets alla policy del ruolo. Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).

Autorizzazione mancante per modificare le regole del listener

Messaggio di errore: Service deployment rolled back because TEST_TRAFFIC_SHIFT lifecycle hook(s) failed. User: arn:aws:sts::123456789012:assumed-role/myELBRole/ECSNetworkingWithELB is not authorized to perform: elasticloadbalancing:ModifyListener on resource: arn:aws:elasticloadbalancing:us-west-2:123456789012:listener/app/my-alb/abc123/def456 because no identity-based policy allows the elasticloadbalancing:ModifyListener action

Soluzione: il ruolo IAM utilizzato per la gestione delle risorse del bilanciamento del carico non dispone dell'autorizzazione per modificare i listener. Aggiungere l'autorizzazione `elasticloadbalancing:ModifyListener` alla policy del ruolo. Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).

Per le blue/green distribuzioni, si consiglia di allegare la policy `AmazonECS-ServiceLinkedRolePolicy` gestita al proprio ruolo di infrastruttura, che include tutte le autorizzazioni necessarie per la gestione delle risorse di bilanciamento del carico.

Problemi relativi agli hook del ciclo di vita

I seguenti problemi riguardano gli agganci del ciclo di vita nelle distribuzioni. blue/green

Policy di attendibilità errata sul ruolo degli hook di Lambda

Messaggio di errore: `Service deployment rolled back because TEST_TRAFFIC_SHIFT lifecycle hook(s) failed. ECS was unable to assume role arn:aws:iam::123456789012:role/Admin`

Soluzione: il ruolo IAM specificato per l'hook del ciclo di vita di Lambda non dispone della policy di attendibilità corretta. Aggiornare la policy di attendibilità del ruolo per consentire al servizio di assumerlo. La policy di attendibilità deve includere:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

L'hook di Lambda restituisce lo stato FAILED

Messaggio di errore: Service deployment rolled back because TEST_TRAFFIC_SHIFT lifecycle hook(s) failed. Lifecycle hook target arn:aws:lambda:us-west-2:123456789012:function:myHook returned FAILED status.

Soluzione: la funzione Lambda specificata come hook del ciclo di vita ha restituito lo stato FAILED. Controlla i log delle funzioni Lambda nei log di Amazon CloudWatch per determinare il motivo dell'errore e aggiorna la funzione per gestire correttamente l'evento di distribuzione.

Autorizzazione mancante per invocare la funzione Lambda

Messaggio di errore: Service deployment rolled back because TEST_TRAFFIC_SHIFT lifecycle hook(s) failed. ECS was unable to invoke hook target arn:aws:lambda:us-west-2:123456789012:function:myHook due to User: arn:aws:sts::123456789012:assumed-role/myLambdaRole/ECS-Lambda-Execution is not authorized to perform: lambda:InvokeFunction on resource: arn:aws:lambda:us-west-2:123456789012:function:myHook because no identity-based policy allows the lambda:InvokeFunction action

Soluzione: il ruolo IAM utilizzato per l'hook del ciclo di vita di Lambda non dispone dell'autorizzazione per invocare la funzione Lambda. Aggiungere l'autorizzazione lambda:InvokeFunction alla policy del ruolo per l'ARN della funzione Lambda specifica. Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).

Timeout della funzione Lambda o risposta non valida

Messaggio di errore: Service deployment rolled back because TEST_TRAFFIC_SHIFT lifecycle hook(s) failed. ECS was unable to parse the response from arn:aws:lambda:us-west-2:123456789012:function:myHook due to HookStatus must not be null

Soluzione: la funzione Lambda è scaduta o ha restituito una risposta non valida. Assicurarsi che la funzione Lambda restituisca una risposta valida con un campo hookStatus impostato su SUCCEEDED o FAILED. Inoltre, verificare che il timeout della funzione Lambda sia impostato in modo appropriato per la logica di convalida. Per ulteriori informazioni, consultare [Hook del ciclo di vita per le implementazioni di servizi Amazon ECS](#).

Esempio di una risposta Lambda valida:

```
{
  "hookStatus": "SUCCEEDED",
  "reason": "Validation passed"
}
```

Implementazioni lineari di Amazon ECS

Le implementazioni lineari spostano gradualmente il traffico dalla revisione precedente del servizio a quella nuova con incrementi uguali nel tempo, consentendoti di monitorare ogni fase prima di passare alla successiva. Con le implementazioni lineari di Amazon ECS, controlla il ritmo dello spostamento del traffico e convalida nuove revisioni dei servizi con quantità crescenti di traffico di produzione. Questo approccio offre un modo controllato per implementare le modifiche con la possibilità di monitorare le prestazioni ad ogni incremento.

Risorse coinvolte in una distribuzione lineare

Le seguenti sono le risorse coinvolte nelle distribuzioni lineari di Amazon ECS:

- **Spostamento del traffico:** il processo utilizzato da Amazon ECS per spostare il traffico di produzione. Per le implementazioni lineari di Amazon ECS, il traffico viene spostato in incrementi percentuali uguali con tempi di attesa configurabili tra ogni incremento.
- **Percentuale:** la percentuale di traffico da spostare in ogni incremento durante un'implementazione lineare. Questo campo utilizza Double come valore e i valori validi sono compresi tra 3,0 e 100,0.
- **Step Bake Time:** la durata di attesa tra un incremento di traffico e l'altro durante una distribuzione lineare. I valori validi sono compresi tra 0 e 1440 minuti.
- **Tempo di attesa dell'implementazione:** il tempo, in minuti, che Amazon ECS attende dopo aver spostato tutto il traffico di produzione alla nuova revisione del servizio, prima di terminare la vecchia revisione del servizio. Si tratta del periodo in cui le revisioni blu e verde del servizio vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
- **Fasi del ciclo di vita:** una serie di eventi nell'operazione di implementazione, ad esempio “dopo lo spostamento del traffico di produzione”.
- **Lifecycle hook:** una funzione Lambda che viene eseguita in una fase specifica del ciclo di vita. È possibile creare una funzione che verifichi la distribuzione. Le funzioni Lambda o gli hook del ciclo di vita configurati per PRODUCTION_TRAFFIC_SHIFT verranno richiamati in ogni fase del trasferimento del traffico di produzione.

- Gruppo target: una risorsa Elastic Load Balancing utilizzata per instradare le richieste verso uno o più obiettivi registrati (ad esempio, EC2 istanze). Quando si crea un listener, si specifica un gruppo di destinazione per l'operazione predefinita. Il traffico viene inoltrato al gruppo di destinazione specificato nella regola del listener.
- Listener: una risorsa di bilanciamento del carico elastico che controlla le richieste di connessione utilizzando il protocollo e la porta configurata. Le regole definite per un listener determinano il modo in cui Amazon ECS instrada le richieste alle destinazioni registrate.
- Regola: una risorsa di bilanciamento del carico elastico associata a un listener. Una regola definisce il modo in cui le richieste vengono instradate e consiste in un'azione, una condizione e una priorità.

Considerazioni

Si tenga in considerazione quanto segue quando si sceglie un tipo di implementazione:

- Utilizzo delle risorse: le distribuzioni lineari eseguono temporaneamente le revisioni del servizio blu e verde contemporaneamente, il che può raddoppiare l'utilizzo delle risorse durante le distribuzioni.
- Monitoraggio dell'implementazione: le implementazioni lineari forniscono informazioni dettagliate sullo stato dell'implementazione, consentendoti di monitorare ogni fase del processo di implementazione e ogni incremento del traffico.
- Rollback: le implementazioni lineari semplificano il ripristino alla versione precedente se vengono rilevati problemi, poiché la revisione blu viene mantenuta attiva fino alla scadenza del tempo di cottura.
- Convalida graduale: le implementazioni lineari consentono di convalidare la nuova revisione con quantità crescenti di traffico di produzione, offrendo maggiore fiducia nell'implementazione.
- Durata dell'implementazione: le implementazioni lineari richiedono più tempo per essere completate rispetto alle implementazioni a causa dello spostamento incrementale del all-at-once traffico e dei tempi di attesa tra le fasi.

Come funziona l'implementazione lineare

Il processo di distribuzione lineare di Amazon ECS segue un approccio strutturato con sei fasi distinte che garantiscono aggiornamenti delle applicazioni sicuri e affidabili. Ogni fase ha uno scopo specifico nella convalida e nella transizione dell'applicazione dalla versione attuale (blu) alla nuova versione (verde).

1. Fase di preparazione: creare l'ambiente verde insieme a quello blu esistente.
2. Fase di implementazione: implementazione della nuova revisione del servizio nell'ambiente verde. Amazon ECS avvia nuove attività utilizzando la revisione aggiornata del servizio mentre l'ambiente blu continua a servire il traffico di produzione.
3. Fase di test: convalida dell'ambiente verde utilizzando l'instradamento del traffico di test. L'Application Load Balancer indirizza le richieste di test verso l'ambiente verde mentre il traffico di produzione rimane sul blu.
4. Fase di spostamento lineare del traffico: sposta gradualmente il traffico di produzione dal blu al verde con incrementi percentuali uguali in base alla strategia di distribuzione configurata.
5. Fase di monitoraggio: monitoraggio dello stato delle applicazioni, delle metriche delle prestazioni e degli stati di allarme durante il periodo di tempo di incorporamento. Quando vengono rilevati problemi, viene avviata un'operazione di rollback.
6. Fase di completamento: finalizza l'implementazione chiudendo l'ambiente blu.

La fase di spostamento lineare del traffico segue le seguenti fasi:

- Iniziale: l'implementazione inizia con il 100% del traffico indirizzato alla revisione blu (attuale) del servizio. La (nuova) revisione del servizio verde riceve traffico di prova ma inizialmente nessun traffico di produzione.
- Spostamento incrementale del traffico: il traffico viene gradualmente spostato dal blu al verde con incrementi percentuali uguali. Ad esempio, con una configurazione a gradini del 10,0%, gli spostamenti del traffico si verificano come segue:
 - Fase 1: dal 10,0% al verde, al 90,0% al blu
 - Fase 2: 20,0% verso il verde, 80,0% verso il blu
 - Fase 3: 30,0% verso il verde, 70,0% verso il blu
 - E così via fino a quando il 100% raggiunge il verde
- Step Bake Time: tra ogni incremento del traffico, l'implementazione attende una durata configurabile (step bake time) per consentire il monitoraggio e la convalida delle prestazioni della nuova revisione con l'aumento del carico di traffico. Tieni presente che il tempo di cottura dell'ultimo passaggio viene saltato quando il traffico viene spostato del 100,0%.
- Lifecycle Hook: le funzioni Lambda opzionali possono essere eseguite in varie fasi del ciclo di vita durante l'implementazione per eseguire convalida, monitoraggio o logica personalizzata automatizzati. Le funzioni Lambda o gli hook del ciclo di vita configurati per

PRODUCTION_TRAFFIC_SHIFT verranno richiamati in ogni fase del trasferimento del traffico di produzione.

Fasi del ciclo di vita dell'implementazione

Il processo di implementazione lineare procede attraverso fasi distinte del ciclo di vita, ognuna con responsabilità e punti di controllo di convalida specifici. La comprensione di queste fasi consente di monitorare l'avanzamento dell'implementazione e risolvere i problemi in modo efficace.

Ogni fase del ciclo di vita può durare fino a 24 ore e inoltre ogni fase di spostamento del traffico in PRODUCTION_TRAFFIC_SHIFT può durare fino a 24 ore. Si consiglia di mantenere il valore al di sotto della soglia delle 24 ore. Questo perché i processi asincroni richiedono tempo per attivare gli hook. Il sistema scade, fallisce l'implementazione e quindi avvia un rollback dopo che una fase raggiunge le 24 ore.

CloudFormation le distribuzioni hanno restrizioni di timeout aggiuntive. Sebbene il limite delle fasi di 24 ore rimanga in vigore, CloudFormation impone un limite di 36 ore all'intera implementazione. CloudFormation fallisce l'implementazione e quindi avvia un rollback se il processo non viene completato entro 36 ore.

Fasi del ciclo di vita	Description	Supporto per Lifecycle Hook
RECONCILE_SERVICE	Questa fase si verifica solo quando si avvia una nuova implementazione del servizio con più di 1 revisione del servizio in uno stato ACTIVE.	Sì
PRE_SCALE_UP	La revisione del servizio verde non è stata avviata. La revisione del servizio blu gestisce il 100% del traffico di produzione. Non è previsto alcun traffico di test.	Sì
SCALE_UP	Il momento in cui la revisione del servizio verde aumenta fino al 100% e avvia nuove attività. La revisione del servizio verde non serve alcun traffico a questo punto.	No
POST_SCALE_UP	La revisione del servizio verde è stata avviata. La revisione del servizio blu gestisce il 100%	Sì

Fasi del ciclo di vita	Description	Supporto per Lifecycle Hook
	del traffico di produzione. Non è previsto alcun traffico di test.	
TEST_TRAFFIC_SHIFT	Le revisioni del servizio blu e verde sono in esecuzione. La revisione del servizio blu gestisce il 100% del traffico di produzione. La revisione del servizio verde migra dallo 0% al 100% del traffico di test.	Sì
POST_TEST_TRAFFIC_SHIFT	Lo spostamento del traffico di test è completo. La revisione del servizio verde gestisce il 100% del traffico di test.	Sì
PRODUCTION_TRAFFIC_SHIFT	Il traffico viene gradualmente spostato dal blu al verde con incrementi percentuali uguali fino a quando il verde riceve il 100% del traffico. Ogni spostamento del traffico richiama il lifecycle hook con un timeout di 24 ore.	
POST_PRODUCTION_TRAFFIC_SHIFT	Lo spostamento del traffico di produzione è completo.	Sì
BAKE_TIME	La durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente.	No
CLEAN_UP	La revisione del servizio blu è stata completamente ridotta a 0 attività in esecuzione. La revisione del servizio verde è ora la revisione del servizio di produzione dopo questa fase.	No

Risorse richieste per le implementazioni lineari di Amazon ECS

Per utilizzare una distribuzione lineare con spostamento del traffico gestito, il servizio deve utilizzare una delle seguenti funzionalità:

- Application Load Balancer
- Service Connect

Note

Le distribuzioni lineari non supportano Network Load Balancer. Per il supporto di Network Load Balancer, utilizza invece le blue/green distribuzioni.

L'elenco seguente fornisce una panoramica di alto livello di ciò che è necessario configurare per le distribuzioni lineari di Amazon ECS:

- Il servizio utilizza un Application Load Balancer o Service Connect. Configurare le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 - Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).
- Impostare il controller di implementazione del servizio su ECS.
- Configurare la strategia di implementazione come `linear` nella definizione del servizio.
- Facoltativamente, configurare parametri aggiuntivi come:
 - Tempo di incorporamento per la nuova implementazione
 - La percentuale di traffico da spostare in ogni incremento.
 - La durata in minuti di attesa tra ogni incremento di spostamento del traffico.
 - CloudWatch allarmi per il rollback automatico
 - Hook del ciclo di vita della distribuzione (si tratta di funzioni Lambda che vengono eseguite in fasi di distribuzione specifiche come `BEFORE_INSTALL`, `PRODUCTION_TRAFFIC_SHIFT` o `POST_PRODUCTION_TRAFFIC_SHIFT`)

Best practice

Segui queste best practice per implementazioni lineari di Amazon ECS di successo:

- Assicurati che la tua applicazione sia in grado di gestire entrambe le revisioni del servizio in esecuzione simultanea.

- Pianifica una capacità del cluster sufficiente per gestire entrambe le revisioni dei servizi durante la distribuzione.
- Verifica le tue procedure di rollback prima di implementarle in produzione.
- Configurare controlli dell'integrità appropriati che riflettano accuratamente l'integrità dell'applicazione.
- Imposta un tempo di cottura che consenta di testare in modo sufficiente la nuova revisione del servizio.
- Implementa CloudWatch allarmi per rilevare automaticamente i problemi e attivare i rollback.
- Scegli le percentuali di passaggi e i tempi di cottura che bilanciano la velocità di implementazione con le esigenze di convalida.
- Utilizza percentuali di graduale inferiori (5-10%) per le applicazioni critiche per ridurre al minimo l'esposizione al rischio.
- Imposta tempi di cottura più lunghi per le applicazioni che richiedono tempo per riscaldarsi o stabilizzarsi.
- Implementa CloudWatch allarmi per rilevare automaticamente i problemi e attivare il rollback a qualsiasi incremento di traffico.
- Monitora attentamente le metriche delle applicazioni durante ogni spostamento di traffico per rilevare tempestivamente il degrado delle prestazioni.
- Assicurati che la tua applicazione sia in grado di gestire entrambe le revisioni dei servizi in esecuzione simultanea.
- Testa le tue procedure di rollback con diverse percentuali di traffico prima di implementarle in produzione.

Creazione di una distribuzione lineare Amazon ECS

Utilizzando le distribuzioni lineari di Amazon ECS, puoi spostare gradualmente il traffico in incrementi uguali su intervalli di tempo specifici. Ciò fornisce una convalida controllata in ogni fase del processo di distribuzione.

Prerequisiti

Eseguite le seguenti operazioni prima di iniziare una distribuzione lineare.

1. Configurazione delle autorizzazioni appropriate.

- Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).
 - Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).
2. Le implementazioni lineari di Amazon ECS richiedono che il servizio utilizzi una delle seguenti funzionalità: Configura le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 3. Crea una regola per indirizzare il traffico verso la nuova revisione del servizio. Per ulteriori informazioni, consulta le [regole dei listener](#) negli Elastic Load Balancing Application Load Balancers.
 4. Creazione di un gruppo di destinazione per la revisione del servizio verde. Quando si usa la modalità di rete awsvpc per le attività, il tipo di destinazione deve essere ip. Per informazioni sui gruppi target, consulta [Target groups](#) in Elastic Load Balancing Application Load Balancers.

Procedura

Puoi utilizzare la console o il AWS CLI per creare un servizio di distribuzione lineare Amazon ECS.

Console

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Determina la risorsa da cui avviare il servizio.

Per avviare un servizio da	Fasi	
Cluster	- Nella pagina Cluster, seleziona il cluster in cui creare il servizio. - Si apre la pagina dei dettagli del cluster. - Nella scheda Servizi, scegli Crea.	

Per avviare un servizio da	Fasi	
Definizione di attività	- Nella pagina Definizioni di attività, selezionare la definizione di attività. - Dal menu Implementa, scegliere Crea servizio.	

Si apre la pagina Crea servizio.

- In Dettagli del servizio, effettuare le seguenti operazioni:
 - Per Famiglia di definizione di attività, scegliere la definizione di attività da utilizzare. Poi, per Revisione della definizione di attività, inserire la revisione da utilizzare.
 - In Nome servizio, specificare un nome per il servizio.
- Per eseguire il servizio in un cluster esistente, per Cluster esistente, scegliere il cluster. Per eseguire il servizio in un nuovo cluster, scegliere Crea cluster
- Scegliere come vengono distribuite le attività nell'infrastruttura cluster. In Configurazione di calcolo, scegliere l'opzione.

Opzione di calcolo	Fasi	
Strategia del fornitore di capacità	- In Opzioni di calcolo, scegli Strategia del provider di capacità. - 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	

Opzione di calcolo	Fasi	
	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 capacità deve essere associato al cluster.	

Opzione di calcolo	Fasi	
Tipo di lancio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Tipo di avvio, selezionare un tipo di avvio. c. (Facoltativo) Quando viene specificato Fargate, per Versione della piattaforma, specificare la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

6. Nella sezione Configurazione implementazione, procedere come segue.
 - a. Per Tipo di servizio, scegliere Replica.
 - b. Per Attività desiderate, immettere il numero di attività da avviare e gestire nel servizio.
 - c. Per consentire ad Amazon ECS di monitorare la distribuzione delle attività tra le zone di disponibilità e ridistribuirle in caso di squilibrio, in Ribilanciamento del servizio delle zone di disponibilità, selezionare Ribilanciamento del servizio della zona di disponibilità.
 - d. Per Periodo di tolleranza dei controlli di integrità, inserire il periodo di tempo (in secondi) durante il quale il pianificatore di servizi ignora i controlli dell'integrità dei container, di bilanciamento dei carichi elastici e di VPC Lattice non integri dopo che è stata avviata un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0.
7. In Configurazione di distribuzione, configura le impostazioni di distribuzione lineare:

- a. Per Strategia di distribuzione, scegli Linear.
- b. Per Percentuale di incremento del traffico, inserisci la percentuale di traffico da spostare in ogni incremento (ad esempio, 10% per spostare il traffico con incrementi del 10%).
- c. Per Intervallo tra gli incrementi, inserisci il tempo di attesa in minuti tra ogni incremento di spostamento del traffico.
- d. Per Bake time, inserisci il numero di minuti in cui entrambe le revisioni del servizio blu e verde verranno eseguite contemporaneamente dopo l'ultimo cambio di traffico prima che la revisione blu venga terminata.
- e. (Facoltativo) Eseguire le funzioni Lambda in fasi specifiche dell'implementazione. In Hook del ciclo di vita di implementazione, selezionare le fasi per eseguire gli hook del ciclo di vita.

Per aggiungere un hook del ciclo di vita:

- a. Scegliere Aggiungi.
 - b. Per Funzione Lambda, immettere il nome della funzione o ARN.
 - c. Per Ruolo, selezionare il ruolo IAM che ha il permesso di invocare la funzione Lambda.
 - d. Per Fasi del ciclo di vita, selezionare le fasi in cui deve essere eseguita la funzione Lambda.
8. 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 l'implementazione all'ultimo stato di implementazione completata quando l'interruttore automatico di implementazione imposta l'implementazione su uno stato di errore, selezionare Rollback in caso di errore.

- 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.

9. (Facoltativo) Per connettere il servizio usando Service Connect, espandere Service Connect, quindi specificare quanto segue:
 - a. Selezionare Attiva Service Connect.
 - b. 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 nel namespace, scegliere 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 nel namespace, scegliere Client e server.
 - c. Per utilizzare un namespace differente da quello del cluster predefinito, per Namespace, scegliere il namespace del servizio. Può trattarsi di uno spazio dei nomi creato separatamente Regione AWS nello stesso spazio dell'utente Account AWS o di uno spazio dei nomi nella stessa regione condiviso con il proprio account utilizzando (). AWS Resource Access Manager AWS RAM Per ulteriori informazioni sugli spazi dei AWS Cloud Map nomi condivisi, consulta [Condivisione dello spazio dei nomi tra AWS Cloud Map](#) account nella Guida per gli sviluppatori. AWS Cloud Map
 - d. (Facoltativo) Configura le regole di test traffic header per le distribuzioni lineari. In Instradamento del traffico di test, specificare quanto segue:
 - i. Selezionare Abilita le regole di intestazione del traffico di test per instradare richieste specifiche alla revisione del servizio verde durante il test.
 - ii. Per Regole di corrispondenza di intestazione, configurare i criteri per l'instradamento del traffico di test:
 - Nome dell'intestazione: inserire il nome dell'intestazione HTTP per la corrispondenza (ad esempio, X-Test-Version o User-Agent).
 - Tipo di corrispondenza: scegliere i criteri di corrispondenza:
 - Corrispondenza esatta: richieste di instradamento in cui il valore dell'intestazione corrisponde esattamente al valore specificato

- Intestazione presente: richieste di instradamento che contengono l'intestazione specificata, indipendentemente dal valore
- Corrispondenza di schema: richieste di instradamento in cui il valore dell'intestazione corrisponde a uno schema specificato
- Valore dell'intestazione (se si utilizza la corrispondenza esatta o la corrispondenza di schema): inserire il valore o lo schema a cui far corrispondere l'intestazione.

È possibile aggiungere più regole di corrispondenza dell'intestazione per creare una logica di instradamento complessa. Le richieste che corrispondono a una qualsiasi delle regole configurate verranno instradate alla revisione del servizio verde per essere testate.

- iii. Scegliere Aggiungi regola di intestazione per configurare condizioni aggiuntive di corrispondenza dell'intestazione.

 Note

Le regole di intestazione del traffico di test consentono di convalidare nuove funzionalità con traffico controllato prima di portare a termine l'implementazione completa. Ciò consente di testare la revisione del servizio verde con richieste specifiche (come quelle provenienti da strumenti di test interni o utenti beta) mantenendo al contempo il normale flusso di traffico verso la revisione del servizio blu.

- e. (Facoltativo) Specificare una configurazione del log. Selezionare Usa la raccolta di log. L'opzione predefinita invia i log dei contenitori a Logs. CloudWatch 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 del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
- **Flusso di dati Amazon Kinesis:** configura il processo per inviare log di container a Kinesis Data Streams. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna 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 log di container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del bucket Amazon S3 valido.

10. (Facoltativo) Configura il bilanciamento del carico per la distribuzione lineare.

Tipo di bilanciamento del carico elastico	Fasi	
Application Load Balancer	a. Per Tipo di bilanciatore del carico, scegliere Application 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 Container, scegliere il container che ospita il servizio.	

Tipo di bilanciamento del carico elastico	Fasi	
	d. Per Nome del bilanciamento del carico, inserire un nome univoco. 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. • Per Regola di produzione, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di produzione. • Per Regola di test, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di test.	

Tipo di bilanciamento del carico elastico	Fasi	
	f. Per Gruppo di destinazione, configurare quanto segue: • Per Nome del gruppo di destinazione, specificare un nome e un protocollo per il gruppo di destinazione a cui l'Application Load Balancer instraderà le richieste. • Per Protocollo, scegliere il 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. • Per il ritardo di annullamento della registrazione, inserisci il numero di secondi in cui il sistema di bilanciamento del carico modificherà lo stato di destinazione. UNUSED Il valore	

Tipo di bilanciamento del carico elastico	Fasi	
	predefinito è 300 secondi. • 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 Nome gruppo alternativo, inserisci il nome del gruppo target per la revisione del servizio di test.	

11. (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).

12. Scegli Create (Crea).

AWS CLI

1. Crea un archivio denominato `linear-service-definition.json` con i seguenti contenuti.

Sostituiscili *user-input* con i tuoi valori.

```
{
  "serviceName": "myLinearService",
  "cluster": "arn:aws:ecs:us-west-2:123456789012:cluster/sample-fargate-cluster",
  "taskDefinition": "sample-fargate:1",
  "desiredCount": 5,
  "launchType": "FARGATE",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "subnets": [
        "subnet-09ce6e74c116a2299",
        "subnet-00bb3bd7a73526788",
        "subnet-0048a611aaec65477"
      ],
      "securityGroups": [
        "sg-09d45005497daa123"
      ],
      "assignPublicIp": "ENABLED"
    }
  },
  "deploymentController": {
    "type": "ECS"
  },
  "deploymentConfiguration": {
    "strategy": "LINEAR",
    "maximumPercent": 200,
    "minimumHealthyPercent": 100,
  }
}
```

```

"linearConfiguration": {
  "stepPercentage": 10.0,
  "stepBakeTimeInMinutes": 6
},
"bakeTimeInMinutes": 10,
"alarms": {
  "alarmNames": [
    "myAlarm"
  ],
  "rollback": true,
  "enable": true
},
"loadBalancers": [
  {
    "targetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-target-group/54402ff563af1197",
    "containerName": "fargate-app",
    "containerPort": 80,
    "advancedConfiguration": {
      "alternateTargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-target-group/cad10a56f5843199",
      "productionListenerRule": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener-rule/app/my-linear-
demo/32e0e4f946c3c05b/9cfa8c482e204f7d/831dbaf72edb911",
      "roleArn": "arn:aws:iam::123456789012:role/LoadBalancerManagementforECS"
    }
  }
]
}

```

2. Esegui `create-service`.

```
aws ecs create-service --cli-input-json file://linear-service-definition.json
```

Fasi successive

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).
- Monitora il processo di implementazione per assicurarti che segua lo schema lineare:
 - La revisione del servizio verde viene creata e aumentata verticalmente

- Il traffico viene spostato in modo incrementale agli intervalli specificati
- Ogni incremento attende l'intervallo specificato prima del turno successivo
- Dopo che tutto il traffico è stato spostato, inizia il tempo di cottura
- Dopo il tempo di incorporamento, la revisione blu viene interrotta

Distribuzioni canary di Amazon ECS

Le implementazioni Canary indirizzano innanzitutto una piccola percentuale di traffico verso la nuova revisione per i test iniziali, quindi spostano tutto il traffico rimanente contemporaneamente dopo il completamento con successo della fase canaria. Con le implementazioni di Amazon ECS Canary, convalida le nuove revisioni dei servizi con traffico utente reale, riducendo al minimo l'esposizione al rischio. Questo approccio offre un modo controllato per implementare le modifiche con la possibilità di monitorare le prestazioni e ripristinare rapidamente se vengono rilevati problemi.

Risorse coinvolte nell'implementazione di un canarino

Le seguenti sono le risorse coinvolte nelle implementazioni di Amazon ECS Canary:

- Spostamento del traffico: il processo utilizzato da Amazon ECS per spostare il traffico di produzione. Per le implementazioni di Amazon ECS Canary, il traffico viene spostato in due fasi: prima verso la percentuale canaria, quindi verso il completamento della distribuzione.
- Percentuale canaria: la percentuale di traffico indirizzato alla nuova versione durante il periodo di valutazione.
- Tempo di cottura di Canary: la durata del monitoraggio della versione Canary prima di procedere con la distribuzione completa.
- Tempo di attesa dell'implementazione: il tempo, in minuti, che Amazon ECS attende dopo aver spostato tutto il traffico di produzione alla nuova revisione del servizio, prima di terminare la vecchia revisione del servizio. Si tratta del periodo in cui le revisioni blu e verde del servizio vengono eseguite contemporaneamente dopo lo spostamento del traffico di produzione.
- Fasi del ciclo di vita: una serie di eventi nell'operazione di implementazione, ad esempio “dopo lo spostamento del traffico di produzione”.
- Lifecycle hook: una funzione Lambda che viene eseguita in una fase specifica del ciclo di vita. È possibile creare una funzione che verifichi la distribuzione.
- Gruppo target: una risorsa Elastic Load Balancing utilizzata per instradare le richieste verso uno o più obiettivi registrati (ad esempio, EC2 istanze). Quando si crea un listener, si specifica un gruppo

di destinazione per l'operazione predefinita. Il traffico viene inoltrato al gruppo di destinazione specificato nella regola del listener.

- **Listener:** una risorsa di bilanciamento del carico elastico che controlla le richieste di connessione utilizzando il protocollo e la porta configurata. Le regole definite per un listener determinano il modo in cui Amazon ECS instrada le richieste alle destinazioni registrate.
- **Regola:** una risorsa di bilanciamento del carico elastico associata a un listener. Una regola definisce il modo in cui le richieste vengono instradate e consiste in un'azione, una condizione e una priorità.

Considerazioni

Si tenga in considerazione quanto segue quando si sceglie un tipo di implementazione:

- **Utilizzo delle risorse:** le implementazioni Canary eseguono contemporaneamente sia i set di attività originali che quelli di Canary durante il periodo di valutazione, aumentando l'utilizzo delle risorse.
- **Volume di traffico:** assicurati che la percentuale di Canary generi traffico sufficiente per una convalida significativa della nuova versione.
- **Complessità del monitoraggio:** le implementazioni di Canary richiedono il monitoraggio e il confronto delle metriche tra due versioni diverse contemporaneamente.
- **Velocità di rollback:** le implementazioni Canary consentono un ripristino rapido riportando il traffico al set di attività originale.
- **Mitigazione del rischio:** le implementazioni Canary offrono un'eccellente mitigazione del rischio limitando l'esposizione a una piccola percentuale di utenti.
- **Durata dell'implementazione:** le implementazioni di Canary includono periodi di valutazione che prolungano il tempo complessivo di implementazione ma offrono opportunità di convalida.

Come funzionano le implementazioni di Canary

Il processo di distribuzione di Amazon ECS Canary segue un approccio strutturato con sei fasi distinte che garantiscono aggiornamenti delle applicazioni sicuri e affidabili. Ogni fase ha uno scopo specifico nella convalida e nella transizione dell'applicazione dalla versione attuale (blu) alla nuova versione (verde).

1. Fase di preparazione: creare l'ambiente verde insieme a quello blu esistente.

2. Fase di implementazione: implementazione della nuova revisione del servizio nell'ambiente verde. Amazon ECS avvia nuove attività utilizzando la revisione aggiornata del servizio mentre l'ambiente blu continua a servire il traffico di produzione.
3. Fase di test: convalida dell'ambiente verde utilizzando l'instradamento del traffico di test. L'Application Load Balancer indirizza le richieste di test verso l'ambiente verde mentre il traffico di produzione rimane sul blu.
4. Fase di trasferimento del traffico nelle Canarie: sposta la percentuale di traffico configurata verso la nuova revisione del servizio ecologico durante la fase canaria, seguita dallo spostamento del 100,0% del traffico verso la revisione del servizio verde
5. Fase di monitoraggio: monitoraggio dello stato delle applicazioni, delle metriche delle prestazioni e degli stati di allarme durante il periodo di tempo di incorporamento. Quando vengono rilevati problemi, viene avviata un'operazione di rollback.
6. Fase di completamento: finalizza l'implementazione chiudendo l'ambiente blu.

La fase di trasferimento del traffico nelle Canarie segue le seguenti fasi:

- Iniziale: l'implementazione inizia con il 100% del traffico indirizzato alla revisione blu (attuale) del servizio. La (nuova) revisione del servizio verde riceve traffico di prova ma inizialmente nessun traffico di produzione.
- Spostamento del traffico nelle Canarie - Questa è una strategia di spostamento del traffico in due fasi.
 - Fase 1: 10,0% verso il verde, 90,0% verso il blu
 - Fase 2: 100,0% verso il verde, 0,0% verso il blu
- Tempo di cottura per Canary Traffic - Attende una durata configurabile (Canary Bake Time) dopo il cambio di marcia per consentire il monitoraggio e la convalida delle prestazioni della nuova revisione con l'aumento del traffico.
- Lifecycle Hook: le funzioni Lambda opzionali possono essere eseguite in varie fasi del ciclo di vita durante l'implementazione per eseguire convalida, monitoraggio o logica personalizzata automatizzati. Le funzioni Lambda o gli hook del ciclo di vita configurati per `PRODUCTION_TRAFFIC_SHIFT` verranno richiamati in ogni fase del trasferimento del traffico di produzione.

Fasi del ciclo di vita dell'implementazione

Il processo di implementazione di Canary procede attraverso fasi distinte del ciclo di vita, ognuna con responsabilità e punti di controllo di convalida specifici. La comprensione di queste fasi consente di monitorare l'avanzamento dell'implementazione e risolvere i problemi in modo efficace.

Ogni fase del ciclo di vita può durare fino a 24 ore e inoltre ogni fase di spostamento del traffico in `PRODUCTION_TRAFFIC_SHIFT` può durare fino a 24 ore. Si consiglia di mantenere il valore al di sotto della soglia delle 24 ore. Questo perché i processi asincroni richiedono tempo per attivare gli hook. Il sistema scade, fallisce l'implementazione e quindi avvia un rollback dopo che una fase raggiunge le 24 ore.

CloudFormation le distribuzioni hanno restrizioni di timeout aggiuntive. Sebbene il limite delle fasi di 24 ore rimanga in vigore, CloudFormation impone un limite di 36 ore all'intera implementazione. CloudFormation fallisce l'implementazione e quindi avvia un rollback se il processo non viene completato entro 36 ore.

Fasi del ciclo di vita

Fasi del ciclo di vita	Description	Supporto per Lifecycle Hook
RECONCILE_SERVICE	Questa fase si verifica solo quando si avvia una nuova implementazione del servizio con più di 1 revisione del servizio in uno stato ACTIVE.	Sì
PRE_SCALE_UP	La revisione del servizio verde non è stata avviata. La revisione del servizio blu gestisce il 100% del traffico di produzione. Non è previsto alcun traffico di test.	Sì
SCALE_UP	Il momento in cui la revisione del servizio verde aumenta fino al 100% e avvia nuove attività. La revisione del servizio verde non serve alcun traffico a questo punto.	No
POST_SCALE_UP	La revisione del servizio verde è stata avviata. La revisione del servizio blu gestisce il 100% del traffico di produzione. Non è previsto alcun traffico di test.	Sì

Fasi del ciclo di vita	Description	Supporto per Lifecycle Hook
TEST_TRAFFIC_SHIFT	Le revisioni del servizio blu e verde sono in esecuzione. La revisione del servizio blu gestisce il 100% del traffico di produzione. La revisione del servizio verde migra dallo 0% al 100% del traffico di test.	Sì
POST_TEST_TRAFFIC_SHIFT	Lo spostamento del traffico di test è completo. La revisione del servizio verde gestisce il 100% del traffico di test.	Sì
PRODUCTION_TRAFFIC_SHIFT	Il traffico di produzione delle Canarie viene indirizzato alla revisione verde e lifecycle hook viene richiamato con un timeout di 24 ore. La seconda fase sposta il traffico di produzione residuo verso la revisione verde.	Sì
POST_PRODUCTION_TRAFFIC_SHIFT	Lo spostamento del traffico di produzione è completo.	Sì
BAKE_TIME	La durata in cui le revisioni del servizio blu e verde vengono eseguite contemporaneamente.	No
CLEAN_UP	La revisione del servizio blu è stata completamente ridotta a 0 attività in esecuzione. La revisione del servizio verde è ora la revisione del servizio di produzione dopo questa fase.	No

Parametri di configurazione

Le implementazioni Canary richiedono i seguenti parametri di configurazione:

- **Percentuale delle Canarie:** la percentuale di traffico da indirizzare verso la nuova revisione del servizio durante la fase canaria. Ciò consente di eseguire test con un sottoinsieme controllato del traffico di produzione.

- **Tempo di cottura delle Canarie:** la durata di attesa durante la fase canaria prima di spostare il traffico rimanente verso la nuova revisione del servizio. In questo modo si ha il tempo necessario per monitorare e convalidare la nuova versione.

Gestione del traffico

Le implementazioni Canary utilizzano gruppi target di sistemi di bilanciamento del carico per gestire la distribuzione del traffico:

- **Gruppo target originale:** contiene le attività dell'attuale versione stabile e riceve la maggior parte del traffico.
- **Gruppo target Canary:** contiene le attività della nuova versione e riceve una piccola percentuale di traffico per i test.
- **Routing ponderato:** il load balancer utilizza regole di routing ponderate per distribuire il traffico tra i gruppi target in base alla percentuale di canarie configurata.

Monitoraggio e convalida

Le implementazioni efficaci di Canary si basano su un monitoraggio completo:

- **Controlli sanitari:** entrambi i set di attività devono superare i controlli di integrità prima di ricevere traffico.
- **Confronto delle metriche:** confronta gli indicatori chiave delle prestazioni tra la versione originale e quella di Canary, come il tempo di risposta, il tasso di errore e la velocità effettiva.
- **Rollback automatico:** configura gli CloudWatch allarmi per attivare automaticamente il rollback se la versione Canary mostra prestazioni inferiori.
- **Convalida manuale:** utilizza il periodo di valutazione per esaminare manualmente i registri, le metriche e il feedback degli utenti prima di procedere.

Le migliori pratiche per le implementazioni di Canary

Segui queste best practice per garantire il successo delle implementazioni di Canary con i relativi servizi.

Scegli le percentuali di traffico appropriate

Considera questi fattori quando selezioni le percentuali di traffico delle Canarie:

- Inizia in piccolo: inizia con il 5-10% del traffico per ridurre al minimo l'impatto in caso di problemi.
- Considerate la criticità delle applicazioni: utilizzate percentuali più basse per le applicazioni mission critical e percentuali più alte per i servizi meno critici.
- Tieni conto del volume di traffico: assicurati che la percentuale canaria generi traffico sufficiente per una convalida significativa.

Stabilisci periodi di valutazione appropriati

Configura i periodi di valutazione in base a queste considerazioni:

- Concedi un tempo sufficiente: imposta periodi di valutazione sufficientemente lunghi da acquisire dati significativi sulle prestazioni, in genere 10-30 minuti.
- Considerate i modelli di traffico: tenete conto dei modelli di traffico dell'applicazione e dei periodi di picco di utilizzo.
- Equilibra velocità e sicurezza: periodi di valutazione più lunghi forniscono più dati ma rallentano la velocità di implementazione.

Implementa un monitoraggio completo

Imposta il monitoraggio per monitorare le prestazioni di implementazione di Canary:

- Metriche chiave: monitora il tempo di risposta, il tasso di errore, la velocità effettiva e l'utilizzo delle risorse per entrambi i set di attività.
- Rollback basato sugli allarmi: configura gli CloudWatch allarmi per attivare automaticamente il rollback quando le metriche superano le soglie.
- Analisi comparativa: configura dashboard per confrontare le metriche tra la versione originale e quella di Canary. side-by-side
- Metriche aziendali: oltre alle metriche tecniche, includi metriche specifiche aziendali come i tassi di conversione o il coinvolgimento degli utenti.

Pianifica le strategie di rollback

Preparati a potenziali scenari di rollback con queste strategie:

- Rollback automatizzato: configura i trigger di rollback automatici in base ai controlli di integrità e alle metriche delle prestazioni.

- Procedure di rollback manuali: documenta procedure chiare per il rollback manuale quando i trigger automatici non rilevano tutti i problemi.
- Test di rollback: verifica regolarmente le procedure di rollback per assicurarti che funzionino correttamente quando necessario.

Effettua una convalida completa prima dell'implementazione

Garantisce una convalida completa prima di procedere con le implementazioni di Canary:

- Test pre-implementazione: verifica accuratamente le modifiche negli ambienti di staging prima dell'implementazione di Canary.
- Configurazione del controllo dello stato: assicurati che i controlli dello stato riflettano accuratamente la disponibilità e la funzionalità delle applicazioni.
- Convalida delle dipendenze: verifica che le nuove versioni siano compatibili con i servizi downstream e upstream.
- Coerenza dei dati: assicurati che le modifiche allo schema del database e le migrazioni dei dati siano compatibili con le versioni precedenti.

Coordina il coinvolgimento del team

Garantisce un coordinamento efficace del team durante l'impiego dei canarini:

- Finestre di implementazione: pianifica le implementazioni di Canary durante l'orario lavorativo, quando i team sono disponibili per il monitoraggio e la risposta.
- Canali di comunicazione: stabilisci canali di comunicazione chiari per lo stato dell'implementazione e la risoluzione dei problemi.
- Assegnazione dei ruoli: definisci ruoli e responsabilità per il monitoraggio, il processo decisionale e l'esecuzione del rollback.

Risorse richieste per le implementazioni di Amazon ECS Canary

Per utilizzare una distribuzione Canary con Managed Traffic Shifting, il tuo servizio deve utilizzare una delle seguenti funzionalità:

- Elastic Load Balancing
- Service Connect

 Note

Le implementazioni Canary non supportano Network Load Balancer. Per le configurazioni di Network Load Balancer, utilizza blue/green invece le distribuzioni.

L'elenco seguente fornisce una panoramica di alto livello di ciò che è necessario configurare per le distribuzioni di Amazon ECS Canary:

- Il servizio utilizza un Application Load Balancer o Service Connect. Configurare le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 - Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).
- Impostare il controller di implementazione del servizio su ECS.
- Configurare la strategia di implementazione come `canary` nella definizione del servizio.
- Facoltativamente, configurare parametri aggiuntivi come:
 - Tempo di incorporamento per la nuova implementazione
 - La percentuale di traffico da indirizzare verso la nuova revisione del servizio durante la fase canaria.
 - La durata di attesa durante la fase canaria prima di spostare il traffico rimanente verso la nuova revisione del servizio.
 - CloudWatch allarmi per il rollback automatico
 - Hook del ciclo di vita della distribuzione (si tratta di funzioni Lambda che vengono eseguite in fasi di implementazione specifiche)

Best practice

Segui queste best practice per implementazioni di Amazon ECS olcanary di successo:

- Assicurati che la tua applicazione sia in grado di gestire entrambe le revisioni dei servizi in esecuzione simultanea.
- Pianifica una capacità del cluster sufficiente per gestire entrambe le revisioni dei servizi durante la distribuzione.

- Verifica le tue procedure di rollback prima di implementarle in produzione.
- Configurare controlli dell'integrità appropriati che riflettano accuratamente l'integrità dell'applicazione.
- Impostare un tempo di incorporamento che consenta di testare in modo sufficiente l'implementazione verde.
- Implementa CloudWatch allarmi per rilevare automaticamente i problemi e attivare i rollback.
- Utilizzare gli hook del ciclo di vita per eseguire test automatici in ogni fase di implementazione.
- Inizia con piccole percentuali di canarino (5-10%) per ridurre al minimo l'impatto in caso di problemi.
- Stabilisci periodi di valutazione appropriati che offrano tempo sufficiente per una raccolta significativa dei dati sulle prestazioni.
- Implementa un monitoraggio completo con CloudWatch allarmi per i trigger di rollback automatici.
- Configura controlli di integrità che riflettano accuratamente la disponibilità e la funzionalità dell'applicazione.
- Monitora sia le metriche tecniche (tempo di risposta, tasso di errore) che le metriche aziendali durante la valutazione.
- Assicurati che la tua applicazione sia in grado di gestire la suddivisione del traffico senza problemi di sessione o di stato.
- Pianifica le procedure di rollback e testale regolarmente per assicurarti che funzionino quando necessario.
- Pianifica le implementazioni di Canary durante l'orario lavorativo, quando i team possono monitorare e rispondere.
- Convalida accuratamente le modifiche negli ambienti di staging prima dell'implementazione di Canary.
- Documenta procedure chiare per l'intervento manuale e le decisioni di ripristino.

Creazione di una distribuzione Amazon ECS Canary

Utilizzando le implementazioni di Amazon ECS Canary, puoi spostare una piccola percentuale di traffico verso la tua nuova revisione del servizio (il «canarino»). Convalida la distribuzione e poi sposta il traffico rimanente tutto in una volta dopo un intervallo specificato. Questo approccio consente di testare nuove funzionalità con un rischio minimo prima dell'implementazione completa.

Prerequisiti

Eseguite le seguenti operazioni prima di iniziare una distribuzione Canary.

1. Configurazione delle autorizzazioni appropriate.
 - Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).
 - Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).
2. Le implementazioni di Amazon ECS Canary richiedono che il servizio utilizzi una delle seguenti funzionalità: Configura le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
3. Crea una regola per indirizzare il traffico verso la nuova revisione del servizio. Per ulteriori informazioni, consulta le [regole dei listener](#) negli Elastic Load Balancing Application Load Balancers.
4. Creazione di un gruppo di destinazione per la revisione del servizio verde. Quando si usa la modalità di rete aws_vpc per le attività, il tipo di destinazione deve essere ip. Per informazioni sui gruppi target, consulta [Target groups](#) in Elastic Load Balancing Application Load Balancers.

Procedura

Puoi utilizzare la console o AWS CLI creare un servizio di distribuzione Amazon ECS Canary.

Console

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
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. Si apre la pagina dei dettagli del cluster.	

Per avviare un servizio da	Fasi	
	b. Nella scheda Servizi, scegli Crea.	
Definizione di attività	a. Nella pagina Definizioni di attività, selezionare la definizione di attività. b. Dal menu Implementa, scegliere Crea servizio.	

Si apre la pagina Crea servizio.

3. In Dettagli del servizio, effettuare le seguenti operazioni:
 - a. Per Famiglia di definizione di attività, scegliere la definizione di attività da utilizzare. Poi, per Revisione della definizione di attività, inserire la revisione da utilizzare.
 - b. In Nome servizio, specificare un nome per il servizio.
4. Per eseguire il servizio in un cluster esistente, per Cluster esistente, scegliere il cluster. Per eseguire il servizio in un nuovo cluster, scegliere Crea cluster
5. Scegliere come vengono distribuite le attività nell'infrastruttura cluster. In Configurazione di calcolo, scegliere l'opzione.

Opzione di calcolo	Fasi	
Strategia del fornitore 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).	

Opzione di calcolo	Fasi	
	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 capacità deve essere associato al cluster.	

Opzione di calcolo	Fasi	
Tipo di lancio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Tipo di avvio, selezionare un tipo di avvio. c. (Facoltativo) Quando viene specificato Fargate, per Versione della piattaforma, specificare la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma LATEST.	

6. Nella sezione Configurazione implementazione, procedere come segue.
 - a. Per Tipo di servizio, scegliere Replica.
 - b. Per Attività desiderate, immettere il numero di attività da avviare e gestire nel servizio.
 - c. Per consentire ad Amazon ECS di monitorare la distribuzione delle attività tra le zone di disponibilità e ridistribuirle in caso di squilibrio, in Ribilanciamento del servizio delle zone di disponibilità, selezionare Ribilanciamento del servizio della zona di disponibilità.
 - d. Per Periodo di tolleranza dei controlli di integrità, inserire il periodo di tempo (in secondi) durante il quale il pianificatore di servizi ignora i controlli dell'integrità dei container, di bilanciamento dei carichi elastici e di VPC Lattice non integri dopo che è stata avviata un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0.
7. In Configurazione di distribuzione, configura le impostazioni di distribuzione di Canary:

- a. Per la strategia di distribuzione, scegli Canary.
- b. Per la percentuale di Canarie, inserisci la percentuale di traffico da passare alla revisione del servizio verde nella prima fase (ad esempio, 10% per il traffico iniziale delle Canarie).
- c. Per Canary Bake time, inserisci il tempo di attesa in minuti prima di spostare il traffico rimanente sulla revisione del servizio ecologico.
- d. Per Tempo di cottura, inserisci il numero di minuti in cui entrambe le revisioni del servizio blu e verde verranno eseguite contemporaneamente dopo l'ultimo cambiamento di traffico prima che la revisione blu venga interrotta.
- e. (Facoltativo) Eseguire le funzioni Lambda in fasi specifiche dell'implementazione. In Hook del ciclo di vita di implementazione, selezionare le fasi per eseguire gli hook del ciclo di vita.

Per aggiungere un hook del ciclo di vita:

- a. Scegliere Aggiungi.
 - b. Per Funzione Lambda, immettere il nome della funzione o ARN.
 - c. Per Ruolo, selezionare il ruolo IAM che ha il permesso di invocare la funzione Lambda.
 - d. Per Fasi del ciclo di vita, selezionare le fasi in cui deve essere eseguita la funzione Lambda.
8. 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 l'implementazione all'ultimo stato di implementazione completata quando l'interruttore automatico di implementazione imposta l'implementazione su uno stato di errore, selezionare Rollback in caso di errore.

- 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.

9. (Facoltativo) Per connettere il servizio usando Service Connect, espandere Service Connect, quindi specificare quanto segue:
 - a. Selezionare Attiva Service Connect.
 - b. 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 nel namespace, scegliere 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 nel namespace, scegliere Client e server.
 - c. Per utilizzare un namespace differente da quello del cluster predefinito, per Namespace, scegliere il namespace del servizio. Può trattarsi di uno spazio dei nomi creato separatamente Regione AWS nello stesso spazio dell'utente Account AWS o di uno spazio dei nomi nella stessa regione condiviso con il proprio account utilizzando (). AWS Resource Access Manager AWS RAM Per ulteriori informazioni sugli spazi dei AWS Cloud Map nomi condivisi, consulta [Condivisione dello spazio dei nomi tra AWS Cloud Map](#) account nella Guida per gli sviluppatori. AWS Cloud Map
 - d. (Facoltativo) Configura le regole di test traffic header per le implementazioni Canary. In Instradamento del traffico di test, specificare quanto segue:
 - i. Selezionare Abilita le regole di intestazione del traffico di test per instradare richieste specifiche alla revisione del servizio verde durante il test.
 - ii. Per Regole di corrispondenza di intestazione, configurare i criteri per l'instradamento del traffico di test:
 - Nome dell'intestazione: inserire il nome dell'intestazione HTTP per la corrispondenza (ad esempio, X-Test-Version o User-Agent).
 - Tipo di corrispondenza: scegliere i criteri di corrispondenza:
 - Corrispondenza esatta: richieste di instradamento in cui il valore dell'intestazione corrisponde esattamente al valore specificato

- Intestazione presente: richieste di instradamento che contengono l'intestazione specificata, indipendentemente dal valore
- Corrispondenza di schema: richieste di instradamento in cui il valore dell'intestazione corrisponde a uno schema specificato
- Valore dell'intestazione (se si utilizza la corrispondenza esatta o la corrispondenza di schema): inserire il valore o lo schema a cui far corrispondere l'intestazione.

È possibile aggiungere più regole di corrispondenza dell'intestazione per creare una logica di instradamento complessa. Le richieste che corrispondono a una qualsiasi delle regole configurate verranno instradate alla revisione del servizio verde per essere testate.

- iii. Scegliere Aggiungi regola di intestazione per configurare condizioni aggiuntive di corrispondenza dell'intestazione.

 Note

Le regole di intestazione del traffico di test consentono di convalidare nuove funzionalità con traffico controllato prima di portare a termine l'implementazione completa. Ciò consente di testare la revisione del servizio verde con richieste specifiche (come quelle provenienti da strumenti di test interni o utenti beta) mantenendo al contempo il normale flusso di traffico verso la revisione del servizio blu.

- e. (Facoltativo) Specificare una configurazione del log. Selezionare Usa la raccolta di log. L'opzione predefinita invia i log dei contenitori a Logs. CloudWatch 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 del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
- **Flusso di dati Amazon Kinesis:** configura il processo per inviare log di container a Kinesis Data Streams. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna 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 log di container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del bucket Amazon S3 valido.

10. (Facoltativo) Configura il bilanciamento del carico per l'implementazione di Canary.

Tipo di bilanciamento del carico elastico	Fasi	
Application Load Balancer	a. Per Tipo di bilanciatore del carico, scegliere Application 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 Container, scegliere il container che ospita il servizio.	

Tipo di bilanciamento del carico elastico	Fasi	
	d. Per Nome del bilanciamento del carico, inserire un nome univoco. 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. • Per Regola di produzione, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di produzione (blu). • Per Regola di test, inserire l'Ordine di valutazione e lo Schema di percorso per la regola. Questa regola riguarda il traffico di revisione del servizio di test (verde).	

Tipo di bilanciamento del carico elastico	Fasi	
	f. Per Gruppo di destinazione, configurare quanto segue: • Per Nome del gruppo di destinazione, specificare un nome e un protocollo per il gruppo di destinazione a cui l'Application Load Balancer instraderà le richieste. • Per Protocollo, scegliere il 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. • Per il ritardo di annullamento della registrazione, inserisci il numero di secondi in cui il sistema di bilanciamento del carico modificherà lo stato di destinazione. UNUSED Il valore	

Tipo di bilanciamento del carico elastico	Fasi	
	predefinito è 300 secondi. • 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 Nome gruppo alternativo, inserisci il nome del gruppo target per la revisione del servizio di test.	

11. (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).

12. Scegli Create (Crea).

AWS CLI

1. Crea un archivio denominato `canary-service-definition.json` con i seguenti contenuti.

Sostituiscili *user-input* con i tuoi valori.

```
{
  "serviceName": "myCanaryService",
  "cluster": "arn:aws:ecs:us-west-2:123456789012:cluster/sample-fargate-cluster",
  "taskDefinition": "sample-fargate:1",
  "desiredCount": 5,
  "launchType": "FARGATE",
  "networkConfiguration": {
    "awsvpcConfiguration": {
      "subnets": [
        "subnet-09ce6e74c116a2299",
        "subnet-00bb3bd7a73526788",
        "subnet-0048a611aaec65477"
      ],
      "securityGroups": [
        "sg-09d45005497daa123"
      ],
      "assignPublicIp": "ENABLED"
    }
  },
  "deploymentController": {
    "type": "ECS"
  },
  "deploymentConfiguration": {
    "strategy": "CANARY",
    "maximumPercent": 200,
    "minimumHealthyPercent": 100,
  }
}
```



```

    "canaryConfiguration" : {
      "canaryPercent" : 5.0,
      "canaryBakeTime" : 10
    },
    "bakeTimeInMinutes": 10,
    "alarms": {
      "alarmNames": [
        "myAlarm"
      ],
      "rollback": true,
      "enable": true
    }
  },
  "loadBalancers": [
    {
      "targetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/blue-target-group/54402ff563af1197",
      "containerName": "fargate-app",
      "containerPort": 80,
      "advancedConfiguration": {
        "alternateTargetGroupArn": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:targetgroup/green-target-group/cad10a56f5843199",
        "productionListenerRule": "arn:aws:elasticloadbalancing:us-
west-2:123456789012:listener-rule/app/my-canary-
demo/32e0e4f946c3c05b/9cfa8c482e204f7d/831dbaf72edb911",
        "roleArn": "arn:aws:iam::123456789012:role/LoadBalancerManagementforECS"
      }
    }
  ]
}

```

2. Esegui `create-service`.

```
aws ecs create-service --cli-input-json file://canary-service-definition.json
```

Fasi successive

Dopo aver configurato la distribuzione di Canary, completa questi passaggi:

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).
- Monitora il processo di implementazione per assicurarti che segua lo schema canarino:

- La revisione del servizio verde viene creata e aumentata verticalmente
- Una piccola percentuale del traffico (canarie) viene spostata verso la versione verde
- Il sistema attende l'intervallo canarino specificato
- Il traffico rimanente viene spostato tutto in una volta alla versione verde
- Dopo il tempo di incorporamento, la revisione blu viene interrotta

Terminologia di implementazione

I seguenti termini vengono utilizzati in tutta la documentazione sulla distribuzione di Amazon ECS:

Implementazione blu-verde

Una strategia di implementazione che crea un nuovo ambiente (verde) accanto all'ambiente esistente (blu), quindi sposta il traffico dal blu al verde dopo la convalida.

Implementazione Canary

Una strategia di distribuzione che indirizza una piccola percentuale di traffico verso una nuova versione mantenendo la maggior parte sulla versione stabile per la convalida.

Implementazione lineare

Una strategia di distribuzione che sposta gradualmente il traffico dalla vecchia versione alla nuova versione con incrementi uguali nel tempo.

Distribuzione continua

Una strategia di distribuzione che sostituisce le istanze della vecchia versione con le istanze della nuova versione una alla volta.

Set di attività

Una raccolta di attività che eseguono la stessa definizione di attività all'interno di un servizio durante una distribuzione.

Gruppo di destinazione

Un raggruppamento logico di destinazioni che ricevono traffico da un sistema di bilanciamento del carico durante le distribuzioni.

Controller di implementazione

Il metodo utilizzato per distribuire nuove versioni del servizio, come Amazon ECS CodeDeploy, o controller esterni.

Rollback

Il processo di ripristino a una versione precedente dell'applicazione quando vengono rilevati problemi durante la distribuzione.

Implementare 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 dell'integrità di un servizio. Se è necessario aggiornare l'opzione di calcolo, la versione della piattaforma, i dettagli del bilanciatore del carico, la configurazione di rete o la definizione di attività, è necessario 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 si utilizza il tipo di implementazione esterna, tenere conto di quanto segue:

- I tipi di load balancer supportati sono un Application Load Balancer o un Network Load Balancer.
- Fargate oppure i tipi di controller di implementazione `EXTERNAL` non supportano la strategia di pianificazione `DAEMON`.
- L' AutoScaling integrazione delle applicazioni con Amazon ECS supporta solo il servizio Amazon ECS come destinazione. La dimensione scalabile supportata per Amazon ECS è

`ecs:service:DesiredCount`: il numero di attività di un servizio Amazon ECS. Non esiste un'integrazione diretta tra i set di attività Application AutoScaling e Amazon ECS. I set di attività di Amazon ECS calcolano il `ComputedDesiredCount` base al `DesiredCount` di servizio di Amazon ECS.

Flusso di lavoro dell'implementazione esterna

Di seguito è riportato il flusso di lavoro di base per la gestione di un'implementazione 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 implementazione 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

Obbligatorio: sì

Il nome del servizio. Sono consentite fino a 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 implementazione 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

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, consultare [Aggiunta di tag alle risorse Amazon ECS](#).

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

key

Tipo: String

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

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

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

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, consultare [Uso dei tag per la risoluzione dei problemi](#).

propagateTags

Tipo: String

Valori validi: TASK_DEFINITION | SERVICE

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`.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

`schedulingStrategy`

La strategia di pianificazione da utilizzare. I servizi che utilizzano un controller di implementazione 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 si utilizza 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 implementazione 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: String

Valori validi: EC2 | FARGATE | EXTERNAL

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

`platformVersion`

─Tipo: stringa

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

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 sul networking per Fargate, consultare [Opzioni di rete di attività di Amazon ECS per Fargate](#).

serviceRegistries

I dettagli dei registri del servizio di individuazione da assegnare a questo servizio. Per ulteriori informazioni, consultare [Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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 implementazione 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": ""
}
```

3. 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.

CodeDeploy implementazioni blu/verdi per Amazon ECS

Ti consigliamo di utilizzare la blue/green distribuzione Amazon ECS. Per ulteriori informazioni, consulta [Creazione di una distribuzione Amazon ECS blue/green](#).

Il tipo di distribuzione blu/verde utilizza il modello di blue/green distribuzione 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: blue/green

- **Canary:** il traffico viene trasferito 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 lo stesso intervallo di tempo, in minuti, tra ciascun 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.
- **Roll-at-once** — Tutto il traffico viene spostato contemporaneamente dal set di attività originale al set di attività aggiornato.

Di seguito sono riportati CodeDeploy i componenti utilizzati da Amazon ECS quando un servizio utilizza il tipo di blue/green distribuzione:

CodeDeploy applicazione

Una raccolta di CodeDeploy risorse. È costituita da uno o più gruppi di implementazione.

CodeDeploy gruppo di distribuzione

Le impostazioni di implementazione. Comprendono:

- Cluster e servizio Amazon ECS
- Informazioni listener e gruppo di destinazione del bilanciatore del carico
- 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, consultare [Working with Deployment Groups](#) 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 implementazione lineare e canary. È inoltre possibile creare implementazioni lineari e canary personalizzate. Per ulteriori informazioni, consultare [Working with Deployment Configurations](#) nella Guida per l'utente di AWS CodeDeploy .

- CodeDeployDefault.ECSAllAtOnce: sposta tutto il traffico verso il container 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 per cento viene distribuito 15 minuti dopo.

Revisione

Una revisione è il file delle specifiche CodeDeploy dell'applicazione (AppSpec file). 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 blue/green distribuzione, considera quanto segue:

- Quando viene inizialmente creato un servizio Amazon ECS che utilizza il tipo di blue/green distribuzione, viene creato un set di attività 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 verso l'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 di destinazione per il bilanciamento del carico. Il gruppo di destinazione 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 blue/green implementazione e si verifica un evento di scalabilità, il traffico continuerà a variare per 5 minuti. Se il servizio non raggiunge lo stato stazionario entro 5 minuti, CodeDeploy interromperà la distribuzione e la contrassegnerà come fallita.
- Le attività che utilizzano 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 di destinazione per il bilanciatore del carico. Un gruppo di destinazione deve essere il gruppo di destinazione iniziale definito per il bilanciatore del carico quando il servizio Amazon ECS è stato creato. L'unico requisito del secondo gruppo di destinazione è che non può essere associato a un bilanciatore del carico 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 Console di gestione AWS 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, consultare [Amazon ECS_FullAccess](#).

- `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`

 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 utilizza il tipo di blue/green implementazione, devi creare un ruolo IAM (`ecsCodeDeployRole`). Per ulteriori informazioni, consulta [Ruolo CodeDeploy IAM di Amazon ECS](#).

Migra CodeDeploy blue/green deployments to Amazon ECS blue/green le distribuzioni

CodeDeploy blue/green and Amazon ECS blue/green le distribuzioni offrono funzionalità simili, ma differiscono nel modo in cui vengono configurate e gestite.

CodeDeploy panoramica sulla distribuzione in blu e verde

Quando crei un servizio Amazon ECS utilizzando CodeDeploy, tu:

1. Creare un bilanciatore del carico con un listener di produzione e (facoltativamente) uno di test. Ogni listener è configurato con un'unica regola (predefinita) che instrada tutto il traffico verso un singolo gruppo di destinazione (il gruppo di destinazione principale).
2. Creare un servizio Amazon ECS, configurato per utilizzare il listener e il gruppo di destinazione, con il tipo `deploymentController` impostato su `CODE_DEPLOY`. La creazione del servizio comporta la creazione di una serie di attività (blu) registrate con il gruppo di destinazione specificato.
3. Crea un gruppo di CodeDeploy distribuzione (come parte di un' CodeDeploy applicazione) e configuralo con i dettagli del cluster Amazon ECS, il nome del servizio, i listener di load balancer, due gruppi target (il gruppo target principale utilizzato nella regola del listener di produzione e un gruppo target secondario da utilizzare per le attività di sostituzione), un ruolo di servizio (per concedere le CodeDeploy autorizzazioni per manipolare le risorse di Amazon ECS ed Elastic Load Balancing) e vari parametri che controllano comportamento di distribuzione.

Con CodeDeploy, le nuove versioni di un servizio vengono distribuite utilizzando `CreateDeployment()`, specificando il nome CodeDeploy dell'applicazione, il nome del gruppo di distribuzione e un AppSpec file che fornisce dettagli sulla nuova revisione e sugli hook opzionali del ciclo di vita. La CodeDeploy distribuzione crea un set di attività sostitutivo (verde) e ne registra le attività con il gruppo target secondario. Quando diventa integra, è disponibile per i test (facoltativo) e per la produzione. In entrambi i casi, il re-instradamento si ottiene modificando la rispettiva regola del listener in modo tale che punti al gruppo di destinazione secondario associato alla serie di attività verde. Il rollback si ottiene modificando la regola del listener di produzione riportandola al gruppo di destinazione primario.

Panoramica della blue/green distribuzione di Amazon ECS

Con le blue/green distribuzioni di Amazon ECS, la configurazione di distribuzione fa parte del servizio Amazon ECS stesso:

1. È necessario preconfigurare il listener di produzione del bilanciatore del carico con una regola che includa due gruppi di destinazione con pesi pari a 1 e 0.
2. È necessario specificare le seguenti risorse o aggiornare quelle del servizio:
 - L'ARN di questa regola del listener
 - I due gruppi di destinazione
 - Un ruolo IAM per concedere ad Amazon ECS l'autorizzazione a chiamare Elastic Load Balancing APIs
 - Un ruolo IAM facoltativo per eseguire le funzioni Lambda
 - Impostare il tipo `deploymentController` su ECS e `deploymentConfiguration.strategy` su `BLUE_GREEN`. Ciò si traduce nella creazione di un'implementazione del servizio (blu) le cui attività sono registrate nel gruppo di destinazione primario.

Con Amazon ECS blu/verde, viene creata una nuova revisione del servizio chiamando `Amazon ECS UpdateService()`, trasmettendo i dettagli della nuova revisione. L'implementazione del servizio crea nuove attività di revisione del servizio (verde) e le registra con il gruppo di destinazione secondario. Amazon ECS gestisce le operazioni di re-instradamento e rollback modificando i pesi sulla regola del listener.

Differenze di implementazione principali

Sebbene entrambi gli approcci comportino la creazione di una serie iniziale di attività, l'implementazione sottostante è diversa:

- CodeDeploy utilizza un set di attività, mentre Amazon ECS utilizza una revisione del servizio. I set di attività di Amazon ECS sono un costrutto precedente sostituito dalle revisioni e dalle implementazioni dei servizi Amazon ECS. Queste offrono una maggiore visibilità sul processo di implementazione, nonché sulla cronologia di implementazione e revisione del servizio.
- Con CodeDeploy, gli hook del ciclo di vita vengono specificati come parte del AppSpec file a cui viene fornito. `CreateDeployment()` Ciò significa che gli hook possono essere modificati da un'implementazione all'altra. Con Amazon ECS blu/verde, gli hook sono specificati come

parte della configurazione del servizio e qualsiasi aggiornamento richiederebbe una chiamata `UpdateService()`.

- CodeDeploy Sia Amazon ECS che Amazon ECS blue/green utilizzano Lambda per l'implementazione degli hook, ma gli input e gli output previsti sono diversi.

Con CodeDeploy, la funzione deve effettuare una chiamata `PutLifecycleEventHookExecutionStatus()` per restituire lo stato dell'hook, che può essere o. SUCCEEDED FAILED Con Amazon ECS, la risposta di Lambda viene utilizzata per indicare lo stato dell'hook.

- CodeDeploy richiama ogni hook come chiamata una tantum e prevede la restituzione dello stato di esecuzione finale entro un'ora. Gli hook di Amazon ECS sono più flessibili in quanto possono restituire un indicatore `IN_PROGRESS`, che segnala che l'hook deve essere invocato ripetutamente fino a quando non determina un SUCCEEDED o FAILED. Per ulteriori informazioni, consultare [Hook del ciclo di vita per le implementazioni di servizi Amazon ECS](#).

Approcci per la migrazione

Esistono tre approcci principali alla migrazione dalle distribuzioni. CodeDeploy blue/green to Amazon ECS blue/green Ogni approccio dispone di caratteristiche diverse in termini di complessità, rischio, capacità di rollback e potenziale tempo di inattività.

Riutilizza le stesse risorse Elastic Load Balancing utilizzate per CodeDeploy

Aggiorna il servizio Amazon ECS esistente per utilizzare il controller di distribuzione Amazon ECS con strategia di blue/green distribuzione anziché il controller di CodeDeploy distribuzione. Quando si adotta questo approccio, considerare quanto segue:

- La procedura di migrazione è più semplice perché si sta aggiornando il controller di implementazione del servizio Amazon ECS e la strategia di implementazione esistenti.
- Non è previsto un tempo di inattività con una configurazione e una migrazione corrette.
- Un rollback richiede l'annullamento della revisione del servizio.
- Il rischio è elevato perché non esiste una configurazione blu/verde parallela.

Utilizzi lo stesso listener di load balancer e gli stessi gruppi target utilizzati per. CodeDeploy Se lo stai usando CloudFormation, vedi. [Migrazione di un modello CloudFormation CodeDeploy blue/green deployment template to an Amazon ECS blue/green CloudFormation](#)

1. Modifica la regola predefinita degli production/test ascoltatori per includere il gruppo target alternativo e imposta il peso del gruppo target primario su 1 e del gruppo target alternativo su 0.

Infatti CodeDeploy, i listener del load balancer collegato al servizio sono configurati con un'unica regola (predefinita) che indirizza tutto il traffico verso un singolo gruppo di destinazione. Per Amazon ECS blu/verde, i listener del bilanciamento del carico devono essere preconfigurati con una regola che includa i due gruppi di destinazione con pesi. Il gruppo di destinazione principale deve essere ponderato a 1 e il gruppo di destinazione alternativo deve essere ponderato a 0.
2. Aggiornare il servizio Amazon ECS esistente chiamando l'API UpdateService e impostando il parametro deploymentController su ECS e il parametro deploymentStrategy su BLUE_GREEN. Specificate il ARNs gruppo target, il gruppo target alternativo, il listener di produzione e un listener di test opzionale.
3. Verificare che il servizio funzioni come previsto.
4. Elimina la CodeDeploy configurazione per questo servizio Amazon ECS poiché ora utilizzi Amazon ECS blu/verde.

Nuovo servizio con il bilanciamento del carico esistente

Questo approccio utilizza la blue/green strategia per la migrazione.

Quando si adotta questo approccio, considerare quanto segue:

- Le interruzioni sono minime. Si verificano solo durante lo scambio di porte per il bilanciamento del carico elastico.
 - Un rollback richiede il ripristino dello swap della porta per il bilanciamento del carico elastico.
 - Il rischio è basso perché vi sono configurazioni parallele. Pertanto è possibile eseguire il test prima dello spostamento del traffico.
-
1. Lascia intatti gli ascoltatori, i gruppi target e il servizio Amazon ECS per la CodeDeploy configurazione in modo da poter tornare facilmente a questa configurazione, se necessario.
 2. Creare nuovi gruppi di destinazione e nuovi listener (con porte diverse dai listener originali) utilizzando il bilanciamento del carico esistente. Quindi, crea un nuovo servizio Amazon ECS che corrisponda al servizio Amazon ECS esistente, tranne che lo usi ECS come controller di distribuzione, BLUE_GREEN come strategia di distribuzione e trasferisci le regole ARNs per i nuovi gruppi target e i nuovi ascoltatori.

3. Verificare la nuova configurazione inviando manualmente il traffico HTTP al servizio. Se tutto va bene, scambiare le porte dei listener originali con quelle dei nuovi listener per instradare il traffico verso la nuova configurazione.
4. Verifica la nuova configurazione e, se tutto continua a funzionare come previsto, elimina la configurazione. CodeDeploy

Nuovo servizio con un nuovo bilanciatore del carico

Come l'approccio precedente, questo approccio utilizza la blue/green strategia per la migrazione. La differenza fondamentale è che il passaggio dalla CodeDeploy configurazione alla configurazione di Amazon ECS avviene a un livello di proxy inverso al di sopra del blue/green sistema di bilanciamento del carico. Le implementazioni di esempio per il livello di reverse proxy sono Route 53 e. CloudFront

Questo approccio è adatto ai clienti che dispongono già di questo livello di proxy inverso e se tutte le comunicazioni con il servizio avvengono attraverso di esso (ad esempio, nessuna comunicazione diretta a livello di bilanciatore del carico).

Quando si adotta questo approccio, considerare quanto segue:

- Ciò richiede un livello di proxy inverso.
 - La procedura di migrazione è più complessa perché è necessario aggiornare il controller di implementazione del servizio Amazon ECS e la strategia di implementazione esistenti.
 - Le interruzioni sono minime. Si verificano solo durante lo scambio di porte per il bilanciamento del carico elastico.
 - Un rollback richiede l'annullamento delle modifiche alla configurazione del proxy.
 - Il rischio è basso perché vi sono configurazioni parallele. Pertanto è possibile eseguire il test prima dello spostamento del traffico.
1. Non modificare intatta la CodeDeploy configurazione esistente (load balancer, listener, gruppi target, servizio Amazon ECS e CodeDeploy gruppo di distribuzione).
 2. Crea un nuovo sistema di bilanciamento del carico, gruppi target e listener configurati per le distribuzioni di Amazon ECS blue/green .

Configurare le risorse appropriate.

- Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).

- **Network Load Balancer:** per ulteriori informazioni, consultare [Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green](#).
3. Creare un nuovo servizio con ECS come controller di implementazione e BLUE_GREEN come strategia di implementazione, puntando alle nuove risorse del bilanciatore del carico.
 4. Verificare la nuova configurazione testandola tramite il nuovo bilanciatore del carico.
 5. Aggiornare la configurazione del proxy inverso per instradare il traffico verso il nuovo bilanciatore del carico.
 6. Osserva la nuova revisione del servizio e, se tutto continua a funzionare come previsto, elimina la configurazione. CodeDeploy

Fasi successive

Dopo la migrazione alle distribuzioni di Amazon ECS: blue/green

- Aggiorna gli script e le CI/CD pipeline di distribuzione per utilizzare l'API Amazon ECS anziché UpdateService l'API. CodeDeploy CreateDeployment
- Aggiorna il monitoraggio e gli avvisi per tenere traccia delle distribuzioni dei servizi Amazon ECS anziché delle distribuzioni. CodeDeploy
- Considerare l'implementazione di test automatici del nuovo processo di implementazione per garantirne il funzionamento previsto.

Migrazione da una distribuzione di servizi CodeDeploy blue/green to an Amazon ECS blue/green

Utilizzando le blue/green distribuzioni di Amazon ECS, puoi apportare e testare modifiche ai servizi prima di implementarle in un ambiente di produzione.

È necessario creare nuovi hook del ciclo di vita per la distribuzione di Amazon ECS. blue/green

Prerequisiti

Esegui le seguenti operazioni prima di iniziare una distribuzione. blue/green

1. Sostituisci il ruolo Amazon ECS CodeDeploy IAM con le seguenti autorizzazioni.
 - Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).
 - Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).

2. Disattiva CodeDeploy l'automazione. Per ulteriori informazioni, consulta [Lavorare con i gruppi di distribuzione CodeDeploy nella Guida CodeDeploy per l'utente](#).
3. Assicurati di disporre delle seguenti informazioni relative alla CodeDeploy blue/green deployment. You can reuse this information for the Amazon ECS blue/green distribuzione:
 - Il gruppo di destinazione della produzione
 - Il listener di produzione
 - La regola di produzione
 - Il gruppo di destinazione di test

Questo è il gruppo di destinazione per la revisione del servizio verde.

4. Assicurarsi che i gruppi di destinazione dell'Application Load Balancer siano associati correttamente alle regole del listener:
 - Se non si utilizzano i listener di test, entrambi i gruppi di destinazione (produzione e test) devono essere associati alle regole dei listener di produzione.
 - Se si utilizzano i listener di test, un gruppo di destinazione deve essere collegato alle regole dei listener di produzione e l'altro gruppo di destinazione deve essere collegato alle regole dei listener di test.

Se questo requisito non viene soddisfatto, l'implementazione del servizio avrà esito negativo con il seguente errore: `Service deployment rolled back because of invalid networking configuration. Both targetGroup and alternateTargetGroup must be associated with the productionListenerRule or testListenerRule.`

5. Verificare che non vi siano implementazioni di servizi in corso per il servizio. Per ulteriori informazioni, consulta [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).
6. blue/green Le implementazioni di Amazon ECS richiedono che il servizio utilizzi una delle seguenti funzionalità: Configura le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).
 - Network Load Balancer: per ulteriori informazioni, consultare [Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green](#).
 - Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).

7. Decidi se vuoi eseguire le funzioni Lambda per le fasi del ciclo di vita delle fasi della distribuzione di Amazon ECS. blue/green

- Prima dell'aumento verticale
- Dopo l'aumento verticale
- Spostamento del traffico di test
- Dopo lo spostamento del traffico di test
- Spostamento del traffico di produzione
- Dopo lo spostamento del traffico di produzione

Creare funzioni Lambda per ogni fase del ciclo di vita. Per ulteriori informazioni, consultare [Create a Lambda function with the console](#) nella Guida per gli sviluppatori di AWS Lambda .

Per ulteriori informazioni sull'aggiornamento di un controller di implementazione del servizio, consultare [Aggiornare i parametri del servizio Amazon ECS](#).

Procedura

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)

2. Nella pagina Cluster, scegliere il cluster.

Si apre la pagina dei dettagli del cluster.

3. Dalla scheda Servizi, scegliere il servizio.

Si apre la pagina dei dettagli del servizio.

4. Nel banner, scegliere Aggiorna tipo di controller di implementazione.

Si apre la pagina Migrare il tipo di controller di implementazione.

5. Espandere Nuovo, quindi specificare i seguenti parametri

- a. Per Tipo di controller di implementazione, scegliere ECS.
- b. Per Strategia di implementazione, scegliere Blu/verde.
- c. Per tempo di incorporamento, inserire l'ora di esecuzione delle revisioni del servizio blu e verde.
- d. Per eseguire le funzioni Lambda per una fase del ciclo di vita, in Hook del ciclo di vita di implementazione, svolgere le seguenti operazioni per ogni funzione Lambda unica:

i. Scegliere Aggiungi.

Ripetere l'operazione per ogni funzione unica che si desidera eseguire.

ii. Per Funzione Lambda, immettere il nome della funzione.

iii. Per Ruolo, scegliere il ruolo creato nei prerequisiti con le autorizzazioni blu/verdi.

Per ulteriori informazioni, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).

iv. Per Fasi del ciclo di vita, selezionare le fasi eseguite dalla funzione Lambda.

v. (Facoltativo) Per Dettagli dell'hook, inserire una coppia chiave-valore che fornisca informazioni sull'hook.

6. Espandere Bilanciamento del carico e configurare quanto segue:

a. Per Ruolo, scegli il ruolo che hai creato nei prerequisiti con le blue/green autorizzazioni.

Per ulteriori informazioni, consulta [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).

b. Per Listener, scegli il listener di produzione dalla tua distribuzione blu/verde. CodeDeploy

c. Per la regola di produzione, scegli la regola di produzione dalla tua implementazione blu/verde. CodeDeploy

d. Per la regola di test, scegli la regola di test dalla tua implementazione CodeDeploy blu/verde.

e. Per Target group, scegli il gruppo target di produzione dalla tua implementazione CodeDeploy blu/verde.

f. Per Gruppo target alternativo, scegli il gruppo target di test dalla tua CodeDeploy implementazione blu/verde.

7. Scegliere Aggiorna.

Fasi successive

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).
- Monitorare il processo di implementazione per assicurarsi che segua lo schema blu/verde:
 - La revisione del servizio verde viene creata e aumentata verticalmente

- Il traffico di test viene instradato alla revisione verde (se configurata)
- Il traffico di produzione viene spostato alla revisione del servizio verde
- Dopo il tempo di incorporamento, la revisione blu viene interrotta

Migrazione di un modello CloudFormation CodeDeploy blue/green deployment template to an Amazon ECS blue/green CloudFormation

Esegui la migrazione di un CloudFormation modello che utilizza una strategia di distribuzione. CodeDeploy blue/green deployments for Amazon ECS services to one that uses the native Amazon ECS blue/green La migrazione segue l'approccio «Riutilizza le stesse risorse Elastic Load Balancing utilizzate CodeDeploy per». Per ulteriori informazioni, consulta [the section called “Migra CodeDeploy blue/green deployments to Amazon ECS blue/green le distribuzioni”](#).

Modello di origine

Questo modello utilizza `AWS::CodeDeployBlueGreen` transform e `AWS::CodeDeploy::BlueGreen` hook per implementare le blue/green distribuzioni per un servizio Amazon ECS.

Origine

Questo è il CloudFormation modello completo che utilizza la distribuzione blu/verde CodeDeploy . Per ulteriori informazioni, consulta l'[esempio del modello di distribuzione blu/verde](#) nella Guida per l'utente: AWS CloudFormation

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": {
      "Type": "AWS::EC2::VPC::Id"
    },
    "Subnet1": {
      "Type": "AWS::EC2::Subnet::Id"
    },
    "Subnet2": {
      "Type": "AWS::EC2::Subnet::Id"
    }
  },
  "Transform": [
    "AWS::CodeDeployBlueGreen"
  ],
```

```

"Hooks": {
  "CodeDeployBlueGreenHook": {
    "Type": "AWS::CodeDeploy::BlueGreen",
    "Properties": {
      "TrafficRoutingConfig": {
        "Type": "TimeBasedCanary",
        "TimeBasedCanary": {
          "StepPercentage": 15,
          "BakeTimeMins": 5
        }
      },
      "Applications": [
        {
          "Target": {
            "Type": "AWS::ECS::Service",
            "LogicalID": "ECSDemoService"
          },
          "ECSAttributes": {
            "TaskDefinitions": [
              "BlueTaskDefinition",
              "GreenTaskDefinition"
            ],
            "TaskSets": [
              "BlueTaskSet",
              "GreenTaskSet"
            ],
            "TrafficRouting": {
              "ProdTrafficRoute": {
                "Type": "AWS::ElasticLoadBalancingV2::Listener",
                "LogicalID": "ALBListenerProdTraffic"
              },
              "TargetGroups": [
                "ALBTargetGroupBlue",
                "ALBTargetGroupGreen"
              ]
            }
          }
        }
      ]
    }
  }
},
"Resources": {
  "ExampleSecurityGroup": {

```

```

    "Type": "AWS::EC2::SecurityGroup",
    "Properties": {
      "GroupDescription": "Security group for ec2 access",
      "VpcId": {"Ref": "Vpc"},
      "SecurityGroupIngress": [
        {
          "IpProtocol": "tcp",
          "FromPort": 80,
          "ToPort": 80,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 8080,
          "ToPort": 8080,
          "CidrIp": "0.0.0.0/0"
        },
        {
          "IpProtocol": "tcp",
          "FromPort": 22,
          "ToPort": 22,
          "CidrIp": "0.0.0.0/0"
        }
      ]
    }
  },
  "ALBTargetGroupBlue": {
    "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties": {
      "HealthCheckIntervalSeconds": 5,
      "HealthCheckPath": "/",
      "HealthCheckPort": "80",
      "HealthCheckProtocol": "HTTP",
      "HealthCheckTimeoutSeconds": 2,
      "HealthyThresholdCount": 2,
      "Matcher": {
        "HttpCode": "200"
      },
      "Port": 80,
      "Protocol": "HTTP",
      "Tags": [
        {
          "Key": "Group",
          "Value": "Example"
        }
      ]
    }
  }
}

```

```

    }
  ],
  "TargetType": "ip",
  "UnhealthyThresholdCount": 4,
  "VpcId": {"Ref": "Vpc"}
}
},
"ALBTargetGroupGreen": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "HealthCheckIntervalSeconds": 5,
    "HealthCheckPath": "/",
    "HealthCheckPort": "80",
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeoutSeconds": 2,
    "HealthyThresholdCount": 2,
    "Matcher": {
      "HttpCode": "200"
    },
    "Port": 80,
    "Protocol": "HTTP",
    "Tags": [
      {
        "Key": "Group",
        "Value": "Example"
      }
    ],
    "TargetType": "ip",
    "UnhealthyThresholdCount": 4,
    "VpcId": {"Ref": "Vpc"}
  }
},
"ExampleALB": {
  "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties": {
    "Scheme": "internet-facing",
    "SecurityGroups": [
      {"Ref": "ExampleSecurityGroup"}
    ],
    "Subnets": [
      {"Ref": "Subnet1"},
      {"Ref": "Subnet2"}
    ],
    "Tags": [

```

```

        {
            "Key": "Group",
            "Value": "Example"
        }
    ],
    "Type": "application",
    "IpAddressType": "ipv4"
}
},
"ALBListenerProdTraffic": {
    "Type": "AWS::ElasticLoadBalancingV2::Listener",
    "Properties": {
        "DefaultActions": [
            {
                "Type": "forward",
                "ForwardConfig": {
                    "TargetGroups": [
                        {
                            "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
                            "Weight": 1
                        }
                    ]
                }
            }
        ],
        "LoadBalancerArn": {"Ref": "ExampleALB"},
        "Port": 80,
        "Protocol": "HTTP"
    }
},
"ALBListenerProdRule": {
    "Type": "AWS::ElasticLoadBalancingV2::ListenerRule",
    "Properties": {
        "Actions": [
            {
                "Type": "forward",
                "ForwardConfig": {
                    "TargetGroups": [
                        {
                            "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
                            "Weight": 1
                        }
                    ]
                }
            }
        ]
    }
}

```

```

    }
  ],
  "Conditions": [
    {
      "Field": "http-header",
      "HttpHeaderConfig": {
        "HttpHeaderName": "User-Agent",
        "Values": [
          "Mozilla"
        ]
      }
    }
  ],
  "ListenerArn": {"Ref": "ALBListenerProdTraffic"},
  "Priority": 1
}
},
"ECSTaskExecutionRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  },
  "ManagedPolicyArns": [
    "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
  ]
},
"BlueTaskDefinition": {
  "Type": "AWS::ECS::TaskDefinition",
  "Properties": {
    "ExecutionRoleArn": {"Fn::GetAtt": ["ECSTaskExecutionRole", "Arn"]},
    "ContainerDefinitions": [
      {

```

```

        "Name": "DemoApp",
        "Image": "nginxdemos/hello:latest",
        "Essential": true,
        "PortMappings": [
            {
                "HostPort": 80,
                "Protocol": "tcp",
                "ContainerPort": 80
            }
        ]
    },
    "RequiresCompatibilities": [
        "FARGATE"
    ],
    "NetworkMode": "awsvpc",
    "Cpu": "256",
    "Memory": "512",
    "Family": "ecs-demo"
},
"ECSDemoCluster": {
    "Type": "AWS::ECS::Cluster",
    "Properties": {}
},
"ECSDemoService": {
    "Type": "AWS::ECS::Service",
    "Properties": {
        "Cluster": {"Ref": "ECSDemoCluster"},
        "DesiredCount": 1,
        "DeploymentController": {
            "Type": "EXTERNAL"
        }
    }
},
"BlueTaskSet": {
    "Type": "AWS::ECS::TaskSet",
    "Properties": {
        "Cluster": {"Ref": "ECSDemoCluster"},
        "LaunchType": "FARGATE",
        "NetworkConfiguration": {
            "AwsVpcConfiguration": {
                "AssignPublicIp": "ENABLED",
                "SecurityGroups": [

```

```

        {"Ref": "ExampleSecurityGroup"}
    ],
    "Subnets": [
        {"Ref": "Subnet1"},
        {"Ref": "Subnet2"}
    ]
}
},
"PlatformVersion": "1.4.0",
"Scale": {
    "Unit": "PERCENT",
    "Value": 100
},
"Service": {"Ref": "ECSDemoService"},
"TaskDefinition": {"Ref": "BlueTaskDefinition"},
"LoadBalancers": [
    {
        "ContainerName": "DemoApp",
        "ContainerPort": 80,
        "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"}
    }
]
}
},
"PrimaryTaskSet": {
    "Type": "AWS::ECS::PrimaryTaskSet",
    "Properties": {
        "Cluster": {"Ref": "ECSDemoCluster"},
        "Service": {"Ref": "ECSDemoService"},
        "TaskSetId": {"Fn::GetAtt": ["BlueTaskSet", "Id"]}
    }
}
}
}
}

```

Fasi della migrazione

Rimuovi risorse specifiche CodeDeploy

Non sono più necessarie le seguenti proprietà:

- La `AWS::CodeDeployBlueGreen` trasformata
- L'hook `CodeDeployBlueGreenHook`

- Le risorse `GreenTaskDefinition` e `GreenTaskSet` (saranno gestite da Amazon ECS)
- La risorsa `PrimaryTaskSet` (Amazon ECS gestirà internamente i set di attività)

Riconfigurazione del listener del bilanciatore del carico

Modificare la risorsa `ALBListenerProdTraffic` per utilizzare un'azione di inoltro con due gruppi di destinazione:

```
{
  "DefaultActions": [
    {
      "Type": "forward",
      "ForwardConfig": {
        "TargetGroups": [
          {
            "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
            "Weight": 1
          },
          {
            "TargetGroupArn": {"Ref": "ALBTargetGroupGreen"},
            "Weight": 0
          }
        ]
      }
    }
  ]
}
```

Aggiornare le proprietà di implementazione

Aggiornare e aggiungere quanto segue:

- Cambiare la proprietà `DeploymentController` da `EXTERNAL` a `ECS`.
- Aggiungere la proprietà `Strategy` e impostarla su `BLUE_GREEN`.
- Aggiungere la proprietà `BakeTimeInMinutes`.

```
{
  "DeploymentConfiguration": {
    "MaximumPercent": 200,
    "MinimumHealthyPercent": 100,
```

```

    "DeploymentCircuitBreaker": {
      "Enable": true,
      "Rollback": true
    },
    "BakeTimeInMinutes": 5,
    "Strategy": "BLUE_GREEN"
  }
}

```

- Aggiungere la configurazione del bilanciatore del carico al servizio:

```

{
  "LoadBalancers": [
    {
      "ContainerName": "DemoApp",
      "ContainerPort": 80,
      "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
      "AdvancedConfiguration": {
        "AlternateTargetGroupArn": {"Ref": "ALBTargetGroupGreen"},
        "ProductionListenerRule": {"Ref": "ALBListenerProdRule"},
        "RoleArn": {"Fn::GetAtt": ["ECSInfrastructureRoleForLoadBalancers", "Arn"]}
      }
    }
  ]
}

```

- Aggiungere il riferimento alla definizione dell'attività al servizio:

```

{
  "TaskDefinition": {"Ref": "BlueTaskDefinition"}
}

```

Crea il ECSInfrastructure RolePolicyForLoadBalancers ruolo Amazon

Aggiungi un nuovo ruolo IAM che consente ad Amazon ECS di gestire le risorse di bilanciamento del carico. Per ulteriori informazioni, consulta [the section called “Ruolo IAM dell'infrastruttura per i bilanciatori del carico”](#)

Consigli sui test

1. Implementare il modello migrato in un ambiente non di produzione.
2. Verificare che il servizio venga implementato correttamente con la configurazione iniziale.

3. Testare un'implementazione aggiornando la definizione dell'attività e osservando il processo di implementazione blu/verde.
4. Verificare che il traffico si sposti correttamente tra le implementazioni blu e verdi.
5. Testare la funzionalità di rollback forzando un errore di implementazione.

Modello dopo la migrazione

Modello finale

Questo è il CloudFormation modello completo che utilizza una blue/green distribuzione Amazon ECS:

```
{
  "AWSTemplateFormatVersion": "2010-09-09",
  "Parameters": {
    "Vpc": {
      "Type": "AWS::EC2::VPC::Id"
    },
    "Subnet1": {
      "Type": "AWS::EC2::Subnet::Id"
    },
    "Subnet2": {
      "Type": "AWS::EC2::Subnet::Id"
    }
  },
  "Resources": {
    "ExampleSecurityGroup": {
      "Type": "AWS::EC2::SecurityGroup",
      "Properties": {
        "GroupDescription": "Security group for ec2 access",
        "VpcId": {"Ref": "Vpc"},
        "SecurityGroupIngress": [
          {
            "IpProtocol": "tcp",
            "FromPort": 80,
            "ToPort": 80,
            "CidrIp": "0.0.0.0/0"
          },
          {
            "IpProtocol": "tcp",
            "FromPort": 8080,
            "ToPort": 8080,
            "CidrIp": "0.0.0.0/0"
          }
        ]
      }
    }
  }
}
```

```

    },
    {
      "IpProtocol": "tcp",
      "FromPort": 22,
      "ToPort": 22,
      "CidrIp": "0.0.0.0/0"
    }
  ]
}
},
"ALBTargetGroupBlue": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "HealthCheckIntervalSeconds": 5,
    "HealthCheckPath": "/",
    "HealthCheckPort": "80",
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeoutSeconds": 2,
    "HealthyThresholdCount": 2,
    "Matcher": {
      "HttpCode": "200"
    },
    "Port": 80,
    "Protocol": "HTTP",
    "Tags": [
      {
        "Key": "Group",
        "Value": "Example"
      }
    ],
    "TargetType": "ip",
    "UnhealthyThresholdCount": 4,
    "VpcId": {"Ref": "Vpc"}
  }
},
"ALBTargetGroupGreen": {
  "Type": "AWS::ElasticLoadBalancingV2::TargetGroup",
  "Properties": {
    "HealthCheckIntervalSeconds": 5,
    "HealthCheckPath": "/",
    "HealthCheckPort": "80",
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeoutSeconds": 2,
    "HealthyThresholdCount": 2,

```

```

    "Matcher": {
      "HttpCode": "200"
    },
    "Port": 80,
    "Protocol": "HTTP",
    "Tags": [
      {
        "Key": "Group",
        "Value": "Example"
      }
    ],
    "TargetType": "ip",
    "UnhealthyThresholdCount": 4,
    "VpcId": {"Ref": "Vpc"}
  }
},
"ExampleALB": {
  "Type": "AWS::ElasticLoadBalancingV2::LoadBalancer",
  "Properties": {
    "Scheme": "internet-facing",
    "SecurityGroups": [
      {"Ref": "ExampleSecurityGroup"}
    ],
    "Subnets": [
      {"Ref": "Subnet1"},
      {"Ref": "Subnet2"}
    ],
    "Tags": [
      {
        "Key": "Group",
        "Value": "Example"
      }
    ],
    "Type": "application",
    "IpAddressType": "ipv4"
  }
},
"ALBListenerProdTraffic": {
  "Type": "AWS::ElasticLoadBalancingV2::Listener",
  "Properties": {
    "DefaultActions": [
      {
        "Type": "forward",
        "ForwardConfig": {

```

```

        "TargetGroups": [
            {
                "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
                "Weight": 1
            },
            {
                "TargetGroupArn": {"Ref": "ALBTargetGroupGreen"},
                "Weight": 0
            }
        ]
    },
    "LoadBalancerArn": {"Ref": "ExampleALB"},
    "Port": 80,
    "Protocol": "HTTP"
},
"ALBListenerProdRule": {
    "Type": "AWS::ElasticLoadBalancingV2::ListenerRule",
    "Properties": {
        "Actions": [
            {
                "Type": "forward",
                "ForwardConfig": {
                    "TargetGroups": [
                        {
                            "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
                            "Weight": 1
                        },
                        {
                            "TargetGroupArn": {"Ref": "ALBTargetGroupGreen"},
                            "Weight": 0
                        }
                    ]
                }
            }
        ]
    }
},
"Conditions": [
    {
        "Field": "http-header",
        "HttpHeaderConfig": {
            "HttpHeaderName": "User-Agent",
            "Values": [

```

```

        "Mozilla"
      ]
    }
  },
  "ListenerArn": {"Ref": "ALBListenerProdTraffic"},
  "Priority": 1
}
},
"ECSTaskExecutionRole": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "",
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs-tasks.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    },
    "ManagedPolicyArns": [
      "arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy"
    ]
  }
},
"ECSInfrastructureRoleForLoadBalancers": {
  "Type": "AWS::IAM::Role",
  "Properties": {
    "AssumeRolePolicyDocument": {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Sid": "AllowAccessToECSForInfrastructureManagement",
          "Effect": "Allow",
          "Principal": {
            "Service": "ecs.amazonaws.com"
          },
          "Action": "sts:AssumeRole"
        }
      ]
    }
  }
}

```

```

    ]
  },
  "ManagedPolicyArns": [
    "arn:aws:iam::aws:policy/AmazonECSInfrastructureRolePolicyForLoadBalancers"
  ]
}
},
"BlueTaskDefinition": {
  "Type": "AWS::ECS::TaskDefinition",
  "Properties": {
    "ExecutionRoleArn": {"Fn::GetAtt": ["ECSTaskExecutionRole", "Arn"]},
    "ContainerDefinitions": [
      {
        "Name": "DemoApp",
        "Image": "nginxdemos/hello:latest",
        "Essential": true,
        "PortMappings": [
          {
            "HostPort": 80,
            "Protocol": "tcp",
            "ContainerPort": 80
          }
        ]
      }
    ]
  },
  "RequiresCompatibilities": [
    "FARGATE"
  ],
  "NetworkMode": "awsvpc",
  "Cpu": "256",
  "Memory": "512",
  "Family": "ecs-demo"
}
},
"ECSDemoCluster": {
  "Type": "AWS::ECS::Cluster",
  "Properties": {}
},
"ECSDemoService": {
  "Type": "AWS::ECS::Service",
  "Properties": {
    "Cluster": {"Ref": "ECSDemoCluster"},
    "DesiredCount": 1,
    "DeploymentController": {

```



```

    "Type": "ECS"
  },
  "DeploymentConfiguration": {
    "MaximumPercent": 200,
    "MinimumHealthyPercent": 100,
    "DeploymentCircuitBreaker": {
      "Enable": true,
      "Rollback": true
    },
    "BakeTimeInMinutes": 5,
    "Strategy": "BLUE_GREEN"
  },
  "NetworkConfiguration": {
    "AwsvpcConfiguration": {
      "AssignPublicIp": "ENABLED",
      "SecurityGroups": [
        {"Ref": "ExampleSecurityGroup"}
      ],
      "Subnets": [
        {"Ref": "Subnet1"},
        {"Ref": "Subnet2"}
      ]
    }
  },
  "LaunchType": "FARGATE",
  "PlatformVersion": "1.4.0",
  "TaskDefinition": {"Ref": "BlueTaskDefinition"},
  "LoadBalancers": [
    {
      "ContainerName": "DemoApp",
      "ContainerPort": 80,
      "TargetGroupArn": {"Ref": "ALBTargetGroupBlue"},
      "AdvancedConfiguration": {
        "AlternateTargetGroupArn": {"Ref": "ALBTargetGroupGreen"},
        "ProductionListenerRule": {"Ref": "ALBListenerProdRule"},
        "RoleArn": {"Fn::GetAtt": ["ECSInfrastructureRoleForLoadBalancers",
"Arn"]}]
      }
    }
  ]
}
}
}
}

```

```
}
```

Migrazione da una distribuzione di servizi di aggiornamento in sequenza di Amazon ECS CodeDeploy blu/verde a una distribuzione del servizio di aggiornamento continuo di Amazon ECS

Puoi migrare le tue distribuzioni di servizi da una distribuzione CodeDeploy blu/verde a una distribuzione di aggiornamenti continui di Amazon ECS. In questo modo puoi passare dalla CodeDeploy dipendenza all'utilizzo di una distribuzione integrata.

Il pianificatore di servizi di Amazon ECS sostituisce le attività attualmente in esecuzione con nuove attività. Il numero di attività che Amazon ECS aggiunge o rimuove dal servizio durante un aggiornamento continuo è controllato dalla configurazione di distribuzione del servizio.

Prerequisiti

Esegui le seguenti operazioni prima di iniziare una blue/green distribuzione.

1. Non è più necessario il ruolo CodeDeploy IAM di Amazon ECS.
2. Disattiva l' CodeDeploy automazione. Per ulteriori informazioni, consulta [Lavorare con i gruppi di distribuzione CodeDeploy nella Guida CodeDeploy per l'utente](#).
3. Verificare che non vi siano implementazioni di servizi in corso per il servizio. Per ulteriori informazioni, consultare [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).

Per ulteriori informazioni sull'aggiornamento di un controller di implementazione del servizio, consultare [Aggiornare i parametri del servizio Amazon ECS](#).

Procedura

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella pagina Cluster, scegliere il cluster.

Si apre la pagina dei dettagli del cluster.
3. Dalla scheda Servizi, scegliere il servizio.

Si apre la pagina dei dettagli del servizio.
4. Nel banner, scegliere Migra.

Si apre la pagina Aggiorna configurazione di implementazione.

5. Espandere Opzioni di implementazione, quindi specificare i seguenti parametri.
 - a. Per Tipo di controller di implementazione, scegliere ECS.
 - b. Per Strategia di implementazione, scegliere Aggiornamento in sequenza.
 - c. 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, consultare [Deployment configuration](#).
 - d. 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).
6. Espandere Bilanciamento del carico e configurare quanto segue:
 - a. Per Ruolo, scegli il ruolo che hai creato nei prerequisiti con le blue/green autorizzazioni.

Per ulteriori informazioni, consulta [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).
 - b. Per Listener, scegli il listener di produzione dalla tua distribuzione blu/verde. CodeDeploy
 - c. Per Target group, scegli il gruppo target di produzione dalla tua implementazione blu/verde. CodeDeploy
7. Scegliere Aggiorna.

Fasi successive

È necessario aggiornare il servizio affinché venga applicata la modifica. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).

Aggiornare la strategia di implementazione di Amazon ECS

Amazon ECS supporta diverse strategie di implementazione per l'aggiornamento dei servizi. È possibile migrare tra queste strategie in base ai requisiti delle applicazioni. Questo argomento spiega come migrare tra distribuzioni continue e distribuzioni. blue/green

Informazioni sulle strategie di implementazione di Amazon ECS

Prima di migrare da una strategia di implementazione all'altra, è importante capire come funziona ciascuna strategia e le relative differenze principali:

Implementazioni in sequenza

In un'implementazione in sequenza, Amazon ECS sostituisce la versione attuale in esecuzione dell'applicazione con una nuova. Il pianificatore del servizio utilizza i parametri di percentuale minima di attività integre e percentuale massima per determinare la strategia di implementazione.

Le implementazioni in sequenza sono più semplici da configurare ma offrono un minore controllo sul processo di implementazione e sull'instradamento del traffico.

Implementazioni blu/verdi

In una blue/green distribuzione, Amazon ECS crea una nuova versione del servizio (verde) accanto alla versione esistente (blu). Ciò consente di verificare la nuova versione prima di instradarvi il traffico di produzione.

Le implementazioni blu/verdi offrono un maggiore controllo sul processo di implementazione, comprese le funzionalità di spostamento del traffico, di test e rollback.

Best practice

Seguire queste best practice quando si esegue la migrazione tra diverse strategie di implementazione:

- Esegui il test in un ambiente non di produzione: verificare sempre l'aggiornamento in un ambiente non di produzione prima di applicare modifiche ai servizi di produzione.
- Pianifica il rollback: disporre di un piano di rollback nel caso in cui l'aggiornamento non funzioni come previsto.
- Monitoraggio durante la transizione: monitorare attentamente il servizio durante e dopo la migrazione per assicurarsi che continui a funzionare correttamente.
- Aggiornamento della documentazione: aggiornare la documentazione di implementazione in modo tale che rifletta la nuova strategia di implementazione.
- Considera l'impatto sul traffico: informazioni sul modo in cui l'aggiornamento potrebbe influire sul traffico verso il servizio e pianificare di conseguenza.

Aggiornare la strategia di implementazione dall'aggiornamento in sequenza ad Amazon ECS blu/verde

Puoi migrare da una distribuzione di aggiornamento continuo a una distribuzione Amazon ECS blue/green quando desideri apportare e testare modifiche ai servizi prima di implementarle in un ambiente di produzione.

Prerequisiti

Prima di migrare il servizio dalla distribuzione progressiva a quella di blue/green distribuzione, assicurati di disporre di quanto segue:

- Attendere il completamento delle implementazioni attuali.
- Un servizio Amazon ECS esistente che utilizza la strategia di implementazione in sequenza.
- Se si dispone di più revisioni del servizio che servono il traffico, Amazon ECS tenta di consolidare il traffico in un'unica revisione durante la migrazione. Se questa operazione ha esito negativo, potrebbe essere necessario aggiornare manualmente il servizio per utilizzare una singola revisione prima della migrazione.
- Configurazione delle autorizzazioni appropriate.
 - Per informazioni sulle autorizzazioni di bilanciamento del carico elastico, consultare [Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico](#).
 - Per informazioni sulle autorizzazioni Lambda, consultare [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).
- A seconda della configurazione, è necessario eseguire una delle seguenti operazioni:
 - Se il servizio utilizza il bilanciamento del carico elastico, aggiornare il servizio con la nuova “advancedConfiguration” e avviare un'implementazione in sequenza.
 - Se il servizio utilizza Service Connect, aggiornarlo e avviare un'implementazione in sequenza.
 - Se il tuo servizio utilizza sia Elastic Load Balancing che Service Connect, esegui entrambi i passaggi precedenti (puoi usare una sola UpdateService richiesta).
 - Se il servizio non utilizza nessuna delle opzioni precedenti, non è necessaria alcuna operazione aggiuntiva.
- Le blue/green implementazioni di Amazon ECS richiedono che il servizio utilizzi una delle seguenti funzionalità. Configurare le risorse appropriate.
 - Application Load Balancer: per ulteriori informazioni, consultare [Risorse Application Load Balancer per implementazioni blu/green, lineari e canary](#).

- Network Load Balancer: per ulteriori informazioni, consultare [Risorse Network Load Balancer per le implementazioni di Amazon ECS blue/green](#).
- Service Connect: per ulteriori informazioni, consultare [Risorse Service Connect per distribuzioni Amazon ECS blu/green, lineari e canary](#).

Procedura

1. Aprire la console Amazon ECS in <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Cluster.
3. Nella pagina Cluster, scegliere il cluster che contiene il servizio su cui eseguire la migrazione.

Si apre la pagina dei dettagli del cluster.

4. Nella pagina Dettagli cluster, scegliere la scheda Servizi.
5. Scegliere il servizio, poi selezionare Aggiorna.

Si apre la pagina Aggiorna servizio.

6. Espandere Opzioni di implementazione, poi svolgere le seguenti operazioni:
7. Per Strategia di implementazione, scegliere Blu/verde.
8. Configura le impostazioni di distribuzione blue/green :
 - a. Per Tempo di incorporamento, inserire il numero di minuti in cui entrambe le revisioni del servizio blu e verde verranno eseguite contemporaneamente prima che la revisione blu venga terminata.

Ciò consente di avere tempo per la verifica e il test.

- b. (Facoltativo) Configurare le funzioni Lambda in fasi specifiche dell'implementazione. In Hook del ciclo di vita di implementazione, configurare le funzioni Lambda per le seguenti fasi:
 - Prima dell'aumento verticale: viene eseguita prima di scalare la revisione del servizio verde
 - Dopo l'aumento verticale: viene eseguita prima di scalare la revisione del servizio verde
 - Spostamento del traffico di test: viene eseguito durante l'instradamento del traffico di test alla revisione del servizio verde.
 - Spostamento del traffico dopo il test: viene eseguito dopo che il traffico di test è stato instradato alla revisione del servizio verde.

- Spostamento del traffico di produzione: viene eseguito durante l'instradamento del traffico di produzione alla revisione del servizio verde.
- Spostamento del traffico dopo la produzione: viene eseguito dopo che il traffico di produzione è stato instradato alla revisione del servizio verde.

Per aggiungere un hook del ciclo di vita:

- a. Scegliere Aggiungi.
- b. Per Funzione Lambda, immettere il nome della funzione o ARN.
- c. Per Ruolo, scegliere il ruolo IAM che ha il permesso di invocare la funzione Lambda.
- d. Per Fasi del ciclo di vita, selezionare le fasi in cui deve essere eseguita la funzione Lambda.
- e. (Facoltativo) Per Dettagli dell'hook, inserire una coppia chiave-valore per fornire informazioni aggiuntive all'hook.

9. Configurazione delle impostazioni del bilanciatore del carico:

- a. In bilanciatore del carico, verificare che il servizio sia configurato per utilizzare un bilanciatore del carico.
- b. Per Gruppo di destinazione, scegliere il gruppo di destinazione primario per l'ambiente di produzione (blu).
- c. Per Gruppo di destinazione alternativo, scegliere il gruppo di destinazione per l'ambiente di test (verde).
- d. Per Regola del listener di produzione, scegliere la regola del listener per l'instradamento del traffico di produzione.
- e. (Facoltativo) Per Regola del listener di test, scegliere una regola del listener per instradare il traffico di test verso l'ambiente verde.
- f. Per Ruolo, scegliere il ruolo IAM che consente ad Amazon ECS di gestire il bilanciatore del carico.

10. Rivedere le modifiche alla configurazione, quindi scegliere Aggiorna.

Fasi successive

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).

- Monitora il processo di distribuzione per assicurarti che segua lo blue/green schema:
 - La revisione del servizio verde viene creata e aumentata verticalmente
 - Il traffico di test viene instradato alla revisione verde (se configurata)
 - Il traffico di produzione viene spostato alla revisione del servizio verde
 - Dopo il tempo di incorporamento, la revisione blu viene interrotta

Aggiornamento della strategia di distribuzione da Amazon ECS blue/green all'aggiornamento progressivo

Puoi migrare una blue/green distribuzione a una distribuzione con aggiornamento progressivo.

Quando si esegue la migrazione a implementazioni in sequenza, tenere presente le seguenti considerazioni:

- Gestione del traffico: con le implementazioni in sequenza, le nuove attività iniziano a ricevere traffico non appena superano i controlli dell'integrità. Non è prevista una fase di test separata come per le blue/green distribuzioni.
- Efficienza delle risorse: le implementazioni in sequenza in genere utilizzano meno risorse rispetto alle blue/green distribuzioni perché sostituiscono le attività in modo incrementale anziché creare un ambiente duplicato completo.
- Complessità del rollback: le implementazioni continue rendono i rollback più complessi rispetto alle implementazioni blue/green. Se è necessario eseguire il rollback, occorre avviare una nuova implementazione con la definizione dell'attività precedente.
- Velocità di implementazione: il completamento delle implementazioni in sequenza può richiedere più tempo rispetto alle blue/green implementazioni, soprattutto per i servizi con molte attività.
- Configurazione del bilanciatore del carico: la configurazione del bilanciatore del carico esistente continuerà a funzionare con le implementazioni in sequenza, ma il comportamento di spostamento del traffico sarà diverso.

Prerequisiti

Prima di migrare il servizio da implementazioni continue, blue/green assicurati di disporre di quanto segue:

- Un servizio Amazon ECS esistente che utilizza la strategia di blue/green distribuzione

- Nessuna implementazione in corso per il servizio (attendere il completamento delle implementazioni attuali)
- Una chiara comprensione del comportamento del servizio con le implementazioni in sequenza

Note

Non è possibile eseguire la migrazione di un servizio all'implementazione in sequenza se dispone di un'implementazione in corso. Attendere il completamento delle implementazioni attuali prima di procedere.

Procedura di migrazione

Segui questi passaggi per migrare il tuo servizio Amazon ECS blue/green da distribuzioni continue:

1. Aprire la console Amazon ECS in <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Cluster.
3. Nella pagina Cluster, scegliere il cluster che contiene il servizio su cui eseguire la migrazione.
4. Nella pagina Dettagli cluster, scegliere la scheda Servizi.
5. Selezionare il servizio da migrare, quindi scegliere Aggiorna.
6. Nella pagina Aggiorna servizio, andare alla sezione Opzioni di implementazione ed espanderla se necessario.
7. Per Strategia di implementazione, scegliere Aggiornamento in sequenza.
8. Configurare le impostazioni di implementazione in sequenza:
 - a. Per Percentuale minima di integrità, inserire la percentuale minima di attività che devono rimanere nello stato RUNNING durante un'implementazione. Questo valore è specificato come una percentuale del numero desiderato di attività per il servizio.
 - b. Per Percentuale massima, inserire la percentuale massima di attività consentite nello stato RUNNING o PENDING durante un'implementazione. Questo valore è specificato come una percentuale del numero desiderato di attività per il servizio.
9. Facoltativo: in Rilevamento degli errori di implementazione, configurare il modo in cui Amazon ECS rileva e gestisce gli errori di implementazione.
 - a. Per abilitare l'interruttore automatico di implementazione, scegliere Usa l'interruttore automatico di implementazione.

- b. Per effettuare il rollback automatico delle implementazioni non riuscite, scegliere Rollback in caso di errore.
10. Rivedere le modifiche alla configurazione, quindi scegliere Aggiorna per salvare le modifiche ed eseguire la migrazione del servizio all'implementazione in sequenza.

Amazon ECS aggiornerà la configurazione del servizio per utilizzare la strategia di implementazione in sequenza. Al prossimo aggiornamento del servizio, questo utilizzerà il processo di implementazione in sequenza.

Note

Quando esegui la migrazione blue/green da una distribuzione progressiva, Amazon ECS gestisce la transizione mediante:

1. Identificazione dell'attuale revisione attiva del servizio che serve il traffico.
2. Mantenendo la configurazione esistente del bilanciatore del carico ma modificando il modo in cui vengono gestite le nuove implementazioni.
3. Preparazione del servizio per le future implementazioni in sequenza.

Fasi successive

- Aggiornare il servizio per avviare l'implementazione. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).

Risoluzione dei problemi relativi agli aggiornamenti della strategia di implementazione di Amazon ECS

Questa sezione fornisce soluzioni ai problemi più comuni che potrebbero verificarsi durante la migrazione delle strategie di implementazione.

Più revisioni del servizio o set di attività

I seguenti problemi riguardano la disponibilità di più revisioni dei servizi per un'implementazione.

Set di più attività durante l'aggiornamento del controller di implementazione ECS

Messaggio di errore: Updating the deployment controller is not supported when there are multiple tasksets in the service. Please ensure your service has only one taskset and try again.

Soluzione: questo errore si verifica quando si tenta di modificare il tipo di controller di implementazione di un servizio con più set di attività attivi. Per risolvere questo problema per il controller di implementazione CODE_DEPLOY o EXTERNAL:

1. Controllare i set di attività attuali:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].taskSets"
```

2. Attendere il completamento delle implementazioni in corso.
3. Forzare una nuova implementazione per eliminare i set di attività:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --force-new-deployment
```

4. Se necessario, eliminare manualmente i set di attività aggiuntivi:

```
aws ecs delete-task-set --cluster your-cluster-name --service your-service-name --task-set task-set-id
```

5. Quando rimane un solo set di attività, riprovare ad aggiornare il controller di implementazione.

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Set di attività primarie mancante durante l'aggiornamento del controller di implementazione ECS

Messaggio di errore: Updating the deployment controller requires a primary taskset in the service. Please ensure your service has a primary taskset and try again.

Soluzione: questo errore si verifica quando si tenta di modificare il tipo di controller di implementazione di un servizio che non dispone di un set di attività primarie. Per risolvere il problema:

1. Verificare lo stato del servizio e i set di attività. Se nel servizio esiste un set di attività, deve essere contrassegnato come ACTIVE.

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].taskSets[*].[status,id]"
```

Se non ci sono set di attività nello stato ACTIVE, effettuare la migrazione dell'implementazione. Per ulteriori informazioni, consultare [Migration approaches](#).

2. Se il servizio non dispone di attività in esecuzione, implementare almeno un'attività aggiornando il servizio:

```
aws ecs update-service-primary-task-set --cluster your-cluster-name --service your-service-name --primary-task-set your-taskset-id
```

Ciò contrassegnerà l'attività (precedentemente ACTIVE) impostata nel servizio come PRIMARY.

3. Attendere che l'attività raggiunga uno stato di esecuzione stabile. È possibile controllare lo stato con:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].deployments"
```

4. Dopo che il servizio dispone di un set di attività primario con attività in esecuzione, riprovare ad aggiornare il controller di implementazione.

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Mancata corrispondenza tra il tipo di rilevamento degli errori di implementazione e il controller di implementazione

I seguenti problemi si riferiscono a una mancata corrispondenza tra il tipo di rilevamento degli errori di implementazione e il controller di implementazione.

Interruttore automatico di implementazione con controller non ECS

Messaggio di errore: Deployment circuit breaker feature is only supported with ECS deployment controller. Update to ECS deployment controller and try again.

Soluzione: questo errore si verifica quando si tenta di abilitare la funzionalità di interruttore automatico di implementazione su un servizio che non utilizza il controller di implementazione ECS. L'interruttore automatico di implementazione è compatibile solo con il controller di implementazione ECS.

1. Verificare il controller di implementazione attuale del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].deploymentController"
```

2. Aggiornare il servizio per utilizzare il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --deployment-controller type=ECS
```

3. Dopo che il servizio sta utilizzando il controller di implementazione ECS, abilitare l'interruttore automatico di implementazione:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --deployment-configuration "deploymentCircuitBreaker={enable=true,rollback=true}"
```

Per ulteriori informazioni, consultare [the section called “In che modo l'interruttore di implementazione rileva i guasti”](#).

Rollback basato sugli allarmi con controller non ECS

Messaggio di errore: Alarm based rollback feature is only supported with ECS deployment controller. Update to ECS deployment controller and try again.

Soluzione: questo errore si verifica quando si tenta di configurare il rollback basato sugli allarmi su un servizio che non utilizza il controller di implementazione ECS. La funzionalità di rollback basata sugli allarmi è compatibile solo con il controller di implementazione ECS.

1. Verificare il controller di implementazione attuale del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].deploymentController"
```

2. Aggiornare il servizio per utilizzare il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --deployment-controller type=ECS
```

3. Dopo che il servizio sta utilizzando il controller di implementazione ECS, configurare il rollback basato sugli allarmi:

```
aws ecs update-service --cluster your-cluster-name --services your-service-name --deployment-configuration "alarms={alarmNames=[your-alarm-name],enable=true,rollback=true}"
```

Per ulteriori informazioni, consultare [the section called “In che modo gli CloudWatch allarmi rilevano gli errori di distribuzione”](#).

Mancata corrispondenza tra Service Connect e il controller di implementazione

I seguenti problemi riguardano una mancata corrispondenza tra Service Connect e il controller di implementazione.

Controller EXTERNAL con Service Connect

Messaggio di errore: The EXTERNAL deployment controller type is not supported for services using Service Connect.

Soluzione: questo errore si verifica quando si tenta di utilizzare il controller di implementazione EXTERNAL con un servizio che ha Service Connect abilitato. Il controller EXTERNAL non è compatibile con Service Connect.

1. Verificare se il servizio dispone di Service Connect abilitato:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].serviceConnectConfiguration"
```

2. Se è necessario il controller di implementazione EXTERNAL, disabilitare Service Connect aggiornando il servizio:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name  
--service-connect-configuration "{}"
```

3. In alternativa, se è necessario utilizzare Service Connect, utilizzare invece il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name  
--deployment-controller type=ECS
```

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Service Connect con controller non ECS

Messaggio di errore: Service Connect feature is only supported with ECS (rolling update) deployment controller. Update to ECS deployment controller and try again.

Soluzione: questo errore si verifica quando si tenta di abilitare Service Connect su un servizio che non utilizza il controller di implementazione ECS. La funzionalità di Service Connect è compatibile solo con il controller di implementazione ECS.

1. Verificare il controller di implementazione attuale del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name  
--query "services[0].deploymentController"
```

2. Aggiornare il servizio per utilizzare il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name  
--deployment-controller type=ECS
```

3. Una volta che il servizio utilizza il controller di implementazione ECS, abilitare Service Connect:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name  
--service-connect-configuration "enabled=true,namespace=your-namespace"
```

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Mancata corrispondenza tra il tipo di controller e la strategia di pianificazione

I seguenti problemi riguardano una mancata corrispondenza tra il tipo di controller e la strategia di pianificazione.

Controller CODE_DEPLOY con strategia di pianificazione DAEMON

Messaggio di errore: The CODE_DEPLOY deployment controller type is not supported for services using the DAEMON scheduling strategy.

Soluzione: questo errore si verifica quando si tenta di utilizzare il controller di implementazione CODE_DEPLOY con un servizio che utilizza la strategia di pianificazione DAEMON. Il controller CODE_DEPLOY è compatibile solo con la strategia di pianificazione REPLICA.

1. Verificare l'attuale strategia di pianificazione del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].schedulingStrategy"
```

2. Se hai bisogno di blue/green implementazioni, modifica il servizio per utilizzare la strategia di pianificazione: REPLICA

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --scheduling-strategy REPLICA
```

3. In alternativa, se è necessario utilizzare la strategia di pianificazione DAEMON, utilizzare invece il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --deployment-controller type=ECS
```

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Controller EXTERNAL con strategia di pianificazione DAEMON

Messaggio di errore: The EXTERNAL deployment controller type is not supported for services using the DAEMON scheduling strategy.

Soluzione: questo errore si verifica quando si tenta di utilizzare il controller di implementazione ESTERNO con un servizio ECS che utilizza la strategia di pianificazione DAEMON. Il controller EXTERNAL è compatibile solo con la strategia di pianificazione REPLICA.

1. Verificare l'attuale strategia di pianificazione del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].schedulingStrategy"
```

2. Se è necessario utilizzare il controller di implementazione EXTERNAL, cambiare il servizio per utilizzare la strategia di pianificazione REPLICA:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --scheduling-strategy REPLICA
```

3. In alternativa, se è necessario utilizzare la strategia di pianificazione DAEMON, utilizzare invece il controller di implementazione ECS:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --deployment-controller type=ECS
```

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Registri di servizio con tipo di avvio esterno

Messaggio di errore: Service registries are not supported for external launch type.

Soluzione: questo errore avviene quando si tenta di configurare il rilevamento servizi (registri dei servizi) per un servizio che utilizza il tipo di avvio EXTERNAL. Il rilevamento servizi non è compatibile con il tipo di avvio EXTERNAL.

1. Controllare il tipo di avvio attuale del servizio:

```
aws ecs describe-services --cluster your-cluster-name --services your-service-name --query "services[0].launchType"
```

2. Se è necessario il rilevamento servizi, modificare il servizio in modo da utilizzare il tipo di avvio EC2 o FARGATE:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --launch-type FARGATE
```

3. In alternativa, se è necessario utilizzare il tipo di avvio EXTERNAL, rimuovere la configurazione del registro del servizio:

```
aws ecs update-service --cluster your-cluster-name --service your-service-name --service-registries "[]"
```

Per ulteriori informazioni, consultare [the section called “Controller e strategie di implementazione dei servizi”](#).

Ripristinare un aggiornamento del controller di implementazione

Se si decide di tornare al controller di implementazione precedente, è possibile eseguire una delle seguenti operazioni:

- Se lo hai utilizzato CloudFormation, puoi utilizzare il modello precedente per creare un nuovo stack. Per ulteriori informazioni, consultare [Create a stack from](#) nella Guida per l'utente di CloudFormation.
- Se hai utilizzato la console Amazon ECS o la AWS CLI, puoi aggiornare il servizio. Per ulteriori informazioni, consulta [the section called “Aggiornamento di un servizio”](#).

Se si utilizza il comando `update-service`, usare l'opzione `--deployment-controller` e impostarla sul controller di implementazione precedente.

Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS

Le implementazioni dei servizi forniscono una visione completa delle implementazioni. Le implementazioni dei servizi forniscono le informazioni seguenti sul servizio:

- La configurazione del carico di lavoro attualmente implementata (la revisione del servizio di origine)
- La configurazione del carico di lavoro in fase di implementazione (la revisione del servizio di destinazione)
- Lo stato dell'implementazione
- 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 implementazione del servizio, consultare [Proprietà incluse in un'implementazione del servizio Amazon ECS](#).

Le implementazioni dei servizi sono di sola lettura e ciascuna dispone di un ID univoco.

Vi sono tre fasi di implementazione del servizio:

Stage	Definizione	Stati associati
Pending (In attesa)	È stata creata un'implementazione del servizio, ma non è stata avviata	IN ATTESA
Continua	Un'implementazione del servizio è in corso	• IN_PROGRESS • STOP_REQUESTED • ROLLBACK_REQUESTED • ROLLBACK_IN_PROGRESS

Stage	Definizione	Stati associati
Completato	Un'implementazione del 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 l'implementazione del servizio ed esaminare gli eventi del servizio.

È possibile visualizzare la cronologia di 90 giorni più recente per le implementazioni create il o dopo il 25 ottobre 2024 utilizzando la console, l'API e la AWS CLI.

È possibile interrompere un'implementazione non completata. Per ulteriori informazioni, consultare [Interruzione delle implementazioni dei servizi Amazon ECS](#).

Ciclo di vita dell'implementazione dei servizi

Amazon ECS crea automaticamente una nuova implementazione di servizi quando si verifica una delle seguenti operazioni:

- Un utente crea un servizio.
- Un utente aggiorna il servizio e utilizza l'opzione Forza nuova implementazione.
- Un utente aggiorna una o più proprietà del servizio che richiedono un'implementazione.

Mentre un'implementazione è in corso, Amazon ECS aggiorna le seguenti proprietà di implementazione del servizio per riflettere i progressi dell'implementazione 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 dell'implementazione. Ad esempio, se sono state avviate attività indipendentemente dall'implementazione del servizio, tali attività non vengono incluse nel conteggio delle attività in esecuzione per la revisione del servizio.

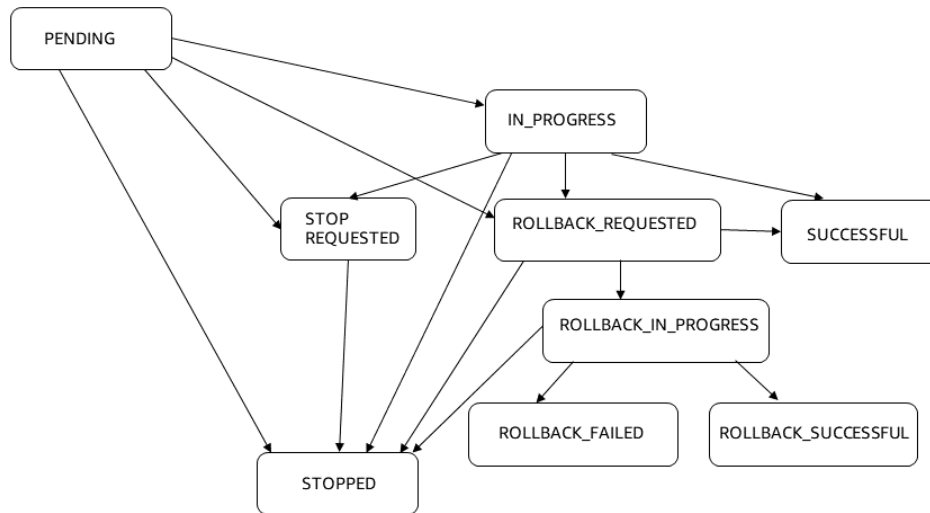
- Rilevamento di guasti all'interruttore automatico:
 - Il numero di attività non riuscite
- CloudWatch rilevamento dei guasti degli allarmi
 - Gli allarmi che sono attivi
- Informazioni di rollback:
 - L'ora di inizio
 - Il motivo del rollback
 - L'ARN della revisione del servizio utilizzata per il rollback
- Il motivo dello stato

Amazon ECS elimina l'implementazione del servizio quando viene eliminato uno.

Stati di implementazione del servizio

Un'implementazione di servizio inizia nello stato PENDING.

La figura seguente mostra gli stati di implementazione del servizio che possono verificarsi dopo lo stato PENDING: IN_PROGRESS, ROLLBACK_REQUESTED, SUCCESSFUL, STOP_REQUESTED, ROLLBACK_IN_PROGRESS, ROLLBACK_FAILED, ROLLBACK_SUCCESSFUL e STOPPED.



Le seguenti informazioni forniscono dettagli sugli stati di implementazione del servizio:

- **PENDING:** è stata creata una implementazione del servizio, 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 implementazione del servizio passa a **STOP_REQUESTED** quando si verifica una delle seguenti condizioni:
 - Un utente avvia una nuova implementazione del servizio.
 - L'opzione di rollback non è utilizzata per il meccanismo di rilevamento degli errori (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 implementazione 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 `SUCCESSFUL`, `ROLLBACK_IN_PROGRESS` e `STOPPED`.

- **SUCCESSFUL**: lo stato di implementazione del servizio passa a `SUCCESSFUL` quando l'implementazione del servizio viene completata correttamente.
- **ROLLBACK_IN_PROGRESS**: lo stato di implementazione del servizio passa a `ROLLBACK_IN_PROGRESS` quando l'opzione di rollback è in uso per il meccanismo di rilevamento dei guasti (l'interruttore automatico o basato sugli allarmi) e il servizio ha esito negativo.

Lo stato passa a `ROLLBACK_SUCCESSFUL` o `ROLLBACK_FAILED`.

Proprietà incluse in un'implementazione del servizio Amazon ECS

Le seguenti proprietà sono incluse in un'implementazione di servizi.

Proprietà	Description	
ARN dell'implementazione dei servizi	L'ARN del servizio di implementazione.	
ARN del servizio	L'ARN del servizio per quest'implementazione del servizio.	
ARN del cluster	L'ARN per il cluster che ospita il servizio.	
Ora di creazione dell'implementazione del servizio	L'ora in cui è stata creata l'implementazione del servizio.	
Ora di inizio dell'implementazione del servizio	L'ora in cui è iniziata l'implementazione del servizio.	
Ora di fine dell'implementazione del servizio	L'ora in cui è terminata l'implementazione del servizio.	
Ora di interruzione dell'implementazione del servizio	L'ora in cui è stata interrotta l'implementazione del servizio.	

Proprietà	Description	
Ora di aggiornamento dell'implementazione del servizio	L'ora dell'ultimo aggiornamento dell'implementazione del servizio.	
Revisioni del servizio di origine	Le revisioni del servizio attualmente in esecuzione. Per informazioni sulle proprietà incluse, consultare e Proprietà incluse in una revisione del servizio Amazon ECS.	

Proprietà	Description	
Configurazione dell'implementazione	I parametri di implementazione che includono la configurazione dell'interruttore automatico, gli allarmi che determinano: • 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. • Il limite superiore del numero di processi che devono essere in esecuzione e 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 un servizio. • La configurazione dell'interruttore automatico. • Gli allarmi utilizzati per il rilevamento dei guasti.	

Proprietà	Description	
Revisione del servizio di destinazione	La revisione del servizio da implementare. Una volta completata correttamente l'implementazione, la revisione del servizio di destinazione è la revisione del servizio in esecuzione.	
Stato dell'implementazione del servizio	Lo stato dell'implementazione del servizio. I valori validi sono PENDING, SUCCESSFUL, STOPPED, STOP_REQUESTED, STOP_IN_PROGRESS, IN_PROGRESS, ROLLBACK_IN_PROGRESS, ROLLBACK_SUCCESSFUL e ROLLBACK_FAILED.	
Informazioni sullo stato dell'implementazione del servizio	Informazioni sul motivo per cui l'implementazione del servizio è nello stato attuale. Ad esempio, l'interruttore automatico ha rilevato un guasto.	
Informazioni di rollback	Le opzioni di rollback utilizzate dall'implementazione del servizio in caso di errore dell'implementazione.	

Proprietà	Description	
Opzioni di interruttore automatico di implementazione del servizio	L'interruttore automatico che determina l'implementazione del servizio ha avuto esito negativo.	
CloudWatch allarmi per l'implementazione del servizio	Gli CloudWatch allarmi che determinano quando l'implementazione di un servizio fallisce.	

Autorizzazioni richieste per visualizzare le implementazioni del servizio di Amazon ECS

Quando si segue la best practice della concessione dei privilegi minimi, è necessario aggiungere ulteriori autorizzazioni per visualizzare le implementazioni del servizio nella console.

È necessario accedere alle seguenti azioni:

- `ListServiceDeployments`
- `DescribeServiceDeployments`
- `DescribeServiceRevisions`

È necessario accedere alle seguenti risorse:

- Servizio
- Implementazioni del servizio
- Revisione del servizio

La seguente policy di esempio contiene le autorizzazioni necessarie e limita le operazioni a un servizio specifico.

Sostituire `account`, `cluster-name` e `service-name` con i propri valori.

JSON

```
{
  "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/*"
      ]
    }
  ]
}
```

Visualizzare le implementazioni del servizio di Amazon ECS

È possibile vedere la cronologia di 90 giorni più recente per le implementazioni create il o dopo il 25 ottobre 2024. Le implementazioni del servizio possono avere uno dei seguenti stati:

- In corso
- Pending (In attesa)
- Completato

È possibile utilizzare queste informazioni per determinare se è necessario aggiornare la modalità di implementazione del servizio o la revisione del servizio. Per informazioni sulle proprietà incluse, consultare [Proprietà incluse in un'implementazione del servizio Amazon ECS](#).

Prima di iniziare, configurare le autorizzazioni richieste per visualizzare le implementazioni dei servizi. Per ulteriori informazioni, consulta [Autorizzazioni richieste per visualizzare le implementazioni del servizio di 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 Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegliere Implementazioni.
5. Scegli l'implementazione del servizio da visualizzare.

Per visualizzare le implementazioni del servizio per questo tipo di implementazione	Esegui questa operazione	
Implementazione continua	In Implementazioni in corso, scegliere l'ID di implementazione.	
Ultima implementazione	In Ultima implementazione, scegliere l'ID di implementazione.	
Implementazioni completate	In Implementazioni del servizio, scegliere l'ID di implementazione.	

Si apre la pagina dei dettagli dell'implementazione del servizio.

6. (Facoltativo) Confrontare le revisioni del servizio per visualizzare le differenze.

In Revisioni del servizio, scegliere Confronta revisioni, quindi selezionare 2 revisioni da confrontare.

Le revisioni del servizio vengono visualizzate side-by-side con le differenze evidenziate.

AWS CLI

1. Eseguire `list-service-deployments` per recuperare l'ARN di implementazione del servizio.

Sostituire le variabili con i propri valori.

```
aws ecs list-service-deployments --cluster cluster-name --service service-name
```

Nota la versione `serviceDeploymentArn` per la 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`. Usare il `serviceDeploymentArn` restituito da `list-service-deployments`.

Sostituire le variabili con i propri valori.

```
aws ecs describe-service-deployments --service-deployment-arns
arn:aws:ecs:region:123456789012:service-deployment/cluster-name/service-name/NCWGC2ZR-taawPAYrIaU5
```

Fasi successive

È possibile visualizzare i dettagli per le revisioni del servizio nell'implementazione. Per ulteriori informazioni, consultare [Visualizzazione dei dettagli della revisione del servizio di Amazon ECS](#)

Revisioni del servizio 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 tentando di distribuire nella revisione del servizio.

Le revisioni del servizio sono di sola lettura e dispongono di identificatori univoci. Per informazioni sulle proprietà incluse, consultare [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 implementata (origine) con quella che verrà implementata (destinazione).
- Quando si utilizza l'opzione di rollback per un'implementazione del servizio, Amazon ECS ripristina automaticamente l'implementazione del servizio all'ultima revisione del servizio implementata 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 si crea un servizio o si aggiorna una proprietà del servizio che avvia un'implementazione.

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 non è modificabile.

Amazon ECS elimina la revisione del servizio quando viene eliminato un servizio.

È possibile visualizzare le revisioni del servizio 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	Description	
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 di attività utilizzato per le attività del servizio.	
Registri del servizio	I dettagli dei registri del servizio utilizzati per il rilevamento servizi. • Il nome del container Il valore che indica il nome del container da utilizzare 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 anche specificato nella definizione dell'attività. • La porta Il valore della porta utilizzato o se l'individuazione dei servizi ha specificato un record SRV. • L'ARN del registro	

Risorsa	Description	
	L'Amazon Resource Name (ARN) del registro del servizio.	
fornitori di capacità	I dettagli della strategia del provider di capacità. • Il nome del provider di capacità Il nome del provider di capacità usato per il servizio. • Il peso del provider di capacità La percentuale relativa del numero totale di attività avviate che devono utilizzare e il provider di capacità specificato. • La base del provider di capacità Il numero minimo di attività da eseguire su un determinato provider di capacità.	
Immagini di container	I dettagli sulle immagini del container. • Il nome del container • Il digest delle immagini • L'immagine del container	

Risorsa	Description	
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 lancio	L'opzione di calcolo utilizzata per il servizio.	
Proprietà specifiche di Fargate	Quando si utilizza Fargate, queste sono informazioni sulla versione di Fargate. • La versione della piattaforma e la famiglia della piattaforma • Archiviazione temporanea • La quantità di archiviazione temporanea da allocare per l'implementazione. • La chiave KMS da utilizzare per la crittografia di archiviazione.	

Risorsa	Description	
Volumi di Amazon EBS configurati all'implementazione	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 filesystem Linux per il volume. • IOPS Il numero di I/O operazioni al secondo (IOPS). • Chiave KMS • Ruolo IAM L'ARN o il ruolo IAM da associare a questo volume. • Dimensione La dimensione del volume in GiB. • ID dell'istantanea L'ID dello snapshot del volume. • Specifiche del tag La configurazione dei tag per il volume.	

Risorsa	Description	
	- Throughput Il throughput per il provisioning del volume. - Tipo di volume Tipo di volume.	
Service Connect	La configurazione di Service Connect. - L'indicazione se Service Connect è in uso - Il namespace - L'elenco dei servizi interconnessi - La configurazione di registrazione di Service Connect	
Bilanciatori del carico del servizio	I bilanciatori del carico che instradano il traffico del servizio. - Il nome - Il nome del container - La porta del container - L'ARN del gruppo di destinazione	
Monitoraggio del runtime	Indica se il monitoraggio del runtime è attivo.	
Data di creazione	La data di creazione della revisione del servizio.	

Risorsa	Description	
Reticolo in VPC	La configurazione di VPC Lattice per la revisione del servizio.	

Visualizzazione dei dettagli della revisione del servizio di Amazon ECS

È possibile visualizzare informazioni sui seguenti tipi di revisione del servizio creati il o dopo il 25 ottobre 2024:

- Origine: la configurazione del carico di lavoro attualmente implementata
- Destinazione: la configurazione del carico di lavoro in fase di implementazione

Amazon ECS Console

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegliere Implementazioni.
5. Scegliere la revisione del servizio da visualizzare.

Per visualizzare le revisioni del servizio per questo tipo di implementazione	Esegui questa operazione	
Implementazioni continue	In Implementazioni in corso, procedere come segue: • Per visualizzare la revisione del servizio di origine, in Revisioni del servizio, scegliere l'ID di revisione del servizio	

Per visualizzare le revisioni del servizio per questo tipo di implementazione	Esegui questa operazione	
	per il tipo di revisione di origine. Per visualizzare la revisione del servizio di destinazione, in Revisioni del servizio, scegliere l'ID di revisione del servizio per il tipo di revisione di destinazione.	
Ultima implementazione	In Ultima implementazione, scegliere la Revisione del servizio di destinazione.	
Implementazioni completate	In Implementazioni del servizio, procedere come segue: In Revisione del servizio di destinazione, scegliere l'ID.	

AWS CLI

1. Eseguire `describe-service-deployments` per recuperare l'ARN della revisione del servizio.

Sostituire le variabili con i propri valori.

```
aws ecs describe-service-deployments --service-deployment-arns
arn:aws:ecs:region:account-id:service/cluster-name/service-name/NCWGC2ZR-
taawPAYrIaU5
```

Annotare il arn per `sourceServiceRevisions` o `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`. Usare il `arn` restituito da `describe-service-deployments`.

Sostituire le variabili con i propri valori.

```
aws ecs describe-service-revisions --service-revision-arns
arn:aws:ecs:region:123456789012:service-revision/cluster-name/service-
name/4980306466373577095
```

Bilanciamento di un servizio Amazon ECS tra zone di disponibilità

A partire dal 5 settembre 2025, Amazon ECS abilita il ribilanciamento delle zone di disponibilità per tutti i servizi adatti a questa funzionalità. Un servizio è adatto quando la distribuzione della zona di disponibilità è la prima strategia di posizionamento delle attività o quando non esiste una strategia di posizionamento.

Per aiutare le applicazioni a raggiungere un'elevata disponibilità, consigliamo di configurare i servizi multi-task per l'esecuzione su più zone di disponibilità. Per i servizi che specificano come prima strategia di collocamento la distribuzione in zone di disponibilità, AWS fa del suo meglio per distribuire equamente le attività di servizio tra le zone di disponibilità disponibili.

Tuttavia, a volte il numero di attività in esecuzione in una zona di disponibilità non è uguale a quello delle altre zone di disponibilità, ad esempio dopo un'interruzione della zona di disponibilità. Per risolvere questo squilibrio delle attività, puoi abilitare la funzione di ribilanciamento della zona di disponibilità.

Con il ribilanciamento delle zone di disponibilità, Amazon ECS monitora continuamente la distribuzione delle attività nelle zone di disponibilità per ciascuno dei tuoi servizi. Quando Amazon ECS rileva una distribuzione irregolare delle attività, interviene automaticamente per ribilanciare il carico di lavoro nelle zone di disponibilità. Ciò comporta l'avvio di nuove attività nelle zone di disponibilità con il minor numero di attività e la chiusura 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 automatico elimina la necessità di interventi manuali, accelerando i tempi di ripristino dopo un evento.

Di seguito è riportata una panoramica del processo di ribilanciamento delle zone 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 l'inizio del ribilanciamento delle zone di disponibilità.
- 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.
- Il pianificatore attende che le attività appena avviate siano HEALTHY e RUNNING prima di interromperle nella zona di disponibilità sovradimensionata.
- Invia un evento di servizio con l'esito del ribilanciamento delle zone di disponibilità.

In che modo Amazon ECS rileva la distribuzione irregolare delle attività

Amazon ECS determina uno squilibrio nel numero di attività in esecuzione in ciascuna zona di disponibilità dividendo il numero di attività desiderate del servizio per il numero di zone di disponibilità configurate. Se il numero di attività desiderate non è suddiviso in modo uniforme, Amazon ECS distribuisce il resto delle attività in modo uniforme tra le zone di disponibilità configurate. Ogni zona di disponibilità deve avere almeno un'attività.

Ad esempio, consideriamo un servizio Amazon ECS con un numero desiderato di due attività configurate per due zone di disponibilità. In questo scenario, il numero di attività desiderate è suddiviso in modo uniforme. Una distribuzione bilanciata corrisponderebbe a un'attività per zona di disponibilità. Se ci sono due attività nella zona di disponibilità 1 e zero attività nella zona di disponibilità 2, Amazon ECS inizierà il ribilanciamento avviando un'attività nella zona di disponibilità 2 prima di interrompere un'attività nella zona di disponibilità 1.

Ora, consideriamo un servizio Amazon ECS con un numero desiderato di tre attività configurate per due zone di disponibilità. In questo scenario, il numero di attività desiderate non viene suddiviso in modo uniforme. Una distribuzione bilanciata sarebbe costituita da un'attività nella zona di disponibilità 1 e due attività nella zona di disponibilità 2, poiché ogni zona di disponibilità ha almeno un'attività e quella rimanente viene posizionata nella zona di disponibilità 2.

Consideriamo un servizio Amazon ECS che ha un numero desiderato di cinque attività configurate per due zone di disponibilità. In questo scenario, il numero di attività desiderate non viene suddiviso in modo uniforme. Una distribuzione bilanciata sarebbe costituita da un'attività nella zona di disponibilità 1 e due attività ciascuna nelle zone di disponibilità 2 e 3. Dopo aver considerato ogni zona di disponibilità con un'attività ciascuna, le due attività rimanenti vengono distribuite equamente tra le zone di disponibilità.

Considerazioni per la configurazione del ribilanciamento delle zone di disponibilità

Quando si desidera configurare il ribilanciamento della zona di disponibilità, considerare quanto segue:

- Il ribilanciamento della zona di disponibilità supporta Fargate e i provider di EC2 capacità ed è supportato sulle istanze gestite di Amazon ECS. Per Fargate, Amazon ECS ridistribuirà automaticamente le attività tra le zone di disponibilità presenti per mantenere l'equilibrio. Per i fornitori di EC2 capacità, 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 avviare nuove istanze in zone di disponibilità sottoutilizzate nell'ambito del processo di ribilanciamento, limitandolo 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 posizionamento delle attività oppure non specificano una strategia di posizionamento.
- Non è possibile utilizzare il ribilanciamento delle zone di disponibilità con servizi che soddisfano uno dei seguenti criteri:
 - Utilizza la strategia `Daemon`
 - Utilizza il tipo di avvio `EXTERNAL` (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 la terminazione del posizionamento delle attività. I vincoli di posizionamento delle attività sono regole che determinano se un'attività può essere eseguita su una specifica istanza di container.

Infatti EC2, puoi utilizzare strategie di posizionamento e vincoli di posizionamento insieme al ribilanciamento delle zone di disponibilità. Tuttavia, affinché il ribilanciamento delle zone di disponibilità funzioni, la strategia di posizionamento della distribuzione delle 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 distribuisce innanzitutto le attività in modo uniforme all'interno delle zone di disponibilità, quindi esegue il bin packing delle 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 distribuzione delle zone di disponibilità viene specificata per prima.

È importante notare che il ribilanciamento delle zone di disponibilità non funzionerà se la prima strategia nella serie di strategia di posizionamento non è un componente di distribuzione 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 posizionamento delle attività, consultare [In che modo Amazon ECS colloca le attività sulle istanze dei container](#).

Per attività che usano 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 seguente di esempio 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. Il ribilanciamento delle zone di disponibilità è compatibile con il servizio perché la strategia spread è prioritaria.

```
"placementStrategy": [  
  {  
    "field": "attribute:ecs.availability-zone",  
    "type": "spread"  
  },  
  {  
    "field": "memory",  
    "type": "binpack"  
  }  
]
```

Attivare il ribilanciamento delle zone di disponibilità

È necessario abilitare il ribilanciamento delle zone di disponibilità per i servizi nuovi ed esistenti.

È possibile abilitare e disabilitare il ribilanciamento della zona di disponibilità utilizzando la console,, e. APIs AWS CLI CloudFormation

Il comportamento predefinito di AvailabilityZoneRebalancing differisce tra le richieste di creazione e di aggiornamento:

- Per le richieste di creazione di servizi, quando non viene specificato alcun valore per AvailabilityZoneRebalancing, Amazon ECS imposta il valore predefinito su ENABLED.
- Per le richieste di aggiornamento dei servizi, quando non viene specificato alcun valore per AvailabilityZoneRebalancing, Amazon ECS imposta il valore predefinito del servizio esistente su AvailabilityZoneRebalancing. Se il servizio non ha mai avuto un valore AvailabilityZoneRebalancing impostato, Amazon ECS lo considera come DISABLED.

Tipo di servizio	"Hello, World!"	Console	CLI
Esistente	UpdateService	Aggiornamento di un servizio Amazon ECS	update-service
Novità	CreateService	Creazione di un'implementazione di aggiornamenti continui di Amazon ECS	create-service

L'esempio seguente mostra come abilitare il ribilanciamento del servizio durante la creazione di un nuovo servizio:

```
aws ecs create-service \
  --cluster my-cluster \
  --service-name my-service \
  --task-definition my-task-definition:1 \
  --desired-count 6 \
  --availability-zone-rebalancing ENABLED
```

Monitorare il ribilanciamento delle zone di disponibilità di Amazon ECS

È possibile verificare se il ribilanciamento delle zone 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 operazioni di servizio per favorire la comprensione del ciclo di vita del ribilanciamento delle zone di disponibilità.

Event	Scenario	Tipo	Ulteriori informazioni
SERVICE_REBALANCING_STARTED	Amazon ECS avvia un'operazione di ribilanciamento delle zone di disponibilità	INFO	service (<i>service-name</i>) non è bilanciato in AZ con <i>number-tasks</i> le attività in, in e in. <i>Availability Zone 1number-tasks Availability Zone 2number-tasks Availability Zone 3</i> Ribilanciamento AZ in corso.
SERVICE_REBALANCING_COMPLETED	L'operazione di ribilanciamento delle	INFO	service (<i>service-name</i>) è bilanciato in AZ con <i>number-</i>

Event	Scenario	Tipo	Ulteriori informazioni
	zone di disponibilità viene completata		<i>tasks number-</i> <i>tasks task in</i> <i>Availability</i> <i>Zone 1Availability</i> <i>ity Zone 2 ,tasks</i> <i>in e number-tasks</i> <i>tasks inAvailabil</i> <i>ity Zone 3 .</i>
TASKS_STARTED	Amazon ECS avvia correttamente le attività nell'ambito dell'operazione di ribilanciamento delle zone di disponibilità	INFO	<i>service-name ha</i> <i>avviato number-</i> <i>tasks attività in</i> <i>Availability</i> <i>Zone AZ Rebalance:.</i> <i>task-ids</i>
TASKS_STOPPED	Amazon ECS interrompe correttam ente le attività nell'ambito dell'oper azione di ribilanci amento delle zone di disponibilità	INFO	<i>service-name ha</i> <i>interrotto l'number-</i> <i>tasks esecuzione</i> <i>e delle attività a</i> <i>Availability</i> <i>Zone causa del</i> <i>ribilanciamento AZ:.</i> <i>task-id</i>

Event	Scenario	Tipo	Ulteriori informazioni
SERVICE_TASK_PLACEMENT_FAILURE	Amazon ECS non ha potuto avviare un'attività nell'ambito dell'operazione di ribilanciamento delle zone di disponibilità	ERRORE	Per EC2, vedi service (<i>service-name</i>) non è in grado di inserire un'attività Availability Zone perché nessuna istanza del contenuto soddisfa tutti i requisiti. Per Fargate, vedere 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 delle zone 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 <i>task-set-name</i> è stato in grado di scalare a causa di di. <i>reason</i>
SERVICE_REBALANCING_STOPPED	L'operazione di ribilanciamento delle zone di disponibilità viene interrotta. Amazon ECS invia eventi aggiuntivi che forniscono informazioni aggiuntive.	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 nell'ambito del processo di ribilanciamento.

Amazon ECS invia un evento di modifica dello stato dell'attività (STOPPED) per ogni attività interrotta nell'ambito del processo di ribilanciamento. Il motivo è impostato su Availability-zone rebalancing initiated by (deployment ecs-svc/*deployment-id*).

Per ulteriori informazioni sugli eventi, consultare [Eventi di modifica dello stato dei processi di Amazon ECS](#).

Risoluzione dei problemi di ribilanciamento del servizio

Se si riscontrano problemi con il ribilanciamento del servizio, considerare i seguenti passaggi per la loro risoluzione:

Il ribilanciamento non si avvia

Verificare che:

- Il ribilanciamento del servizio è abilitato per il servizio
- Il servizio utilizza una configurazione supportata (vedere [the section called “Considerazioni per la configurazione del ribilanciamento delle zone di disponibilità”](#))
- Il servizio ha raggiunto uno stato stazionario

Errori nel posizionamento delle attività durante il ribilanciamento

Se si vedono eventi SERVICE_TASK_PLACEMENT_FAILURE:

- Per EC2: verifica se sono disponibili istanze di container nella zona di disponibilità di destinazione
- Per Fargate: verificare se ci sono vincoli di risorse o quote di servizio che limitano il posizionamento delle attività
- Esaminare i vincoli relativi al posizionamento delle attività per assicurarsi che non impediscano una corretta distribuzione delle attività

Il ribilanciamento si interrompe in modo imprevisto

Se si vedono eventi SERVICE_REBALANCING_STOPPED:

- Verificare la protezione delle attività che potrebbero bloccare l'operazione
- Cercare implementazioni simultanee di servizi che potrebbero interrompere il ribilanciamento

- Rivedere gli eventi di servizio per ulteriori informazioni sul motivo per cui il ribilanciamento è stato interrotto

Best practice per il ribilanciamento del servizio

Seguire le best practice per ottimizzare il ribilanciamento del servizio:

- Monitora le operazioni di ribilanciamento: imposta gli CloudWatch allarmi per monitorare gli eventi di servizio relativi al ribilanciamento e identificare rapidamente eventuali problemi.
- Considerare l'impatto sulle prestazioni: tenere presente che le operazioni di ribilanciamento possono aumentare temporaneamente l'utilizzo delle risorse man mano che vengono avviate nuove attività prima che quelle precedenti vengano interrotte.
- Utilizzare la protezione delle attività in modo strategico: se si dispone di attività critiche che non devono essere interrotte durante il ribilanciamento, considerare l'utilizzo della protezione delle attività.
- Pianifica la EC2 capacità: assicurati infatti di disporre di un numero sufficiente di istanze di container in tutte le zone di disponibilità per EC2 supportare un ribilanciamento efficace.
- Testare il comportamento di ribilanciamento: prima di affidarsi al ribilanciamento in produzione, testare il comportamento dei servizi durante le operazioni di ribilanciamento in un ambiente non di produzione.

Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS

È possibile scegliere di configurare il servizio per l'utilizzo del bilanciamento del carico elastico per l'implementazione uniforme del traffico fra le attività nel 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. Utilizzare la tabella seguente per ottenere maggiori informazioni sul tipo bilanciatore del carico da utilizzare.

Tipo di bilanciatore del carico	Utilizzare in questi casi	
Application Load Balancer	Traffico di routing HTTP/HTTP S (o livello 7). Gli Application Load Balancer offrono diverse funzionalità che li rendono particolarmente 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	

Tipo di bilanciatore del carico	Utilizzare in questi casi	
	listener su un singolo Application Load Balancer).	
Network Load Balancer	Instradare il traffico TCP o UDP (o livello 4).	
Gateway Load Balancer	Instradare il traffico TCP o UDP (o livello 4). Usare le appliance virtuali, come firewall, sistemi di prevenzione e rilevamento delle intrusioni, e sistemi di ispezione approfondita dei pacchetti.	

È consigliabile utilizzare gli Application Load Balancer per i tuoi servizi Amazon ECS, in modo da sfruttare queste funzioni recenti, a meno che il servizio non richieda una funzionalità disponibile solo con i 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, consultare [Prezzi di Elastic Load Balancing](#).

Ottimizzare i parametri di controllo dell'integrità del bilanciatore del carico per Amazon ECS

I bilanciatori del carico instradano le richieste solamente sui target integri all'interno delle zone di disponibilità per il bilanciatore del carico. Ogni destinazione è registrata in un gruppo di destinazione. Il bilanciatore del carico controlla l'integrità di ogni destinazione, utilizzando le impostazioni di controllo dell'integrità del gruppo di destinazione. Dopo aver registrato una destinazione, deve superare un controllo dell'integrità per essere considerata integra. Amazon ECS monitora il bilanciatore del carico. Il bilanciatore del carico invia periodicamente controlli dell'integrità al container di Amazon ECS. L'agente Amazon ECS monitora e attende che il bilanciatore del carico riferisca sull'integrità del container. Ciò avviene prima di considerare il container in uno stato integro.

Due parametri di controllo dell'integrità del bilanciamento del carico elastico influiscono sulla velocità di implementazione:

- **Intervallo del controllo dell'integrità:** determina il periodo di tempo approssimativo, in secondi, tra i controlli dell'integrità di un singolo container. Per impostazione predefinita, il bilanciatore del carico effettua i controlli ogni 30 secondi.

Questo parametro è nominato:

- `HealthCheckIntervalSeconds` nell'API di bilanciamento del carico elastico
- `Interval` sulla console Amazon EC2
- **Numero di soglia di integrità:** determina il numero di controlli dell'integrità consecutivi superati necessari prima di considerare integro un container non integro. Per impostazione predefinita, il bilanciatore del carico richiede il superamento di cinque controlli di integrità prima di segnalare che il container di destinazione è integro.

Questo parametro è nominato:

- `HealthyThresholdCount` nell'API di bilanciamento del carico elastico
- `Soglia integra` sulla EC2 console Amazon

Importante: per le destinazioni appena registrate, è necessario un solo controllo dell'integrità riuscito per considerare la destinazione integra, indipendentemente dall'impostazione del conteggio delle soglie di integrità. Il numero delle soglie di integrità si applica solo quando una destinazione passa da uno stato non integro a uno stato integro.

Con le impostazioni predefinite, se una destinazione diventa non integra e viene ripristinata, il tempo totale per determinare l'integrità di un container è di due minuti e 30 secondi ($30 \text{ seconds} * 5 = 150 \text{ seconds}$).

È possibile accelerare il processo di controllo dell'integrità se il servizio si avvia e si stabilizza in meno di 10 secondi. Per accelerare il processo, ridurre l'intervallo del controllo dell'integrità e il numero delle soglie di integrità.

- `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 dell'integrità richiede 10 secondi rispetto all'impostazione predefinita di due minuti e 30 secondi.

Per ulteriori informazioni sui parametri del controllo dell'integrità del bilanciamento del carico elastico, consultare [Health checks for your target groups](#) nella Guida per l'utente del bilanciamento del carico elastico.

Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore 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 bilanciatore del carico. Il bilanciatore del carico controlla periodicamente se il client ha chiuso la connessione keep-alive. Amazon ECS monitora il bilanciatore del carico e attende che il bilanciatore del carico segnali che la connessione keep-alive è chiusa (la destinazione è in uno stato UNUSED).

La quantità di tempo che il bilanciatore del carico attende per spostare la destinazione verso lo stato UNUSED corrisponde al ritardo di annullamento della registrazione. È possibile configurare il seguente parametro del bilanciatore del carico per velocizzare le implementazioni.

- `deregistration_delay.timeout_seconds`: 300 (impostazione predefinita)

Se si dispone di un servizio con un tempo di risposta inferiore a 1 secondo, impostare il parametro sul valore seguente per fare in modo che il bilanciatore del carico attenda solo 5 secondi prima di interrompere la connessione tra il client e il servizio di backend:

- `deregistration_delay.timeout_seconds`: 5

Note

Non impostare il valore su 5 secondi quando si dispone di un servizio con richieste di lunga durata, come caricamenti lenti di file o connessioni in streaming.

Reattività SIGTERM

Amazon ECS invia innanzitutto un segnale di arresto all'attività per notificare che l'applicazione deve essere completata e chiusa. Questo segnale può essere definito nell'immagine del contenitore con

l'istruzione `STOPSIGNAL` e verrà impostato automaticamente su `SIGTERM`. Poi, 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 di 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, impostare il parametro dell'agente Amazon ECS sul valore seguente:

- `ECS_CONTAINER_STOP_TIMEOUT`: 2

Se l'applicazione impiega più di 1 secondo, moltiplicare il valore per 2 e usare quel numero come valore.

In questo caso, Amazon ECS attende 2 secondi che il container si arresti, quindi invia un messaggio `SIGKILL` quando l'applicazione non si ferma.

È anche possibile 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 eventuali nuove richieste, finisca di rispondere a tutte le richieste in corso e quindi il processo Node.js termina perché il ciclo di eventi non ha nulla da fare. Pertanto, se impiega solo 500 ms per completare le richieste in corso, termina in anticipo senza dover attendere il timeout di arresto e ricevere un `SIGKILL`.

Utilizzo di 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). Una volta avviata l'attività, il container NGINX viene registrato nell'Application 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, consultare la [Guida per l'utente per Application Load Balancer](#).

Per informazioni sulle best practice per impostare i parametri per velocizzare le implementazioni, consultare:

- [Ottimizzare i parametri di controllo dell'integrità del bilanciatore del carico per Amazon ECS](#)
- [Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore del carico per Amazon ECS](#)

Quando si utilizzano gli Application Load Balancer con Amazon ECS, tenere presenti le considerazioni seguenti:

- 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 in una configurazione IPv6 -only, è necessario impostare il tipo di indirizzo IP del gruppo di destinazione dell'Application Load Balancer `dualstack` su `o. dualstack-without-public-ipv4`.
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando si crea un gruppo di destinazione per il servizio, è necessario 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 EC2 Amazon.
- Se il tuo servizio richiede l'accesso a più porte con bilanciamento del carico, come la porta 80 e la porta 443 per un HTTP/HTTPS servizio, 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, consultare [Create a listener to your 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 Console di gestione AWS. 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 o 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 l'attività di un servizio non supera i criteri di controllo dell'integrità previsti per il bilanciamento del carico, viene arrestata 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 dei bilanciatori del carico per il servizio in Amazon ECS](#).
- Quando si utilizza il tipo di destinazione `instance`, le attività e il bilanciamento del carico devono trovarsi nello stesso VPC. Quando si utilizza il tipo di destinazione `ip`, è supportata la connettività tra VPC.
- Utilizzare un gruppo di destinazione unico per ogni servizio.

L'utilizzo dello stesso gruppo di destinazione per più servizi potrebbe causare problemi durante le implementazioni dei servizi.

- È necessario specificare i gruppi target associati a un Application Load Balancer.

Per informazioni su come creare un Application Load Balancer, consultare [Create an Application Load Balancer](#) in Application Load Balancer

Utilizzo di 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 bilanciamento del carico 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, consultare [User Guide for Network Load Balancers](#).

Per informazioni sulle best practice per impostare i parametri per velocizzare le implementazioni, consultare:

- [Ottimizzare i parametri di controllo dell'integrità del bilanciatore del carico per Amazon ECS](#)
- [Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore del carico per Amazon ECS](#)

Quando si utilizzano i Network Load Balancer con Amazon ECS, tenere presenti le considerazioni seguenti:

- 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, consultare [Uso di ruoli collegati ai servizi per Amazon ECS](#).
- Non è possibile collegare più di cinque gruppi di destinazione a un servizio.
- Per i servizi in una configurazione IPv6 -only, è necessario impostare il tipo di indirizzo IP del gruppo di destinazione del Network Load dualstack Balancer su.
- Per i servizi con attività che utilizzano la modalità di rete awsvpc, quando si crea un gruppo di destinazione per il servizio, 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.
- 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 Console di gestione AWS. 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 l'attività di un servizio non supera i criteri di controllo dell'integrità previsti per il bilanciatore del carico, viene arrestata 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 la conservazione dell'IP client è disabilitata, le richieste vengono viste come provenienti dall'indirizzo IP privato del Gateway Load Balancer. Ciò significa che i servizi dietro un Gateway Load Balancer vengono effettivamente aperti al mondo non appena si consentono le richieste in entrata e i controlli di integrità nel gruppo di sicurezza di destinazione.
- 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 dei bilanciatori del carico per il servizio in Amazon ECS](#).
- Quando si utilizza il tipo di destinazione `instance`, le attività e il bilanciatore del carico devono trovarsi nello stesso VPC. Quando si utilizza il tipo di destinazione `ip`, è supportata la connettività tra VPC.
- La conservazione dell'indirizzo IP client del Network Load Balancer è compatibile con le destinazioni Fargate.
- Utilizzare un gruppo di destinazione unico per ogni servizio.

L'utilizzo dello stesso gruppo di destinazione per più servizi potrebbe causare problemi durante le implementazioni dei servizi.

- È necessario specificare i gruppi target associati a un Network Load Balancer.

Per informazioni su come creare un Network Load Balancer, consultare [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 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.

Utilizzo di 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 un dispositivo di destinazione specifico utilizzando 5 tuple (per i flussi) o 3 tuple (per i flussi non TCP/UDP). 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). Una volta avviata l'attività, il container NGINX viene registrato con il Gateway 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, consultare [What is a Gateway Load Balancer](#) in Gateway Load Balancer.

Per informazioni sulle best practice per impostare i parametri per velocizzare le implementazioni, consultare:

- [Ottimizzare i parametri di controllo dell'integrità del bilanciatore del carico per Amazon ECS](#)
- [Ottimizzazione dei parametri di svuotamento della connessione del bilanciatore del carico per Amazon ECS](#)

Quando si utilizzano i Gateway Load Balancer con Amazon ECS, tenere presenti le considerazioni seguenti:

- 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 in una configurazione IPv6 solo, è necessario impostare il tipo di indirizzo IP del gruppo di destinazione del Gateway Load dualstack Balancer su.
- 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 Console di gestione AWS. 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 l'attività di un servizio non supera i criteri di controllo dell'integrità previsti per il bilanciatore del carico, viene arrestata 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 dietro un Gateway Load Balancer vengono effettivamente aperti al mondo non appena si consentono le richieste in entrata e i controlli di integrità nel gruppo di sicurezza di destinazione.
- 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 dei bilanciatori del carico per il servizio in Amazon ECS](#).
- Quando si utilizza il tipo di destinazione `instance`, le attività e il bilanciatore del carico devono trovarsi nello stesso VPC. Quando si utilizza il tipo di destinazione `ip`, è supportata la connettività tra VPC.

- Utilizzare un gruppo di destinazione unico per ogni servizio.

L'utilizzo dello stesso gruppo di destinazione per più servizi potrebbe causare problemi durante le implementazioni dei servizi.

- È necessario specificare i gruppi target associati a un Gateway Load Balancer.

Per informazioni su come creare un Gateway Load Balancer, consultare [Getting started with Gateway Load Balancers](#) in Gateway Load Balancer

Important

Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, necessaria per 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 di destinazione 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. CloudFormation Dopo aver creato il servizio, è possibile visualizzare il servizio e i gruppi di destinazione registrati in esso con la Console di gestione AWS. 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). Può trattarsi della distribuzione native/blue ecologica di Amazon ECS o della distribuzione di aggiornamento progressivo.
- 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 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 load balancer 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
  }
]
```

Esporre più porte dallo stesso container

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
  }
]
```

Esporre porte da 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
  }
]
```

Scalabilità automatica del servizio Amazon ECS

La scalabilità automatica è la capacità di aumentare o diminuire automaticamente il numero di processi nel servizio Amazon ECS. Amazon ECS utilizza il servizio di Application Auto Scaling per fornire questa funzionalità. Per ulteriori informazioni, consultare [Application Auto Scaling User Guide](#).

Amazon ECS pubblica CloudWatch parametri con l'utilizzo medio di CPU e memoria del tuo servizio. Per ulteriori informazioni, consulta [Parametri 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](#): aumenta o riduce il numero di processi eseguiti dal servizio in base alla data e all'ora.
- [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#): aumenta o riduce il numero di attività eseguite dal servizio in base all'analisi storica dei dati di carico per rilevare 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 un numero desiderato del servizio è impostato al di sotto del suo valore di capacità minimo e un allarme avvia un'attività di dimensionamento orizzontale, Service Auto Scaling dimensiona il numero desiderato fino al valore di capacità minimo e continua ad aumentare in base alle necessità e alla policy di dimensionamento 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 un numero desiderato del servizio è impostato al di sopra del suo valore di capacità massimo e un allarme avvia un'attività di riduzione, Service Auto Scaling aumenta il numero desiderato fino al valore di capacità massimo e continua a ridurre in base alle necessità e alla policy di dimensionamento 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. Questo comportamento non si applica ai servizi di Amazon ECS che utilizzano il controller di implementazione esterno. Per ulteriori informazioni, consultare [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.

Per ulteriori informazioni sulle best practice per il dimensionamento automatico dei servizi, consultare [Ottimizzazione del dimensionamento automatico del servizio Amazon ECS](#).

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 l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Questo comportamento non si applica ai servizi di Amazon ECS che utilizzano il controller di implementazione esterno. 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, consultare [Suspending and resuming scaling for 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 politica di scalabilità aggiunge o rimuove le attività di servizio necessarie per mantenere la metrica pari o vicina al valore target specificato. Oltre a mantenere la metrica vicina al valore target, una politica di scalabilità basata sul tracciamento degli obiettivi si adatta anche alle fluttuazioni della metrica dovute a uno schema di carico fluttuante e riduce al minimo le rapide fluttuazioni nel numero di attività in esecuzione nel servizio.

Le policy di tracciamento di Target eliminano la necessità di definire manualmente gli allarmi e gli aggiustamenti di scalabilità. CloudWatch Amazon ECS lo gestisce automaticamente in base alla destinazione impostata.

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. Questo comportamento non si applica ai servizi di Amazon ECS che utilizzano il controller di implementazione esterno. Per ulteriori informazioni, consultare [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.
- La ALBRequestCountPerTarget metrica per le politiche di scalabilità del tracciamento degli obiettivi non è supportata per il tipo di implementazione. blue/green

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.

Creazione di una policy di dimensionamento con monitoraggio delle destinazioni per dimensionamento automatico del servizio Amazon ECS

Creare una policy di scalabilità di tracciamento di destinazione per consentire ad Amazon ECS di aumentare o diminuire automaticamente il numero di attività desiderate nel servizio. Il monitoraggio della destinazione 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, consultare [Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi Amazon ECS](#).
2. Determinare 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 Cluster, scegliere il cluster.
5. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

6. Scegliere Imposta il numero delle attività.
7. Nella sezione Conteggio delle attività del servizio di Amazon ECS, scegliere Utilizzare il dimensionamento automatico.

Si apre la Sezione dedicata al conteggio delle attività.

- a. Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. Per Massimo, inserire il limite superiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Si apre la pagina delle policy.

8. Scegliere Crea una policy di dimensionamento.

Si apre la pagina Crea policy.

9. In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi.
10. In Nome policy, inserire il nome della policy.
11. In Tipo di parametro, scegliere i parametri nell'elenco di opzioni.
12. Per Utilizzo di destinazione, inserire il valore della destinazione per la percentuale di attività che Amazon ECS deve mantenere. Il dimensionamento automatico del servizio aumenta orizzontalmente la capacità finché l'utilizzo medio arriva a quello di destinazione o finché non raggiunge il numero massimo di attività specificato.
13. In Impostazioni aggiuntive, procedere come segue
 - a. In Tempo di raffreddamento di riduzione orizzontale, immettere la quantità di tempo che deve passare, in secondi, tra il completamento di un'attività di riduzione e l'inizio di un'altra attività di questo tipo.
 - b. Per Tempo di raffreddamento di aumento orizzontale, immettere la quantità di tempo, espressa in secondi, da aspettare per rendere effettiva un'attività di aumento precedente.

- c. Per creare una policy di aumento, selezionare Disabilita riduzione.
14. Scegliere Crea una policy di dimensionamento.

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 policy di dimensionamento in fasi ridimensiona le attività utilizzando una serie di regolazioni, note come regolazioni della fase. 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 applicherà la prima modifica.
- Se la violazione supera la seconda soglia, Amazon ECS applicherà la seconda modifica, e così via.

Si consiglia vivamente di utilizzare policy di dimensionamento del monitoraggio delle destinazioni per eseguire un dimensionamento in base a parametri come l'utilizzo medio della CPU o il numero di richieste medio per destinazione. I parametri che diminuiscono quando la capacità aumenta e aumentano quando la capacità diminuisce possono essere utilizzati per dimensionare proporzionalmente il numero di attività mediante il monitoraggio degli obiettivi. Questo contribuisce a garantire che Amazon ECS segua da vicino la curva della domanda per le applicazioni.

Creazione di una policy di dimensionamento graduale per il dimensionamento automatico del servizio Amazon ECS

Creare una politica di scalabilità graduale per consentire ad Amazon ECS di aumentare o diminuire automaticamente il numero di attività desiderate nel servizio. Il dimensionamento graduale viene eseguito in base a un set di adeguamenti del dimensionamento, noti come adeguamenti di fasi, che variano in base alla dimensione dell'utilizzo fuori limite segnalato dall'allarme.

Console

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consultare [Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi Amazon ECS](#).
2. Determinare 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 Cluster, scegliere il cluster.
6. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

7. Scegliere Imposta il numero delle attività.
8. Nella sezione Conteggio delle attività del servizio di Amazon ECS, scegliere Utilizzare il dimensionamento automatico.

Si apre la Sezione dedicata al conteggio delle attività.

- a. Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. Per Massimo, inserire il limite superiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Si apre la pagina delle policy.

9. Scegliere Crea una policy di dimensionamento.

Si apre la pagina Crea policy.

10. In Tipo di policy di dimensionamento, scegliere Dimensionamento graduale.
11. Configurare le proprietà di aumento orizzontale. In Passaggi per aggiungere attività, procedere come segue:
 - a. In Nome policy, inserire il nome della policy.
 - b. Per il nome della CloudWatch sveglia, scegli la CloudWatch sveglia.
 - c. In Tipo di aggregazione delle metriche, scegliere come confrontare il parametro selezionato con la soglia definita.
 - d. In Tipi di regolazione, scegliere se la regolazione si basa su una modifica del numero delle attività o su una modifica della percentuale di attività.
 - e. In Azioni da intraprendere, inserire i valori relativi all'azione da intraprendere.

Scegliere Aggiungere passaggio per aggiungere azioni aggiuntive.

12. Configurare le proprietà di riduzione orizzontale. In Passaggi per rimuovere attività, procedere come segue:
 - a. In Nome policy, inserire il nome della policy.
 - b. Per il nome della CloudWatch sveglia, scegli la CloudWatch sveglia.
 - c. In Tipo di aggregazione delle metriche, scegliere come confrontare il parametro selezionato con la soglia definita.
 - d. In Tipi di regolazione, scegliere se la regolazione si basa su una modifica del numero delle attività o su una modifica della percentuale di attività.
 - e. In Azioni da intraprendere, inserire i valori relativi all'azione da intraprendere.

Scegliere Aggiungere passaggio per aggiungere azioni aggiuntive.

13. In Tempo di raffreddamento, inserire la quantità di tempo, espressa in secondi, necessaria per rendere effettiva un'attività di dimensionamento precedente. Per una policy di aggiunta, si tratta del periodo successivo a un'attività di aumento orizzontale in cui la politica di scalabilità blocca le attività con aumento orizzontale e limita il numero di attività che possono essere aumentate orizzontalmente alla volta. Per una policy di rimozione, questo è il tempo che deve passare tra un'attività di riduzione e l'inizio di un'altra attività di questo tipo.
14. Scegliere Crea una policy di dimensionamento.

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 il dimensionamento pianificato, è possibile impostare il dimensionamento automatico per l'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 in altri momenti non fornisce il numero di attività eccedente.

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'operazione pianificata di dimensionamento, la policy di dimensionamento può continuare a prendere decisioni sull'opportunità di dimensionare ulteriormente il numero di attività. In questo modo è possibile garantire di disporre di un numero di attività sufficiente per la gestione dei carichi dell'applicazione. Sebbene l'applicazione si dimensioni per soddisfare la domanda, la capacità corrente deve rientrare nel numero minimo e massimo di attività impostati dall'operazione pianificata.

È possibile configurare il dimensionamento della pianificazione utilizzando AWS CLI. Per ulteriori informazioni sul dimensionamento pianificato, consultare [Scheduled Scaling](#) nella Guida per l'utente del dimensionamento automatico delle applicazioni.

Creare un'operazione pianificata per il dimensionamento automatico del servizio Amazon ECS

Creare un'operazione pianificata per fare in modo che Amazon ECS aumenti o diminuisca il numero di attività eseguite dal servizio in base alla data e all'ora.

Console

1. [Apri la console nella versione 2https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/).
2. Nella pagina Cluster, scegliere il cluster.

3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Scegliere Dimensionamento automatico del servizio.

Si apre la pagina di dimensionamento automatico del servizio.

5. Se non è stato configurato il dimensionamento automatico del servizio, scegliere Imposta il numero di attività.

Viene visualizzata la sezione Numero di attività del servizio Amazon ECS.

In Numero di attività del servizio Amazon ECS, scegliere Usa il dimensionamento automatico del servizio per modificare il numero di attività desiderato per il servizio.

Si apre la Sezione dedicata al conteggio delle attività.

- a. Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. Per Massimo, inserire il limite superiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegliere Salva.

Si apre la pagina delle policy.

6. Scegliere Operazioni pianificate e poi scegliere Crea.

Si apre la pagina Crea operazione pianificata.

7. Per Nome operazione, inserire 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, consultare [List of tz database time zones](#).

9. Per Ora di inizio, inserire la data e l'ora di inizio dell'operazione.

Se si sceglie 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 dimensionare in base a una pianificazione ricorrente, scegliere la frequenza con cui Amazon ECS esegue l'operazione pianificata.
- Se si sceglie un'operazione che inizia con Rate, viene creata l'espressione cron.
- Se si sceglie Cron, inserire un'espressione cron che specifichi quando eseguire l'operazione.
- Per dimensionare una sola volta, scegliere Una volta.

11. In Regolazioni delle attività, procedere come segue:

- Per Minimo, inserire il numero minimo di attività che il servizio deve eseguire.
- Per Massimo, inserire 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
```

Ad esempio: per pianificare il dimensionamento in base a una pianificazione ricorrente

Utilizza il seguente comando [put-scheduled-action](#). *user input* Sostituiscili 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, includere l'opzione `--time-zone` e specificare il nome del fuso orario IANA, come nel seguente esempio.

```
--time-zone "America/New_York"
```

Per ulteriori informazioni, consultare [List of tz database time zones](#).

Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva

Il dimensionamento predittivo analizza i dati di caricamento passati provenienti dai flussi di traffico per analizzare i modelli giornalieri o settimanali. Poi utilizza l'analisi per anticipare le esigenze future e aumentare in modo proattivo le attività del tuo servizio secondo necessità.

Il dimensionamento automatico predittivo è particolarmente utile nelle seguenti situazioni:

- **Traffico ciclico:** un uso elevato di risorse durante i normali orari di ufficio e un utilizzo ridotto di risorse durante la notte e il weekend.
- **Modelli 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:** possono influire sulle prestazioni delle applicazioni durante gli eventi di aumento orizzontale, causando una notevole latenza.

Se le applicazioni richiedono molto tempo per l'inizializzazione e il traffico aumenta con uno schema regolare, è consigliabile utilizzare il dimensionamento predittivo. Aiuta a scalare più velocemente aumentando in modo proattivo il numero di attività per i carichi previsti, anziché utilizzare solo politiche di dimensionamento dinamico, come Target Tracking o Step Scaling. Aiutando a evitare la possibilità di effettuare un provisioning eccessivo del numero di attività, il dimensionamento predittivo può anche consentire di 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, il dimensionamento predittivo può aumentare orizzontalmente 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 dedicare tempo alla verifica dei modelli di carico dell'applicazione e al tentativo di pianificare la giusta quantità di attività con il dimensionamento programmato.

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). EC2 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

- [Funzionamento del dimensionamento predittivo in Amazon ECS](#)
- [Creazione di una policy di dimensionamento predittivo per dimensionamento automatico del servizio Amazon ECS](#)
- [Valutazione delle policy di dimensionamento predittivo per Amazon ECS](#)
- [Utilizzare azioni pianificate per sovrascrivere i valori di previsione per Amazon ECS](#)
- [Configurazioni avanzate delle policy di dimensionamento predittivo utilizzando parametri personalizzati per Amazon ECS](#)

Funzionamento del dimensionamento predittivo in Amazon ECS

Qui si possono scoprire le considerazioni sull'utilizzo del dimensionamento predittivo, come funziona e quali sono i limiti.

Considerazioni sull'utilizzo del dimensionamento predittivo

- Il dimensionamento predittivo deve essere adeguato al carico di lavoro. Si può verificare configurando le policy di dimensionamento in modalità solo previsione e vedere cosa consiglia la console. È necessario valutare la previsione e i consigli prima di iniziare a utilizzare il dimensionamento predittivo.
- Prima che il dimensionamento predittivo possa avviare la previsione, servono almeno 24 ore di dati cronologici. Maggiore è il numero di dati cronologici disponibili, più efficace è la previsione, con un periodo ideale di due settimane. Si dovrà inoltre attendere 24 ore prima che il dimensionamento predittivo possa generare nuove previsioni quando si elimina un servizio Amazon ECS e se ne crea uno nuovo. Un modo per velocizzare questa operazione è utilizzare parametri personalizzati per aggregare i parametri tra il vecchio e il nuovo servizio Amazon ECS.

- Scegliere una metrica di carico che rappresenta con precisione il carico completo dell'applicazione ed è l'aspetto dell'applicazione su cui è più importante basare il dimensionamento.
- Il dimensionamento dinamico con dimensionamento predittivo consente di seguire da vicino la richiesta della propria applicazione, in modo ridurre le attività durante le pause e aumentarle durante gli aumenti imprevisti del traffico. Quando sono attive più policy di dimensionamento, ciascuna di esse determina il numero desiderato di attività in modo indipendente, che viene impostato al massimo.
- È possibile utilizzare il dimensionamento predittivo insieme alle policy di dimensionamento dinamico, come il monitoraggio della destinazione o il dimensionamento graduale, in modo che le applicazioni si scalino in base a modelli storici e in tempo reale. Di per sé, il dimensionamento predittivo non consente di ridurre orizzontalmente le attività.
- Se si utilizza un ruolo personalizzato durante la chiamata all'`register-scalable-target` API, si potrebbe ricevere un errore che indica che la policy di dimensionamento predittivo può funzionare solo con SLR abilitata. In questo caso, è necessario chiamare di nuovo `register-scalable-target` ma senza `role-arn`. Usare SLR per registrare la destinazione scalabile e chiamare l'API `put-scaling-policy`.

Funzionamento del dimensionamento predittivo

Puoi utilizzare la scalabilità predittiva creando una politica di scalabilità predittiva che specifica la metrica da monitorare e analizzare. CloudWatch Per avviare valori futuri di previsione, il dimensionamento predittivo richiede almeno 24 ore di dati.

Dopo aver creato la policy, il dimensionamento predittivo inizia ad analizzare i dati metrici relativi agli ultimi 14 giorni per identificare i modelli. Questa analisi viene utilizzata per generare previsioni orarie dei requisiti 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, il dimensionamento predittivo migliora continuamente l'accuratezza delle previsioni future.

La prima volta che si abilita il dimensionamento predittivo, questo viene eseguito in modalità di sola previsione. Genera previsioni in questa modalità, ma non dimensiona il servizio Amazon ECS in base a tali previsioni. Ciò significa che è possibile valutare l'accuratezza e l'idoneità della previsione. È possibile visualizzare i dati di previsione utilizzando l'operazione API `GetPredictiveScalingForecast` o la Console di gestione AWS.

Quando si decide di iniziare a utilizzare il dimensionamento predittivo, si deve passare dalla policy di dimensionamento alla modalità di previsione e scalabilità. In questa modalità si verifica quanto segue.

Per impostazione predefinita, il servizio Amazon ECS viene ridimensionato all'inizio di ogni ora in base alla previsione per quell'ora. È possibile scegliere di iniziare prima utilizzando la proprietà `SchedulingBufferTime` nell'operazione API `PutScalingPolicy`. Ciò consente l'avvio di nuove attività prima della domanda della previsione e si dispone del tempo per l'avvio e la preparazione a gestire il traffico.

Limite massimo di attività

Quando si registrano i servizi Amazon ECS per il dimensionamento, si definisce un numero massimo di attività che possono essere avviate per ogni servizio. Per impostazione predefinita, quando vengono impostate le policy di dimensionamento, non possono aumentare il numero di attività oltre il limite massimo.

In alternativa, è possibile 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

Fare attenzione quando si consente 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 per il servizio Amazon ECS fino a quando non viene aggiornato 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 Pacifico (Seoul)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Asia Pacifico (Tokyo)
- Canada (Centrale)
- Cina (Pechino)
- Cina (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)

Creazione di una policy di dimensionamento predittivo per dimensionamento automatico del servizio Amazon ECS

Creare una policy di dimensionamento predittivo per consentire ad Amazon ECS di aumentare o diminuire il numero di attività eseguite dal servizio in base ai dati cronologici.

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, consultare [Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi Amazon ECS](#).

2. Determinare le metriche da utilizzare per la policy. Sono disponibili i seguenti parametri:
 - ECSServiceMedio 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, è possibile utilizzare una metrica personalizzata. È necessario definire i seguenti valori:

- Carico: una metrica che rappresenta con precisione il carico completo dell'applicazione ed è l'aspetto dell'applicazione su cui è più importante basare il dimensionamento.
 - Metrica di dimensionamento: il miglior indicatore del grado di utilizzo ideale per l'applicazione.
3. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
 4. Nella pagina Cluster, scegliere il cluster.
 5. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

6. Scegliere Dimensionamento automatico del servizio, quindi scegliere Imposta il numero di attività.
7. Nella sezione Conteggio delle attività del servizio di Amazon ECS, scegliere Utilizzare il dimensionamento automatico.

Si apre la Sezione dedicata al conteggio delle attività.

- a. Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
- b. Per Massimo, inserire il limite superiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
- c. Scegli Save (Salva).

Si apre la pagina delle policy.

8. Scegliere Crea una policy di dimensionamento.

Si apre la pagina Crea policy.

9. Per Tipo di policy di dimensionamento, scegliere Dimensionamento predittivo.

10. In Nome policy, inserire il nome della policy.

11. In Coppia di parametri, scegliere i parametri dall'elenco di opzioni.

Se è stato scelto Conteggio delle richieste Application Load Balancer per destinazione, scegliere un gruppo di destinazione in Gruppo di destinazione. Conteggio di richieste Application Load Balancer per destinazione è supportato solo se è stato allegato un gruppo di destinazione Application Load Balancer per il servizio.

Se è stato scelto Coppia di parametri personalizzati, scegliere i singoli parametri dagli elenchi per Parametro del carico e Parametro di dimensionamento.

12. Per Utilizzo di destinazione, inserire il valore della destinazione per la percentuale di attività che Amazon ECS deve mantenere. Il dimensionamento automatico del servizio aumenta orizzontalmente la capacità finché l'utilizzo medio arriva a quello di destinazione o finché non raggiunge il numero massimo di attività specificato.

13. Scegliere Crea una policy di dimensionamento.

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](#)!Amazon EC2 Auto Scaling API Reference.

Esempio 1: una policy di dimensionamento predittivo 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 policy 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 esito positivo, questo comando restituisce l'ARN della policy.

```

{
  "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 policy di dimensionamento predittivo con CPU predefinita.

Di seguito è riportato un esempio di policy con una configurazione di 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 esito positivo, questo comando restituisce l'ARN della policy.

```
{  
  "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": []  
}
```

Valutazione delle policy di dimensionamento predittivo per Amazon ECS

Prima di utilizzare una policy di dimensionamento predittivo per scalare i servizi, esaminare i consigli e gli altri dati per la policy nella console di Amazon ECS. È un'opzione importante per assicurarsi che le previsioni siano accurate prima di applicare una policy di dimensionamento predittivo che dimensioni la capacità effettiva.

Se il servizio è nuovo, attendere 24 ore per creare la prima previsione.

Quando AWS crea una previsione, utilizza dati storici. Se il servizio non dispone ancora di molti dati cronologici recenti, il dimensionamento automatico potrebbe temporaneamente riempire 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 dimensionamento automatico del servizio dovrebbe avere almeno due policy di dimensionamento predittivo 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 policy di dimensionamento predittivo, Amazon ECS inizia immediatamente a valutare quale policy è in grado di dimensionare il gruppo in modo ottimale.

Per visualizzare i suggerimenti nella console Amazon ECS

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Scegliere Dimensionamento automatico del servizio.
5. Scegliere la policy di dimensionamento predittivo, quindi scegliere Operazioni, dimensionamento predittivo, Visualizza consigli.

È possibile visualizzare i dettagli di una policy e i relativi suggerimenti. 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: indica se l'utilizzo della policy eviterebbe un impatto negativo sulla disponibilità eseguendo il provisioning di un numero sufficiente di attività per gestire il carico di lavoro, rispetto al mancato utilizzo della policy.
- Impatto sui costi: indica se l'utilizzo della policy eviterebbe un impatto negativo sui costi non eseguendo un provisioning eccessivo delle attività, rispetto al mancato utilizzo della policy. Se il provisioning è eccessivo, i servizi risultano sottoutilizzati o inattivi, comportando un maggiore 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 dei suggerimenti, scegliere 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, è possibile esaminare le previsioni dei giorni, delle settimane o dei mesi precedenti per visualizzare le prestazioni della policy nel tempo. È inoltre possibile utilizzare queste informazioni per valutare l'accuratezza delle previsioni nel momento in cui si decide di applicare la policy per dimensionare il numero effettivo di attività.

Per esaminare i grafici di monitoraggio di dimensionamento predittivo nella console Amazon ECS

1. Apri la console nella <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Scegliere dimensionamento automatico del servizio.

5. Scegliere la policy di dimensionamento predittivo, quindi scegliere Operazioni, dimensionamento predittivo, 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 Capacità mostra il numero di attività previste dalla policy e 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. Numero effettivo osservato di attività mostra la capacità effettiva del servizio Amazon ECS in passato, che dipende dalle altre policy di ridimensionamento 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 dedotto richiesto di attività mostra il numero ideale di attività nel servizio per mantenere il parametro di dimensionamento al valore target scelto.
4. Il numero minimo delle attività indica il numero minimo di attività nel servizio.
5. La capacità massima indica il numero massimo di attività nel servizio.

Per calcolare la capacità richiesta differita, partiamo dal presupposto che ogni attività sia utilizzata allo stesso modo a un determinato valore target. In pratica, il numero di attività non viene utilizzato allo stesso modo. Tuttavia, ipotizzando che l'utilizzo sia distribuito uniformemente tra le attività, possiamo fare una stima verosimile della quantità di capacità necessaria. Il requisito per il numero di attività viene quindi calcolato in modo che sia inversamente proporzionale al parametro di dimensionamento utilizzato per la relativa policy. In altre parole, all'aumentare del numero di attività, il parametro di dimensionamento diminuisce alla stessa velocità. Ad esempio, se il numero delle attività raddoppia, il parametro di dimensionamento si dimezza.

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 policy di dimensionamento non pianificato. Quindi, effettua il dimensionamento in base alla policy che fornisce il maggior numero di attività.
- Per consentire la riduzione orizzontale quando il carico diminuisce, il servizio deve sempre disporre di almeno una policy di dimensionamento predittivo con la porzione di riduzione orizzontale abilitata.
- Puoi migliorare le prestazioni di dimensionamento assicurandoti che la capacità minima e massima non siano troppo restrittive. Una policy con un numero consigliato di attività che non rientra nell'intervallo di capacità minima e massima non sarà in grado di effettuare l'aumento o la riduzione orizzontale.

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 policy di dimensionamento predittivo 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. Si possono anche vedere le tendenze 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. Scegliere il parametro del namespace Dimensionamento automatico delle applicazioni.
4. Scegliere Previsioni del carico di dimensionamento predittivo.
5. Nel campo di ricerca, immettere il nome della policy di dimensionamento predittivo o il nome del gruppo di servizio Amazon ECS, quindi premere 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'instestazione 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 dei parametri, è possibile rappresentare graficamente i dati generati dal dimensionamento automatico del servizio per il dimensionamento predittivo 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` m2è la CloudWatch metrica che contiene i valori di carico previsti. La sintassi matematica per il parametro MAPE è la seguente:

Media di $(\text{abs}((\text{Effettivo} - \text{Previsto})/(\text{Effettivo})))$

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.

Utilizzare 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 del forecast potrebbero sottovalutare 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'operazione pianificata con un numero di attività superiore a quello previsto. In fase di runtime, Amazon ECS aggiorna il numero minimo di attività del servizio. Poiché il dimensionamento predittivo ottimizza il numero di attività, viene osservata un'azione pianificata con un numero di attività minimo superiore ai valori previsti. Ciò fa sì che il numero di attività non sia inferiore al previsto. Per interrompere l'override della previsione, utilizza una seconda operazione pianificata per riportare il numero di attività minimo 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 policy di dimensionamento predittivo, utilizzare invece la modalità di sola previsione. In modalità di sola previsione, il dimensionamento predittivo 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:

- Inserire il nome del cluster nel parametro `resource-id`.
- Inserire 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 alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.

3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Scegliere Dimensionamento automatico del servizio.

Si apre la pagina delle policy.

5. Scegliere Operazioni pianificate e poi scegliere Crea.

Si apre la pagina Crea operazione pianificata.

6. Per Nome operazione, inserire 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, consultare [List of tz database time zones](#).

8. Per Ora di inizio, inserire la data e l'ora di inizio dell'operazione.

9. Per Recurrence (Ricorrenza), scegliere Once (Una tantum).

10. In Regolazione delle attività, per Minimo, inserire un valore inferiore o uguale al numero massimo di attività.

11. Scegli Crea operazione pianificata.

Si apre la pagina delle policy.

12. Configurare una seconda operazione programmata per ripristinare l'impostazione originale del numero di attività minimo alla fine dell'evento. Il dimensionamento predittivo può dimensionare 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 operazioni pianificate, consultare [Usa azioni pianificate per scalare i servizi Amazon ECS](#).

Configurazioni avanzate delle policy di dimensionamento predittivo utilizzando parametri personalizzati per Amazon ECS

In una policy di dimensionamento predittivo, è possibile utilizzare parametri predefiniti o personalizzati. I parametri personalizzati sono utili quando i parametri predefiniti, ad esempio la CPU, la memoria, ecc.) non sono sufficienti per descrivere sufficientemente il carico dell'applicazione.

Quando crei una politica di scalabilità predittiva con metriche personalizzate, puoi specificare altre metriche fornite da CloudWatch AWS. In alternativa, è possibile specificare le metriche definite e pubblicate autonomamente. Puoi anche utilizzare la matematica metrica per aggregare e trasformare le metriche esistenti in una nuova serie temporale che non viene tracciata automaticamente. AWS Un esempio consiste nel combinare valori nei dati calcolando nuove somme o medie, un processo detto aggregazione. 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 da te, 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:

- Il parametro più utile per la specifica del parametro di carico è un parametro che rappresenta il carico su un gruppo Auto Scaling nel suo complesso.
- Il parametro più utile per la specifica del parametro di dimensionamento con cui effettuare la scalabilità è il throughput medio o utilizzo medio per parametro di attività.
- L'utilizzo di destinazione deve corrispondere al tipo di parametro di dimensionamento. Per una configurazione di policy che usa l'utilizzo della CPU, questa è ad esempio una percentuale di destinazione.
- 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 gli oggetti restituiti da una chiamata all'`LoadForecastAPI`.
[GetPredictiveScalingForecast](#)
- Ti suggeriamo vivamente di configurare il dimensionamento predittivo in modalità Solo previsione, in modo da poter valutare il Forecast prima che il dimensionamento predittivo inizi attivamente la capacità.

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 policy di dimensionamento predittivo con parametri personalizzati

Se si verifica un problema durante l'utilizzo di parametri personalizzati, si consiglia di procedere come segue:

- Se riscontri un problema in una blue/green distribuzione 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 anche necessario verificare che la query trovi solo i gruppi Auto Scaling che eseguono l'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, vi è la possibilità che questo comando non riesca a identificare la causa esatta degli errori rilevati. Per risolvere i problemi, risolvetevi gli errori che ricevete in risposta a una richiesta

al comando. [get-metric-data](#) 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 il dimensionamento predittivo 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 delle metriche con policy di dimensionamento predittivo nella policy.

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 si sceglie di utilizzare un'espressione matematica dei parametri nella policy di dimensionamento predittivo, considerare i seguenti punti:

- Le operazioni matematiche metriche utilizzano i punti dati della combinazione univoca di nome della metrica, namespace e coppie di metriche dimensionali. `keys/value`
- È possibile 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": []
}
```

Interconnessione dei 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 “Usare il bilanciamento del carico per distribuire il traffico del 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 il rilevamento 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 si utilizza Service Connect, Amazon ECS gestisce tutte le parti del rilevamento servizi: crea i nomi che possono essere rilevati, gestisce dinamicamente le voci per ogni attività all'inizio e all'interruzione, esegue un agente in ogni attività configurato per rilevare 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 modificare l'applicazione per utilizzare Service Connect.

Si fornisce la configurazione completa all'interno di ogni definizione di servizio e attività. Amazon ECS gestisce le modifiche a questa configurazione in ogni implementazione del servizio, per garantire che tutte le attività di un'implementazione si comportino nello stesso modo. Ad esempio, un problema comune relativo al DNS come rilevamento servizi 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, l'implementazione del client aggiorna la configurazione sostituendo le attività del client. È possibile configurare l'interruttore automatico di implementazione e altre configurazioni di

implementazione per influire sulle modifiche di Service Connect allo stesso modo di qualsiasi altra implementazione.

Per ulteriori informazioni, consultare [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, si può utilizzare l'integrazione del rilevamento servizi di AWS Cloud Map 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 nel VPC di Amazon possono utilizzare questo nome host DNS per inviare il traffico direttamente a un altro container utilizzando il suo indirizzo IP interno.

Questo approccio alla comunicazione offre una bassa latenza. service-to-service Non ci sono componenti aggiuntivi tra i container. Il traffico viaggia direttamente da un container all'altro.

Questo approccio è adatto quando si utilizza la modalità di rete `awsvpc`, in cui ogni attività ha il proprio indirizzo IP univoco. La maggior parte dei software supporta solo l'uso di record A del DNS, che si risolvono direttamente negli indirizzi IP. Quando si utilizza la modalità di rete `awsvpc`, l'indirizzo IP per ogni attività è un record A. Tuttavia, se si utilizza la modalità di rete `bridge`, è possibile che più container condividano lo stesso indirizzo IP. Inoltre, le mappature dinamiche delle porte fanno sì che ai container vengano assegnati in modo casuale i numeri di porta su quel singolo indirizzo IP. A questo punto, un record A non è più sufficiente per il rilevamento servizi. È inoltre necessario utilizzare un record SRV. 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 i record SRV.

Un altro vantaggio della modalità di rete `awsvpc` 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.

Il principale svantaggio della service-to-service comunicazione diretta che utilizza il 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 tale che punti a un altro container non più presente. L'applicazione deve gestire i nuovi tentativi e disporre di una logica per ignorare i backend non validi.

Per ulteriori informazioni, consultare [Usare il rilevamento servizi per connettere i servizi Amazon ECS con nomi DNS](#)

- Amazon VPC Lattice

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 di destinazione 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 di destinazione di VPC Lattice, i clienti possono ora abilitare le attività Amazon ECS come destinazioni IP in VPC Lattice. Quando vengono avviate le attività per il servizio registrato, Amazon ECS le registra automaticamente nel gruppo di destinazione di VPC Lattice.

Per ulteriori informazioni, consultare [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.

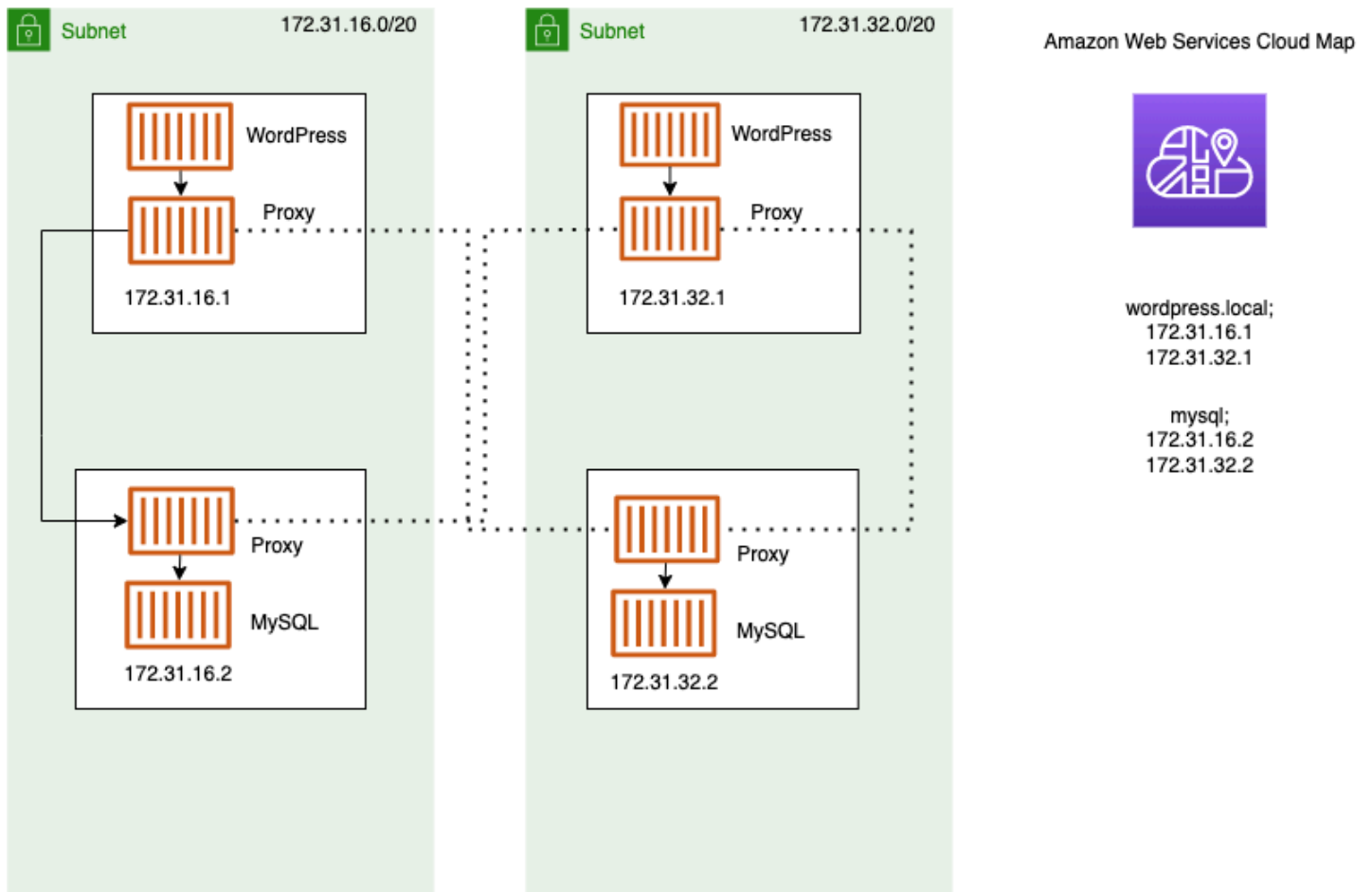
Opzioni di interconnessione	Collegato	awsvpc	Host
Individuazione dei servizi	sì, ma richiede che i clienti conoscano i record SRV nel DNS senza hostPort.	sì	sì, ma richiede che i clienti conoscano i record SRV nel DNS senza hostPort.
Service Connect	sì	sì	no

Opzioni di interconnessione	Collegato	awsvpc	Host
Reticolo in VPC	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 il rilevamento servizi che un mesh di servizi in Amazon ECS. Ciò fornisce la configurazione completa all'interno di ogni servizio gestito in base alle implementazioni dei servizi, un modo unificato per fare riferimento ai servizi all'interno di namespace che non dipendono dalla configurazione DNS del VPC e metriche e log standardizzati per monitorare tutte le 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 namespace.

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.

namespace

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. Lo spazio dei nomi può essere condiviso con Account AWS using AWS Resource Access Manager (AWS RAM) in quanto è disponibile Regioni AWS in AWS RAM . Per ulteriori informazioni sui namespace condivisi, consultare [Cross-account AWS Cloud Map namespace sharing](#) nella Guida per gli sviluppatori di AWS Cloud Map .

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 namespace. I servizi all'interno di un namespace possono essere distribuiti su diversi cluster Amazon ECS all'interno della stessa Regione AWS. Se il namespace è un namespace condiviso, i servizi possono essere distribuiti tra il proprietario del namespace e il consumer del namespace 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 un namespace configurato. Ogni attività del servizio può rilevare e connettersi a tutti gli endpoint nel namespace tramite un container proxy Service Connect.

Se uno dei container dell'attività deve connettersi a un endpoint da un servizio in un namespace, scegli un servizio client. Se un'applicazione frontend, un proxy inverso o un bilanciatore 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 un namespace 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 nel medesimo namespace, ma la configurazione del client è necessaria. Un backend, un middleware (software intermediario), un livello aziendale o la maggior parte dei microservizi possono utilizzare questo tipo di configurazione Service Connect. Se si desidera che un'applicazione frontend, un proxy

inverso o un bilanciatore del carico riceva traffico da altri servizi configurati con Service Connect nello stesso namespace, 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 container del proxy nell'attività.

Service Connect è adatto per connessioni tra servizi Amazon ECS all'interno dello stesso namespace. 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 che non sono 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 Amazon ECS, è necessario utilizzare un altro metodo di interconnessione.

Argomenti

- [Prezzi](#)
- [Componenti di Amazon ECS Service Connect](#)
- [Panoramica della configurazione di Amazon ECS Service Connect](#)
- [Amazon ECS Service Connect con namespace AWS Cloud Map condivisi](#)
- [Log di accesso di Amazon ECS Service Connect](#)
- [Crittografare il traffico di 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, consultare [Prezzi di Amazon ECS](#).
- L'utilizzo di Amazon ECS Service Connect non comporta costi supplementari.
- Non sono previsti costi aggiuntivi per AWS Cloud Map l'utilizzo se utilizzato da Service Connect.
- I clienti pagano per le risorse di calcolo 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 Service di Amazon ECS.
- 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 maggiori dettagli, consultare [Prezzi di AWS Autorità di certificazione privata](#). 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 al mese per servizio Amazon ECS.

Componenti di Amazon ECS Service Connect

Quando si utilizza Amazon ECS Service Connect, viene configurato 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 di Service Connect nel namespace. 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 aumentando orizzontalmente, le nuove connessioni dai client verranno bilanciate tra tutte le attività. Se un servizio client-server viene aggiornato, le nuove connessioni dai 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 che hanno una configurazione Service Connect nello stesso namespace e che iniziano a essere eseguite dopo questa implementazione possono essere risolte 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 implementato un servizio nel namespace. Le attività esistenti e quelle sostitutive continuano a comportarsi allo stesso modo dopo l'implementazione più recente.

Considera i seguenti esempi:

Innanzitutto, supponi di creare un'applicazione disponibile sulla rete Internet pubblica in un unico AWS CloudFormation modello e in un unico stack. CloudFormation La scoperta pubblica e la raggiungibilità dovrebbero essere create per ultimo 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 frontend è attivo e disponibile al pubblico, ma un backend 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 CloudFormation che non è possibile creare più servizi Amazon ECS in parallelo o contemporaneamente. È necessario aggiungere `dependsOn` al servizio client di frontend per ogni servizio client-server di backend a cui si connettono le attività del client.

In secondo luogo, supponiamo che esista un servizio di frontend senza la configurazione di Service Connect. Le attività si connettono a un servizio di backend esistente. Aggiungi prima una configurazione di Service Connect client-server al servizio di backend utilizzando lo stesso nome nel DNS o `clientAlias` utilizzato dal frontend. 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. Console di gestione AWS AWS CLI Se sei soddisfatto delle prestazioni e del comportamento del servizio di backend, aggiungi una configurazione di Service Connect client o client-server al servizio di frontend. 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 Console di gestione AWS 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 è in ascolto sulla `containerPort` dalla mappatura della porta di definizione dell'attività. Le regole del gruppo di sicurezza devono consentire il traffico in entrata (in ingresso) 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 viene modificato questo numero di porta, è necessario consentire il traffico in entrata su questa porta utilizzata dal traffico verso 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 namespace. Le applicazioni di questi due servizi possono risolvere ogni endpoint nel namespace 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 di rete `bridge`, 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 si crea o si aggiorna un servizio con la configurazione di Service Connect, Amazon ECS aggiunge un nuovo container 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 di container 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 e alla memoria dell'attività per il container del 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 di log nella configurazione di 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 nel namespace o 2.000 attività in totale su tutti i servizi Amazon ECS all'interno del namespace, 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 nel namespace.

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 container in modo che le applicazioni si connettano al proxy solo quando l'applicazione è connessa ai nomi degli endpoint nello stesso namespace. 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.

Sistema di 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 namespace. Nell' WordPress operazione, l' WordPress applicazione si connette al database utilizzando il nome dell'endpoint. Le connessioni con questo nome vengono instradate al proxy in esecuzione in un container sidecar per 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.

Usando l'esempio precedente, il proxy può connettersi a una delle due 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 15 secondi, ma può essere aggiornato.

Parametri facoltativi:

`idleTimeout`: la quantità di tempo in secondi in cui una connessione rimane attiva mentre è inattiva. Un valore di 0 disabilita il `idleTimeout`

Il valore predefinito `idleTimeout` per HTTP/HTTP2/GRPC è 5 minuti.

Il valore predefinito `idleTimeout` per TCP è 1 ora.

`perRequestTimeout`- Il periodo di attesa che l'upstream risponda con una risposta completa per richiesta. Un valore di 0 si spegne in `perRequestTimeout`. Questo può essere impostato solo quando il `appProtocol` per il container dell'applicazione è HTTP/HTTP2/GRPC. L'impostazione predefinita è 15 secondi.

 Note

Se `idleTimeout` è impostato su un valore inferiore a `perRequestTimeout`, la connessione si chiuderà quando viene raggiunto il `idleTimeout`, non il `perRequestTimeout`.

Considerazioni

Quando si utilizza Service Connect, si consideri quanto segue:

- Per utilizzare Service Connect, le attività eseguite in Fargate devono utilizzare la piattaforma Fargate Linux versione 1.4.0 o successiva.
- La versione dell'agente Amazon ECS sull'istanza di container deve essere 1.67.2 o successiva.
- 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, consultare [Ruolo IAM delle istanze di container Amazon ECS](#).
- Le attività che utilizzano la modalità di rete `bridge` e utilizzano Service Connect non supportano il parametro di definizione del container `hostname`.
- Per utilizzare Service Connect, le definizioni di attività devono impostare il limite di memoria dell'attività. Per ulteriori informazioni, consultare [Proxy Service Connect](#).
- Le definizioni delle attività che impostano i limiti di memoria del container 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 che si trova nella stessa area Account AWS o è condiviso con l'utente Account AWS . AWS Resource Access Manager Per ulteriori informazioni sull'uso dei namespace condivisi, consultare [Amazon ECS Service Connect con namespace AWS Cloud Map condivisi](#).
- 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 un namespace.
- Tutti i nomi di rilevamento devono essere univoci all'interno di un namespace.
- È necessario implementare nuovamente i servizi esistenti prima che le applicazioni possano risolvere nuovi endpoint. I nuovi endpoint aggiunti al namespace dopo l'implementazione più recente non verranno aggiunti alla configurazione dell'attività. Per ulteriori informazioni, consultare [the section called “Componenti di 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 nella modalità di rete awsvpc. Se si desidera che il traffico non di servizio aggiri l'agente Service Connect, utilizzare il parametro [ingressPortOverride](#) nella configurazione del servizio Service Connect.
- Service Connect con TLS non supporta la modalità di rete bridge. Solo la modalità di rete awsvpc è supportata.
- In awsvpc modalità, il proxy Service Connect inoltra il traffico all'IPv4 host locale 127.0.0.1 per i servizi nelle configurazioni IPv4 -only e dualstack. Per i servizi in configurazione IPv6 -only, il proxy inoltra il traffico al localhost. IPv6 ::1 Ti consigliamo di configurare le applicazioni per ascoltare entrambi gli host locali per un'esperienza senza interruzioni quando un servizio viene aggiornato dalla configurazione -only o dualstack a IPv4 -only. IPv6 Per ulteriori informazioni, consultare [Opzioni di task networking di Amazon ECS per EC2](#) e [Opzioni di rete di attività di Amazon ECS per Fargate](#).

Service Connect non supporta quanto segue:

- Container Windows
- HTTP 1.0
- Attività autonome
- Servizi che utilizzano i tipi di distribuzione powered by ed esterni blue/green CodeDeploy
- Le istanze di container External per Amazon ECS Anywhere non sono supportate con Service Connect.
- PPv2
- modalità FIPS

Panoramica della configurazione di Amazon ECS Service Connect

Quando si utilizza Service Connect, è necessario configurare alcuni parametri nelle risorse.

Nella tabella seguente vengono descritti i parametri di configurazione per le risorse di Amazon ECS.

Posizione dei parametri	Tipo di app	Description	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, consultare 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 il namespace a cui	Sì

Posizione dei parametri	Tipo di app	Description	Richiesto
		aderire. Questo namespace deve contenere tutti i servizi del server che questo servizio deve individuare. Per ulteriori informazioni, consultare serviceConnectConfiguration —.	
Definizione di servizio	Client-server	I servizi server devono aggiungere una <code>serviceConnectConfiguration</code> per configurare i nomi DNS, i numeri di porta e il namespace da cui è disponibile il servizio. Per ulteriori informazioni, consultare serviceConnectConfiguration .	Sì
Cluster	Client	I cluster possono aggiungere un namespace Service Connect predefinito. Quando Service Connect è configurato in un servizio, i nuovi servizi nel cluster ereditano il namespace.	No
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

Nella procedura seguente viene illustrato come configurare Service Connect.

Important

- Service Connect crea AWS Cloud Map servizi nel tuo account. La modifica manuale di queste AWS Cloud Map risorse mediante registering/deregistering istanze, la modifica

degli attributi dell'istanza o l'eliminazione di un servizio può comportare un comportamento imprevisto del traffico dell'applicazione o delle successive implementazioni.

- Service Connect non supporta i link 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 AWS Cloud Map spazio dei nomi, usa uno spazio dei nomi condiviso o crea lo spazio dei nomi separatamente. Per un'organizzazione semplice, crea un cluster con il nome desiderato per il namespace e specifica lo stesso nome per il namespace. In questo caso, Amazon ECS crea un nuovo namespace HTTP con la configurazione necessaria. Service Connect non utilizza né crea zone ospitate DNS in Amazon Route 53.
3. Configurare i servizi per creare endpoint Service Connect all'interno del namespace.
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 namespace 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 un namespace predefinito per Service Connect al momento della creazione o dell'aggiornamento del cluster. Il nome dello spazio dei nomi specificato come predefinito può trovarsi nello stesso account oppure nello stesso Regione AWS Regione AWS e condiviso da un altro Account AWS utente. AWS Resource Access Manager

Se si crea un cluster e si specifica un namespace Service Connect predefinito, il cluster attende lo stato PROVISIONING mentre Amazon ECS crea il namespace. È possibile vedere un attachment nello stato del cluster che mostra lo stato del namespace. Per impostazione predefinita, gli allegati

non vengono visualizzati in. È necessario aggiungerli `--include ATTACHMENTS` per visualizzarli.

AWS CLI

Se desideri utilizzare uno spazio dei nomi condiviso con i tuoi utenti AWS IAM, specifica Account AWS l'Amazon Resource Name (ARN) dello spazio dei nomi nella configurazione del cluster. Per ulteriori informazioni sugli spazi dei nomi condivisi, consulta. AWS Cloud Map [Amazon ECS Service Connect con namespace AWS Cloud Map condivisi](#)

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, devi attivare Service Connect e selezionare uno spazio dei nomi nel tuo Account AWS o uno spazio dei nomi condiviso per creare un servizio client. 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.

Configurazione iniziale di Health Check

Service Connect garantisce l'integrità delle attività prima di indirizzare il traffico verso di esse. All'avvio di un'attività (durante le distribuzioni, il ridimensionamento o le sostituzioni), Service Connect monitora lo stato dell'attività per garantire che sia pronta ad accettare il traffico. È necessario definire i controlli di integrità per il contenitore essenziale nella definizione dell'attività per abilitare questo comportamento.

Il comportamento del controllo sanitario iniziale tiene conto dei potenziali ritardi nel raggiungere lo stato in cui un'attività è pronta ad accettare traffico:

- Se un'attività lo è `HEALTHY`, è immediatamente disponibile per il traffico.
- Se lo stato di salute di un'attività è lo stesso `UNKNOWN`, Service Connect segue la configurazione del controllo dello stato del contenitore (vedi [the section called "Controllo dell'integrità"](#)) dei contenitori essenziali dell'attività per calcolare un timeout, fino a `8 minutes`, prima di renderlo disponibile al traffico, anche se rimane `UNKNOWN`.
- Se un'attività lo è `UNHEALTHY`, Amazon ECS può avviare attività sostitutive. Se non sono disponibili attività funzionanti, è possibile che la distribuzione venga ripristinata in base alla configurazione del servizio.

Per tutto il traffico in corso, Service Connect utilizza controlli di integrità passivi basati sul rilevamento dei valori anomali per indirizzare il traffico in modo efficiente.

Amazon ECS Service Connect con namespace AWS Cloud Map condivisi

Amazon ECS Service Connect supporta l'uso di AWS Cloud Map namespace condivisi tra più Account AWS nomi all'interno dello stesso. Regione AWS Questa funzionalità consente di creare applicazioni distribuite in cui i servizi eseguiti in aree diverse Account AWS possono rilevarsi e comunicare tra loro tramite Service Connect. I namespace condivisi vengono gestiti utilizzando AWS Resource Access Manager (AWS RAM), che consente la condivisione sicura delle risorse tra account. Per ulteriori informazioni sugli spazi dei nomi condivisi, consulta [Condivisione AWS Cloud Map dello spazio dei nomi tra account nella Guida per gli sviluppatori](#).AWS Cloud Map

Important

È necessario utilizzare l'autorizzazione gestita da `AWSRAMPermissionCloudMapECSFullPermission` per condividere il namespace affinché Service Connect funzioni correttamente con il namespace.

Quando si utilizzano AWS Cloud Map namespace condivisi con Service Connect, i servizi di più nomi Account AWS possono partecipare allo stesso spazio dei nomi di servizio. Ciò è particolarmente utile per le organizzazioni con più account Account AWS che devono mantenere la service-to-service comunicazione oltre i confini degli account preservando al contempo la sicurezza e l'isolamento.

Note

Per comunicare con servizi diversi VPCs, è necessario configurare la connettività inter-VPC. Ciò si può fare utilizzando una connessione peering VPC. Per ulteriori informazioni, consultare [Create or delete a VPC Peering connection](#) nella Guida al peering VPC di Amazon Virtual Private Cloud.

Utilizzo di AWS Cloud Map namespace condivisi con Amazon ECS Service Connect

La configurazione di AWS Cloud Map namespace condivisi per Service Connect prevede i seguenti passaggi: il proprietario del namespace crea lo spazio dei nomi, il proprietario lo condivide tramite AWS Resource Access Manager (AWS RAM), il consumatore accetta la condivisione delle risorse e il consumatore configura Service Connect per l'utilizzo dello spazio dei nomi condiviso.

Fase 1: Creare il namespace AWS Cloud Map

Il proprietario dello spazio dei nomi crea uno spazio dei nomi che verrà condiviso con altri account.
AWS Cloud Map

Per creare uno spazio dei nomi per la condivisione utilizzando il Console di gestione AWS

1. Apri la AWS Cloud Map console all'indirizzo. <https://console.aws.amazon.com/cloudmap/>
2. Selezionare Crea namespace.
3. Inserire un Nome del namespace. Questo nome verrà utilizzato dai servizi in tutti gli account partecipanti.
4. Per Tipo di namespace, scegliere il tipo appropriato per il caso d'uso:
 - Chiamate API: namespace HTTP per il rilevamento servizi senza funzionalità DNS.
 - Chiamate API e query DNS in VPCs - Namespace DNS privati per l'individuazione di servizi con query DNS private in un VPC.
 - Chiamate API e query DNS pubbliche: namespace DNS pubblici per il rilevamento servizi con query DNS pubbliche.
5. Selezionare Crea namespace.

Passaggio 2: condividi lo spazio dei nomi utilizzando AWS RAM

Il proprietario dello spazio dei nomi utilizza AWS RAM per condividere lo spazio dei nomi con altri.
Account AWS

Per condividere uno spazio dei nomi utilizzando la console AWS RAM

1. Apri la AWS RAM console all'indirizzo. <https://console.aws.amazon.com/ram/>
2. Seleziona Crea condivisione risorse.
3. Per Nome, inserire un nome descrittivo della risorsa da condividere.
4. Nella sezione Risorse:
 - a. Per Tipo di risorsa, scegliere Namespace della mappa cloud.
 - b. Selezionare il namespace creato nella fase precedente.
5. Nella sezione Autorizzazioni gestite, specifica
AWSRAMPermissionCloudMapECSFullAutorizzazione.

⚠ Important

È necessario utilizzare l'autorizzazione gestita da `AWSRAMPermissionCloudMapECSFullPermission` per condividere il namespace affinché Service Connect funzioni correttamente con il namespace.

6. Nella sezione Principali, specificare gli Account AWS con cui condividere il namespace. Puoi inserire l'account IDs o l'unità IDs organizzativa.
7. Seleziona Crea condivisione risorse.

Fase 3: Accettare la condivisione delle risorse

Gli account consumer del namespace devono accettare l'invito alla condivisione delle risorse per utilizzare il namespace condiviso.

Per accettare un invito alla condivisione di risorse utilizzando la AWS RAM console

1. Nell'account consumer, apri la AWS RAM console all'indirizzo <https://console.aws.amazon.com/ram/>.
2. Nel riquadro di navigazione, scegliere Condivise con me, poi scegliere Condivisioni di risorse.
3. Selezionare l'invito alla condivisione delle risorse e scegliere Accetta la condivisione delle risorse.
4. Dopo aver accettato, annotare l'ARN del namespace condiviso dai dettagli della risorsa. Si utilizzerà questo ARN per configurare i servizi Service Connect.

Fase 4: Configurare un servizio Amazon ECS con il namespace condiviso

Dopo aver accettato il namespace condiviso, il consumer del namespace può configurare i servizi Amazon ECS per utilizzare il namespace condiviso. La configurazione è simile all'utilizzo di un namespace normale, ma è necessario specificare l'ARN del namespace anziché il nome. Per una procedura dettagliata di creazione del servizio, consultare [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#).

Per creare un servizio con uno spazio dei nomi condiviso utilizzando il Console di gestione AWS

1. [Aprire la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Sulla pagina Cluster, scegliere il cluster in cui creare il servizio.

3. In Servizi, scegliere Crea.
4. Dopo aver inserito altri dettagli a seconda del carico di lavoro, nella sezione Service Connect, scegliere Usa Service Connect.
5. Per Namespace, inserire l'ARN completo del namespace condiviso.

Il formato ARN è: `arn:aws:servicediscovery:region:account-id:namespace/namespace-id`

6. Configurare le impostazioni rimanenti di Service Connect in base alle esigenze per il tipo di servizio (client o client-server).
7. Completare il processo di creazione del servizio.

È inoltre possibile configurare i servizi utilizzando AWS CLI o AWS SDKs specificando lo spazio dei nomi condiviso ARN nel namespace parametro di `serviceConnectConfiguration`

```
aws ecs create-service \
  --cluster my-cluster \
  --service-name my-service \
  --task-definition my-task-def \
  --service-connect-configuration '{
    "enabled": true,
    "namespace": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
abcdef1234567890",
    "services": [{
      "portName": "web",
      "discoveryName": "my-service",
      "clientAliases": [{
        "port": 80,
        "dnsName": "my-service"
      }]
    }]
  }'
```

Considerazioni

Considerate quanto segue quando utilizzate AWS Cloud Map namespace condivisi con Service Connect:

- AWS RAM deve essere disponibile nel luogo in Regione AWS cui desideri utilizzare lo spazio dei nomi condiviso.

- Lo spazio dei nomi condiviso deve trovarsi nello stesso dei servizi Regione AWS e dei cluster Amazon ECS.
- È necessario utilizzare l'ARN del namespace, non l'ID, quando si configura Service Connect con un namespace condiviso.
- Sono supportati tutti i tipi di namespace: HTTP, DNS privato e namespace DNS pubblico.
- Se l'accesso a un namespace condiviso viene revocato, le operazioni di Amazon ECS che richiedono l'interazione con lo il namespace (come `CreateService`, `UpdateService` e `ListServicesByNamespace`) avranno esito negativo. Per ulteriori informazioni sulle autorizzazioni per la risoluzione dei problemi relativi ai namespace condivisi, consultare [Risoluzione dei problemi di Amazon ECS Service Connect con namespace condivisi AWS Cloud Map](#).
- Per il rilevamento servizi tramite query DNS in un namespace DNS privato condiviso:
 - Il proprietario del namespace dovrà chiamare `create-vpc-association-authorization` con l'ID della zona ospitata privata associata al namespace e il VPC del consumer.

```
aws route53 create-vpc-association-authorization --hosted-zone-id Z1234567890ABC --vpc VPCRegion=us-east-1,VPCId=vpc-12345678
```

- Il consumer del namespace dovrà chiamare `associate-vpc-with-hosted-zone` con l'ID della zona ospitata privata.

```
aws route53 associate-vpc-with-hosted-zone --hosted-zone-id Z1234567890ABC --vpc VPCRegion=us-east-1,VPCId=vpc-12345678
```

- Solo il proprietario del namespace può gestire la condivisione delle risorse.
- I consumer del namespace possono creare e gestire servizi all'interno del namespace condiviso ma non possono modificare il namespace stesso.
- I nomi Discovery devono essere univoci all'interno del namespace condiviso, indipendentemente dall'account che crea il servizio.
- I servizi nello spazio dei nomi condiviso possono scoprire e connettersi ai servizi di altri AWS account che hanno accesso allo spazio dei nomi.
- Quando si abilita TLS per Service Connect e si utilizza un namespace condiviso, l'Autorità di certificazione (CA) AWS Private CA ha l'ambito del namespace. Quando l'accesso al namespace condiviso viene revocato, l'accesso alla CA viene interrotto.
- Quando si lavora con uno spazio dei nomi condiviso, i proprietari e i consumatori di namespace non hanno accesso ai parametri Amazon tra account diversi per impostazione predefinita.

CloudWatch Le metriche di Target vengono pubblicate solo per gli account che possiedono servizi client. Un account proprietario di servizi client non ha accesso alle metriche ricevute da un account che possiede servizi client-server e viceversa. Per consentire l'accesso alle metriche da più account, configura l'osservabilità tra account. CloudWatch Per ulteriori informazioni sulla configurazione dell'osservabilità tra account, consulta l'osservabilità [CloudWatch tra account nella Amazon User Guide](#). CloudWatch Per ulteriori informazioni sulle CloudWatch metriche per Service Connect, vedere [Metriche di Amazon ECS CloudWatch](#).

Risoluzione dei problemi di Amazon ECS Service Connect con namespace condivisi AWS Cloud Map

Utilizza le seguenti informazioni per risolvere i problemi relativi ai AWS Cloud Map namespace condivisi e Service Connect. Per ulteriori informazioni sull'individuazione dei messaggi di errore, consultare [Risoluzione dei problemi di Amazon ECS](#).

I messaggi di errore relativi ai problemi di autorizzazione vengono visualizzati a causa di autorizzazioni mancanti o se l'accesso al namespace viene revocato.

Important

È necessario utilizzare l'autorizzazione gestita da `AWSRAMPermissionCloudMapECSFullPermission` per condividere il namespace affinché Service Connect funzioni correttamente con il namespace.

Il messaggio di errore appare in uno dei seguenti formati:

Si è verificato un errore (ClientException) durante la chiamata all'operazione < OperationName >:
User: arn:aws:iam: :user/ non è autorizzato a eseguire: < > ActionName on resource: < > perché nessuna politica basata sulle risorse consente l'azione < > ResourceArn ActionName <account-id><user-name>

I seguenti scenari possono generare un messaggio di errore in questo formato:

Errore di creazione o aggiornamento del cluster

Questi problemi si verificano quando le operazioni di Amazon ECS `UpdateCluster` non funzionano `CreateCluster` o falliscono a causa di AWS Cloud Map autorizzazioni mancanti. Le operazioni richiedono le autorizzazioni per le seguenti azioni: AWS Cloud Map

- `servicediscovery:GetNamespace`

Assicurarsi che l'invito alla condivisione delle risorse sia stato accettato nell'account consumer e che nella configurazione di Service Connect venga utilizzato l'ARN del namespace corretto.

Errore di creazione o aggiornamento del servizio

Questi problemi si verificano quando le operazioni di Amazon ECS UpdateService non funzionano CreateService o falliscono a causa di AWS Cloud Map autorizzazioni mancanti. Le operazioni richiedono le autorizzazioni per le seguenti azioni: AWS Cloud Map

- `servicediscovery:CreateService`
- `servicediscovery:GetNamespace`
- `servicediscovery:GetOperation` (per creare un nuovo servizio AWS Cloud Map)
- `servicediscovery:GetService` (per quando un servizio AWS Cloud Map esiste già)

Assicurarsi che l'invito alla condivisione delle risorse sia stato accettato nell'account consumer e che nella configurazione di Service Connect venga utilizzato l'ARN del namespace corretto.

L'operazione **ListServicesByNamespace** ha esito negativo

Questo problema si verifica quando l'operazione ListServicesByNamespace di Amazon ECS ha esito negativo. Questa operazione richiede autorizzazioni per le seguenti azioni AWS Cloud Map :

- `servicediscovery:GetNamespace`

Per risolvere il problema:

- Verificare che l'account consumer disponga dell'autorizzazione `servicediscovery:GetNamespace`.
- Usare l'ARN del namespace quando si chiama l'API, non il nome.
- Assicurarsi che la condivisione delle risorse sia attiva e che l'invito sia stato accettato.

Utente: non è autorizzato a eseguire: < ActionName > sulla risorsa: < ResourceArn > con una negazione esplicita in una politica basata sull'identità. <iam-user>

I seguenti scenari possono generare un messaggio di errore in questo formato:

L'eliminazione del servizio ha esito negativo e rimane bloccata nello stato **DRAINING**

Questo problema si verifica quando le operazioni DeleteService di Amazon ECS hanno esito negativo a causa dell'autorizzazione `servicediscovery:DeleteService` mancante quando

l'accesso al namespace viene revocato. Inizialmente potrebbe sembrare che il servizio sia stato eliminato correttamente, ma rimarrà bloccato nello stato DRAINING. Il messaggio di errore viene visualizzato come un evento del servizio di Amazon ECS.

Per risolvere questo problema, il proprietario del namespace deve condividere quest'ultimo con l'account consumer per consentire il completamento dell'eliminazione del servizio.

Le attività in servizio non vengono eseguite

Questo problema si verifica quando le attività non vengono avviate a causa della mancanza di autorizzazioni. Il messaggio di errore viene visualizzato come errore di attività interrotta. Per ulteriori informazioni, consulta [Risolvi gli errori relativi alle attività interrotte in Amazon ECS](#).

Le seguenti AWS Cloud Map azioni sono necessarie per eseguire un'attività:

- `servicediscovery:GetOperation`
- `servicediscovery:RegisterInstance`

Assicurarsi che l'account consumer disponga delle autorizzazioni richieste e che il namespace condiviso sia accessibile.

Le attività non vengono interrotte correttamente o rimangono bloccate nello stato **DEACTIVATING** o **DEPROVISIONING**

Questo problema si verifica quando le attività non riescono ad annullare la registrazione dal AWS Cloud Map servizio durante l'arresto a causa della mancanza di autorizzazioni. L'errore viene visualizzato come `statusReason` nell'allegato dell'attività che può essere recuperato utilizzando l'API `DescribeTasks`. Per ulteriori informazioni, consulta il riferimento [DescribeTasks](#) all'API di Amazon Elastic Container Service.

Per interrompere un'attività sono necessarie le seguenti AWS Cloud Map azioni:

- `servicediscovery:DeregisterInstance`
- `servicediscovery:GetOperation`

Se l'accesso al namespace viene revocato, le attività possono rimanere in uno stato DEACTIVATING o DEPROVISIONING fino al ripristino dell'accesso al namespace. Richiedere al proprietario del namespace di ripristinare l'accesso al namespace.

Log di accesso di Amazon ECS Service Connect

Amazon ECS Service Connect supporta i log di accesso per fornire telemetria dettagliata sulle singole richieste elaborate dal proxy Service Connect. I log di accesso completano i log delle applicazioni esistenti acquisendo metadati sul traffico per richiesta come metodi HTTP, percorsi, codici di risposta, flag e informazioni temporali. Ciò consente una maggiore osservabilità dei modelli di traffico a livello di richiesta e delle interazioni di servizio per una risoluzione e un monitoraggio efficaci.

Per abilitare i log di accesso, specificate sia gli oggetti che `logConfiguration` gli `accessLogConfiguration` oggetti nell'oggetto. `serviceConnectConfiguration` È possibile configurare il formato dei log e se i log devono includere i parametri di interrogazione in. `accessLogConfiguration` I log vengono consegnati al gruppo di log di destinazione dal driver di registro specificato in. `logConfiguration`

```
{
  "serviceConnectConfiguration": {
    "enabled": true,
    "namespace": "myapp.namespace",
    "services": [
      ...
    ],
    "logConfiguration": {
      "logDriver": "awslogs",
      "options": {
        "awslogs-group": "my-envoy-log-group",
        "awslogs-region": "us-west-2",
        "awslogs-stream-prefix": "myapp-envoy-logs"
      }
    },
    "accessLogConfiguration": {
      "format": "TEXT",
      "includeQueryParameters": "ENABLED"
    }
  }
}
```

Considerazioni

Quando si abilita l'accesso ai log di accesso, si consideri quanto segue

- I registri di accesso e i registri delle applicazioni vengono entrambi scritti su. `/dev/stdout`
Per separare i log di accesso dai log delle applicazioni, si consiglia di utilizzare il driver di `awsfirelens` log con una configurazione OR personalizzata. `Fluent Bit` `Fluentd`
- Si consiglia di utilizzare il driver di `awslogs` registro per inviare i log delle applicazioni e degli accessi alla stessa destinazione. `CloudWatch`
- i registri di accesso sono supportati sui servizi Fargate che utilizzano 1.4.0 versioni della piattaforma o successive.
- I parametri di interrogazione come gli ID e i token delle richieste sono esclusi dai log di accesso per impostazione predefinita. Per includere i parametri di interrogazione nei log di accesso, impostate su. `includeQueryParameters "ENABLED"`

Formati dei log di accesso

i registri di accesso possono essere formattati in dizionari in formato JSON o stringhe in formato testo, con differenze negli operatori di comando supportati per i diversi tipi di log di accesso.

registri di accesso HTTP

I seguenti operatori di comando sono inclusi per impostazione predefinita nei log HTTP:

Text

```
[%START_TIME%] "%REQ(:METHOD)% %REQ(X-ENVOY-ORIGINAL-PATH?:PATH)% %PROTOCOL%"
%RESPONSE_CODE% %BYTES_RECEIVED% %BYTES_SENT% %DURATION%
%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)% "%REQ(X-FORWARDED-FOR)%" "%REQ(USER-AGENT)%"
"%REQ(X-REQUEST-ID)%" "%REQ(:AUTHORITY)%" "%UPSTREAM_HOST%"`n
```

JSON

```
{
  "start_time": "%START_TIME%",
  "method": "%REQ(:METHOD)%",
  "path": "%REQ(X-ENVOY-ORIGINAL-PATH?:PATH)%",
  "protocol": "%PROTOCOL%",
  "response_code": "%RESPONSE_CODE%",
  "bytes_received": "%BYTES_RECEIVED%",
  "bytes_sent": "%BYTES_SENT%",
  "duration_ms": "%DURATION%",
  "upstream_service_time": "%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)%"
```

```

"forwarded_for": "%REQ(X-FORWARDED-FOR)%",
"user_agent": "%REQ(USER-AGENT)%",
"request_id": "%REQ(X-REQUEST-ID)%",
"authority": "%REQ(:AUTHORITY)%",
"upstream_host": "%UPSTREAM_HOST%"
}

```

HTTP2 registri di accesso

Oltre agli operatori di comando inclusi per i log HTTP, i HTTP2 log includono l'`%STREAM_ID` operatore per impostazione predefinita.

Text

```

[%START_TIME%] "%REQ(:METHOD)% %REQ(X-ENVOY-ORIGINAL-PATH?:PATH)% %PROTOCOL%"
%RESPONSE_CODE% %BYTES_RECEIVED% %BYTES_SENT% %DURATION%
%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)% "%REQ(X-FORWARDED-FOR)%" "%REQ(USER-AGENT)%"
"%REQ(X-REQUEST-ID)%" "%REQ(:AUTHORITY)%" "%UPSTREAM_HOST%" "%STREAM_ID%"

```

JSON

```

{
  "start_time": "%START_TIME%",
  "method": "%REQ(:METHOD)%",
  "path": "%REQ(X-ENVOY-ORIGINAL-PATH?:PATH)%",
  "protocol": "%PROTOCOL%",
  "response_code": "%RESPONSE_CODE%",
  "bytes_received": "%BYTES_RECEIVED%",
  "bytes_sent": "%BYTES_SENT%",
  "duration": "%DURATION%",
  "upstream_service_time": "%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)%",
  "forwarded_for": "%REQ(X-FORWARDED-FOR)%",
  "user_agent": "%REQ(USER-AGENT)%",
  "request_id": "%REQ(X-REQUEST-ID)%",
  "authority": "%REQ(:AUTHORITY)%",
  "upstream_host": "%UPSTREAM_HOST%",
  "stream_id": "%STREAM_ID%"
}

```

Log di accesso gRPC

Oltre agli operatori di comando inclusi per i log HTTP, i log di accesso gRPC includono %STREAM_ID % l'operatore %GRPC_STATUS()% and per impostazione predefinita.

Text

```
[%START_TIME%] "%REQ(:METHOD)% %REQ(X-ENVOY-ORIGINAL-PATH?:PATH)% %PROTOCOL%"
%RESPONSE_CODE% %GRPC_STATUS()% %BYTES_RECEIVED% %BYTES_SENT% %DURATION%
%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)% "%REQ(X-FORWARDED-FOR)% " "%REQ(USER-AGENT)% "
"%REQ(X-REQUEST-ID)% " "%REQ(:AUTHORITY)% " "%UPSTREAM_HOST%" "%STREAM_ID%"`n
```

JSON

```
{
  "start_time": "%START_TIME%",
  "method": "%REQ(:METHOD)%",
  "path": "%REQ(X-ENVOY-ORIGINAL-PATH?:PATH)%",
  "protocol": "%PROTOCOL%",
  "response_code": "%RESPONSE_CODE%",
  "grpc_status": "%GRPC_STATUS()",
  "bytes_received": "%BYTES_RECEIVED%",
  "bytes_sent": "%BYTES_SENT%",
  "duration": "%DURATION%",
  "upstream_service_time": "%RESP(X-ENVOY-UPSTREAM-SERVICE-TIME)%",
  "forwarded_for": "%REQ(X-FORWARDED-FOR)%",
  "user_agent": "%REQ(USER-AGENT)%",
  "request_id": "%REQ(X-REQUEST-ID)%",
  "authority": "%REQ(:AUTHORITY)%",
  "upstream_host": "%UPSTREAM_HOST%",
  "stream_id": "%STREAM_ID%"
}
```

Registri di accesso TCP

I seguenti operatori di comando sono inclusi per impostazione predefinita nei registri di accesso TCP:

Text

```
[%START_TIME%] %DOWNSTREAM_REMOTE_ADDRESS% %DOWNSTREAM_REMOTE_PORT%
%BYTES_RECEIVED% %BYTES_SENT% %DURATION%
```



```
%CONNECTION_TERMINATION_DETAILS% %CONNECTION_ID%\n
```

JSON

```
{
  "start_time": "%START_TIME%",
  "downstream_remote_address": "%DOWNSTREAM_REMOTE_ADDRESS%",
  "downstream_remote_port": "%DOWNSTREAM_REMOTE_PORT%",
  "bytes_received": "%BYTES_RECEIVED%",
  "bytes_sent": "%BYTES_SENT%",
  "duration": "%DURATION%",
  "connection_termination_details": "%CONNECTION_TERMINATION_DETAILS%",
  "connection_id": %CONNECTION_ID%
}
```

Per ulteriori informazioni su questi operatori di comando, vedete Operatori di [comando nella documentazione](#) di Envoy.

Abilitazione dei log di accesso per Amazon ECS Service Connect

I log di accesso non sono abilitati per impostazione predefinita per i servizi Amazon ECS che utilizzano Service Connect. Puoi abilitare i log di accesso nei seguenti modi.

Abilitare i log di accesso utilizzando il AWS CLI

Il comando seguente mostra come abilitare i log di accesso per un servizio Amazon ECS utilizzando AWS CLI specificando un `accessLogConfiguration` quando si crea il servizio:

```
aws ecs create-service \
  --cluster my-cluster \
  --service-name my-service \
  --task-definition my-task-def \
  --service-connect-configuration '{
    "enabled": true,
    "namespace": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
abcdef1234567890",
    "services": [{
      "portName": "web",
      "discoveryName": "my-service",
      "clientAliases": [{
        "port": 80,
        "dnsName": "my-service"
```

```
    }  
  },  
  "logConfiguration": {  
    "logDriver": "awslogs",  
    "options": {  
      "awslogs-group": "my-envoy-log-group",  
      "awslogs-region": "us-west-2",  
      "awslogs-stream-prefix": "myapp-envoy-logs"  
    }  
  },  
  "accessLogConfiguration": {  
    "format": "TEXT",  
    "includeQueryParameters": "ENABLED"  
  }  
}'
```

Abilita i log di accesso utilizzando la console

Per una procedura dettagliata di creazione del servizio, consultare [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#).

Per creare un servizio con uno spazio dei nomi condiviso utilizzando Console di gestione AWS

1. [Aprire la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Sulla pagina Cluster, scegliere il cluster in cui creare il servizio.
3. In Servizi, scegliere Crea.
4. Dopo aver inserito altri dettagli a seconda del carico di lavoro, nella sezione Service Connect, scegliere Usa Service Connect.
5. Configura le impostazioni di Service Connect in base alle esigenze per il tipo di servizio (client o client-server).
6. Espandi la configurazione del registro di accesso. Per Format, scegli JSON o TEXT.
7. Per includere i parametri di interrogazione nei log di accesso, selezionate Includi parametri di interrogazione.
8. Completare il processo di creazione del servizio.

Crittografare il traffico di 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 si indirizzano i servizi Amazon ECS verso

un [AWS Autorità di certificazione privata \(AWS Private CA\)](#), Amazon ECS fornisce automaticamente certificati TLS per crittografare il traffico tra i 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 Autorità di certificazione privata TLS con crittografia 256-bit ECDSA. 2048-bit RSA Si dispone anche del pieno controllo sui certificati privati e sulle chiavi di firma per 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.

Controlli dell'integrità di Service Connect e Application Load Balancer

È possibile utilizzare Service Connect con i controlli dell'integrità di Application Load Balancer e la crittografia TLS 1.3.

Configurazione di Application Load Balancer

Configurare l'Application Load Balancer con le impostazioni seguenti:

- Configura un listener TLS con una politica di sicurezza TLS 1.3 (come `Policy-1-2-2021-06`ELBSecurity). TLS13 Per ulteriori informazioni, consultare [Security policies for your Application Load Balancer](#).
- Creare un gruppo di destinazione con le seguenti impostazioni:
 - Impostare il protocollo su HTTPS
 - Allegare il gruppo di destinazione al listener TLS
 - Configurare la porta di controllo dell'integrità in modo tale che corrisponda alla porta del container del servizio Service Connect

Configurazione di Service Connect

Configurare un servizio con le seguenti impostazioni:

- Configurare il servizio per utilizzare la modalità di rete `awsvpc`, poiché la modalità di rete `bridge` non è supportata.
- Abilita Service Connect per il servizio.
- Impostare la configurazione del bilanciatore del carico con le seguenti impostazioni:
 - Specificare il gruppo di destinazione configurato per l'Application Load Balancer
 - Impostare la porta del container in modo che corrisponda a quella del servizio Service Connect TLS
- Non impostare `ingressPortOverride` per il servizio. Per ulteriori informazioni, consulta il riferimento [ServiceConnectService](#) all'API di Amazon Elastic Container Service.

Considerazioni

Considerare quanto segue durante l'utilizzo di Application Load Balancer, TLS e Service Connect:

- Utilizzare la modalità di rete `awsvpc` anziché la modalità di rete `bridge` per i controlli dell'integrità HTTPS quando si usa Service Connect con crittografia TLS. I controlli dell'integrità HTTP continueranno a funzionare con la modalità `bridge`.
- Configurare la porta di controllo dell'integrità del gruppo di destinazione in modo che corrisponda alla porta del container del servizio Service Connect, non alla porta HTTPS predefinita (443).

AWS Autorità di certificazione privata certificati e Service Connect

È necessario disporre del ruolo IAM dell'infrastruttura. Per ulteriori informazioni su questo ruolo, consultare [Amazon ECS infrastructure IAM role](#).

AWS Autorità di certificazione privata modalità per Service Connect

AWS Autorità di certificazione privata può funzionare in due modalità: generica e di breve durata.

- Generica: 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. Si dispone dell'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 hanno esito negativo 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 dei segreti ha esito negativo, viene pubblicato un evento `RotationFailed` 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 la replica è stata già aggiunta, eseguire il comando riportato di seguito.

```
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 firmatario e radice di attendibilità 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 on-premises

Per utilizzare la CA on-premises, è necessario creare e configurare una CA subordinata in AWS Autorità di certificazione privata. Ciò garantisce che tutti i certificati TLS emessi per i carichi di lavoro di Amazon ECS condividano la catena di attendibilità con i carichi di lavoro eseguiti on-premises e siano in grado di connettersi in modo sicuro.

Important

Aggiungi il tag richiesto `AmazonECSManaged` : `true` nel tuo AWS Private CA.

Infrastructure as code (IaC)

Quando si utilizza Service Connect TLS con gli strumenti Infrastructure as Code (IaC), è importante configurare correttamente le dipendenze per evitare problemi, come i servizi bloccati nel drenaggio. La AWS KMS chiave, se fornita, il ruolo IAM e AWS Private CA le dipendenze devono essere eliminati dopo il servizio Amazon ECS.

Se lo spazio dei nomi utilizzato per Service Connect è uno spazio dei nomi condiviso, puoi scegliere di utilizzare una risorsa condivisa. AWS Private CA Per ulteriori informazioni, consultare [Attach a policy for cross-account access](#) nella Guida per l'utente di AWS Autorità di certificazione privata .

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 di 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 servizi. Tuttavia, è

importante essere consapevoli delle potenziali implicazioni per la sicurezza. Altri ruoli IAM con 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, seguire queste best practice:

- Gestire e limitare con attenzione l'accesso a Secrets Manager, in particolare per i segreti creati da Service Connect.
- Verificare regolarmente i ruoli IAM e le relative autorizzazioni per garantire il rispetto del principio del privilegio minimo.

Quando si concede l'accesso in lettura a Secrets Manager, valutare 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'operazione `GetSecretValue` a tutti i segreti con il prefisso `ecs-sc!`:

JSON

```
{
  "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. Testare sempre attentamente le

policy IAM per assicurarsi che forniscano l'accesso previsto mantenendo al contempo la sicurezza.

Comprendendo come Service Connect interagisce con Secrets Manager, si può gestire meglio la sicurezza dei 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.

JSON

```
{
  "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 policy della chiave, consultare [Creating a key policy](#) nella Guida per gli sviluppatori di AWS Key Management Service .

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, vedi [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 di TLS per Amazon ECS Service Connect

La crittografia del traffico viene abilitata quando si crea o si aggiorna un servizio di Service Connect.

Per abilitare la crittografia del traffico per un servizio in un namespace esistente utilizzando Console di gestione AWS

1. È necessario disporre del ruolo IAM dell'infrastruttura. Per ulteriori informazioni su questo ruolo, consultare [Amazon ECS infrastructure IAM role](#).
2. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
3. Nel riquadro di navigazione, selezionare Namespaces.
4. Scegliere il Namespace con il Servizio per cui si desidera abilitare la crittografia del traffico.
5. Scegliere il Servizio per cui si desidera abilitare la crittografia del traffico.
6. Scegliere Aggiorna servizio nell'angolo in alto a destra e scorrere in basso fino alla sezione Service Connect.
7. Scegliere Attiva la crittografia del traffico tra le informazioni sul servizio per abilitare TLS.

8. Per Ruolo di Service Connect TLS, scegliere un ruolo IAM dell'infrastruttura esistente o crearne uno nuovo.
9. Per Autorità di certificato del firmatario, scegliere un'autorità del certificato esistente o crearne una nuova.

Per ulteriori informazioni, consultare le [AWS Autorità di certificazione privata certificati e Service Connect](#).

10. Per Choose an AWS KMS key, scegli una chiave AWS posseduta e gestita oppure puoi scegliere una chiave diversa. Si può 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](#).

Verificare 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. Seguire questa procedura per verificare che TLS sia abilitato.

1. Includere la CLI openssl nell'immagine dell'applicazione.
2. Abilitare [ECS Exec](#) sui servizi per connettersi alle attività tramite SSM. In alternativa, puoi avviare un' EC2 istanza Amazon nello stesso Amazon VPC del servizio.
3. Recuperare l'IP e la porta di un'attività da un servizio che si desidera verificare. Puoi recuperare l'indirizzo IP dell'attività nella console. AWS Cloud Map Le informazioni si trovano nella pagina dei dettagli del servizio relativa al namespace.
4. Accedere a una qualsiasi delle attività utilizzando `execute-command` come 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, utilizzare la CLI `openssl` 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

È possibile creare un servizio Amazon ECS per un'attività Fargate che usa Service Connect con la 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:

- Verifica che sia installata e configurata la versione più recente di. AWS CLI Per ulteriori informazioni, consultare [Installing or updating to the latest version of the AWS CLI](#).
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS_FullAccess](#).
- Sono disponibili un VPC, una sottorete, una tabella di instradamento e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consultare [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

Utilizzare la procedura seguente per creare un cluster e un namespace 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 il namespace 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. Il namespace 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 Console di gestione AWS 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: Creazione del 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 servizi e il mesh di servizi che indirizza il traffico verso questo servizio. La configurazione riutilizza il namespace 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.

Creazione del 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 nel namespace.

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. Console di gestione AWS 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. Creare 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 si crea il servizio ECS, specificare 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`. Il namespace non è richiesto perché il cluster ha un namespace predefinito configurato. Le applicazioni client in esecuzione in ECS nel namespace 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 namespace 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 configurazione di `tls`, aggiungere il certificato `arn` per il `awsPcaAuthorityArn`, `kmsKey` e `roleArn` del 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. Console di gestione AWS 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,
```

```

"enableECSManagedTags": 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. Creare un servizio utilizzando il file `service-connect-nginx-service.json`:

```
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 Console di gestione AWS 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.

Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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, consultare [What Is 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
Stati Uniti orientali (Virginia settentrionale)	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 (Città del Capo)	af-south-1
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacifico (Taipei)	ap-east-2
Asia Pacifico (Mumbai)	ap-south-1
Asia Pacifico (Hyderabad)	ap-south-2
Asia Pacifico (Tokyo)	ap-northeast-1
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Singapore)	ap-southeast-1

Nome della regione	Regione
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacifico (Malesia)	ap-southeast-5
Asia Pacifico (Nuova Zelanda)	ap-southeast-6
Asia Pacifico (Thailandia)	ap-southeast-7
Canada (Centrale)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europe (Milan)	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

Nome della regione	Regione
Messico (Centrale)	mx-central-1
Medio Oriente (Bahrein)	me-south-1
Sud America (San Paolo)	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:

- **Namespace di rilevamento servizi:** un raggruppamento logico di servizi di rilevamento dei servizi che condividono lo stesso nome di dominio, ad esempio `example.com`, che rappresenta la posizione in cui si vuole instradare il traffico. È possibile creare un namespace con una chiamata al comando `aws servicediscovery create-private-dns-namespace` o nella console classica di Amazon ECS. È possibile utilizzare il comando `aws servicediscovery list-namespaces` per visualizzare le informazioni di riepilogo dei namespace creati dall'account corrente. Per ulteriori informazioni sui comandi di individuazione dei servizi, vedere [create-private-dns-namespace](#) e [list-namespaces](#) nella Guida di AWS CLI riferimento AWS Cloud Map (service discovery).
- **Servizio di individuazione dei servizi:** è incluso nel namespace di individuazione dei servizi ed è costituito dal nome del servizio e dalla configurazione DNS per il namespace. 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_IPV6**— Per un AAAA record, l'IPv6 indirizzo che Route 53 restituisce in risposta alle query DNS e AWS Cloud Map restituisce quando rileva i dettagli dell'istanza, ad esempio, `2001:0db8:85a3:0000:0000:abcd:0001:2345`. Sia `AWS_INSTANCE_IPv4` che `AWS_INSTANCE_IPv6` vengono aggiunti per i servizi dualstack di Amazon ECS. `AWS_INSTANCE_IPv6` viene aggiunto solo per i servizi IPv6 solo di Amazon ECS.
- **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 EC2, questa è la zona di disponibilità in cui si trova l'istanza del contenitore. Per i processi che utilizzano 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 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, consultare [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.
- Amazon ECS non supporta la registrazione di servizi in namespace condivisi AWS Cloud Map .
- 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 si utilizzano i record SRV, è necessaria una porta. Inoltre, è possibile creare record AAAA se il servizio utilizza sottoreti dualstack. Se il servizio utilizza IPv6 solo sottoreti, non puoi creare record. A
- 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 si esegue una query DNS sul nome del servizio, i record A e AAAA restituiscono un set di indirizzi IP corrispondenti alle attività. I record SRV restituiscono un set di indirizzi IP e porte per ciascuna 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 come UNHEALTHY fino a quando i controlli dell'integrità del container non restituiscono un valore. Se i controlli dell'integrità risultano positivi, lo stato viene aggiornato a HEALTHY. Se i controlli dell'integrità del container hanno esito negativo, la registrazione dell'istanza di rilevamento servizi 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, consultare [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 il rilevamento servizi.

Maggiori informazioni su come creare un servizio contenente un'attività Fargate che usa il rilevamento servizi con la AWS CLI.

Per un elenco delle scoperte di Regioni AWS tali servizi di supporto, consulta. [Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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, consultare [Installing or updating to the latest version of the AWS CLI](#).
- Le fasi descritte in [Configurazione per l'uso di Amazon ECS](#) sono complete.
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS_ FullAccess](#).
- Hai creato almeno un VPC e un gruppo di sicurezza. Per ulteriori informazioni, consulta [the section called "Crea un cloud privato virtuale"](#).

Passaggio 1: creare le risorse di Service Discovery in AWS Cloud Map

Utilizza la procedura seguente per creare il namespace di individuazione dei servizi e il servizio di individuazione dei servizi.

1. Crea un namespace privato di individuazione dei servizi Cloud Map. In questo esempio viene creato un namespace 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": "h2qe3s6dxftvvt7riu6lfy2f6c3jlfh4-je6chs2e"  
}
```

2. Usando `OperationId` dall'output della fase precedente, verificare che il namespace privato sia stato creato correttamente. Annotare l'ID del namespace perché verrà utilizzato nei comandi successivi.

```
aws servicediscovery get-operation \
  --operation-id h2qe3s6dxftvvt7riu6lfy2f6c3jlfh4-je6chs2e
```

L'output è il seguente.

```
{
  "Operation": {
    "Id": "h2qe3s6dxftvvt7riu6lfy2f6c3jlfh4-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 si crea il servizio ECS, specificare Fargate e la versione della piattaforma LATEST che supporta il rilevamento 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 il namespace 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": "i-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 il namespace, 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 del namespace, ottenere le informazioni sul namespace 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": "xhu73bsertlyffhm3faqi7kumsmx274n-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 del namespace di rilevamento servizi utilizzando l'ID del namespace.

```
aws servicediscovery delete-namespace \
  --id ns-uejictsjen2i4eeg
```

L'output è il seguente.

```
{
  "OperationId": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj"
}
```

- Usando l'OperationId dall'output della fase precedente, verificare che il namespace di rilevamento 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 completamente gestito che consente ai clienti di Amazon ECS di osservare, proteggere e monitorare le applicazioni create su servizi di AWS elaborazione e account VPCs, senza richiedere alcuna modifica al codice.

VPC Lattice utilizza gruppi di destinazione 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 di destinazione di VPC Lattice, i clienti possono ora abilitare le attività Amazon ECS come destinazioni IP in VPC Lattice. Quando vengono avviate le attività per il servizio registrato, Amazon ECS le registra automaticamente nel gruppo di destinazione di VPC Lattice.

Note

Quando si utilizzano cinque configurazioni di VPC Lattice, il tempo di implementazione potrebbe essere leggermente più lungo rispetto a quando si utilizzano meno configurazioni.

Una regola del listener viene utilizzata per inoltrare il traffico a un gruppo di destinazione specifico quando le condizioni sono soddisfatte. Un listener verifica le richieste di connessione utilizzando il protocollo sulla porta configurata. Un servizio instrada le richieste verso i suoi obiettivi registrati in base alle regole definite al momento della configurazione del listener.

Inoltre, Amazon ECS sostituisce automaticamente un'attività anche se diventa non integra in base ai controlli dell'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 di Amazon ECS possono trarre vantaggio da VPC Lattice nei modi seguenti:

- **Maggiore produttività degli sviluppatori:** VPC Lattice aumenta la produttività degli sviluppatori consentendo loro di concentrarsi 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 policy di autenticazione di VPC Lattice. Ciò consente di adottare una postura di sicurezza più solida che soddisfa i principali requisiti normativi e di conformità del settore.

- **Migliore scalabilità e resilienza delle applicazioni:** VPC Lattice consente di creare una rete di applicazioni implementate con funzionalità come autorizzazione, autenticazione, monitoraggio e instradamento 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 la modalità di utilizzo di altri servizi Amazon ECS, mentre altri rimangono invariati.

Application Load Balancer

Non è più necessario creare un Application Load Balancer specifico da utilizzare con il tipo di gruppo di destinazione Application Load Balancer in VPC Lattice che poi si collega al servizio Amazon ECS. È solo necessario configurare il servizio Amazon ECS con un gruppo di destinazione VPC Lattice. È anche possibile scegliere di utilizzare Application Load Balancer con Amazon ECS contemporaneamente.

Implementazioni in sequenza di Amazon ECS

Solo le implementazioni in sequenza di Amazon ECS funzionano con VPC Lattice, e Amazon ECS inserisce le attività in modo sicuro e le rimuove dai servizi durante l'implementazione. La distribuzione e le distribuzioni del codice non sono supportate. Blue/Green

Per saperne di più su VPC Lattice, consultare [Amazon VPC Lattice User Guide](#).

Crea un servizio che utilizza VPC Lattice

È possibile utilizzare Console di gestione AWS 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, consultare [Installing the 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.
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS_FullAccess](#).

Crea un servizio che utilizza VPC Lattice con Console di gestione AWS

Seguire questi passaggi per creare un servizio con VPC Lattice utilizzando Console di gestione AWS.

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, scegliere il cluster in cui creare il servizio.
4. Nella scheda Servizi, scegliere Crea.

Se non è mai stato creato un servizio in precedenza, seguire i passaggi riportati in [Creazione di un servizio Amazon ECS utilizzando la console](#), poi continuare con questi passaggi quando si raggiunge la sezione relativa a VPC Lattice.

5. Scegliere Attiva VPC Lattice selezionando il pulsante.
6. Per utilizzare un ruolo esistente, per Ruolo di infrastruttura ECS per Amazon ECS, sceglierne uno già creato da utilizzare durante la creazione del gruppo di destinazione VPC Lattice. Per creare un nuovo ruolo, Crea ruolo di infrastruttura ECS.
7. Scegliere il VPC.

Il VPC dipende dalla modalità di rete selezionata al momento della registrazione della definizione dell'attività. Se usi la network modalità host o con EC2, scegli il tuo VPC.

Per la modalità `awsvpc`, il VPC viene selezionato automaticamente in base al VPC scelto in Rete e non può essere modificato.

8. In Gruppi di destinazione, scegliere il gruppo o i gruppi di destinazione. È necessario scegliere un gruppo di destinazione compreso tra 1 e 5. Scegliere Aggiungi gruppo di destinazione per aggiungere altri gruppi di destinazione. Scegliere il Nome della porta, il Protocollo e la Porta per ogni gruppo di destinazione scelto. Per rimuovere un gruppo di destinazione, scegliere Rimuovi.

Note

- Se si desidera aggiungere gruppi di destinazione esistenti, è necessario usare la AWS CLI. Per istruzioni su come aggiungere gruppi target utilizzando il AWS CLI, consulta [register-targets](#) nel Reference. AWS Command Line Interface
- Sebbene un servizio VPC Lattice possa essere collegato a più gruppi di destinazione, un gruppo di destinazione può essere aggiunto solo a un singolo servizio.
- Per creare un servizio in una configurazione IPv6 -only, scegli i gruppi di destinazione con un tipo di indirizzo IP di. IPv6

9. A questo punto, accedere alla console VPC Lattice per continuare la configurazione. È qui che si includono i nuovi gruppi di destinazione nell'azione predefinita del listener o nelle regole di un servizio VPC Lattice esistente.

Per ulteriori informazioni, consultare [Listener rules for your VPC Lattice service](#).

Important

È necessario consentire il prefisso `vpc-lattice` della regola in entrata per il gruppo o le attività di sicurezza e i controlli di integrità possono avere esito negativo.

Creare un servizio che utilizza VPC Lattice con la AWS CLI

Usa AWS CLI per creare un servizio con VPC Lattice. Sostituisci ogni *user input placeholder* con le tue informazioni.

1. Creazione di un file di configurazione del gruppo di destinazione. L'esempio seguente è denominato `tg-config.json`

```
{
  "ipAddressType": "IPv4",
  "port": 443,
  "protocol": "HTTPS",
  "protocolVersion": "HTTP1",
  "vpcIdentifier": "vpc-f1663d9868EXAMPLE"
}
```

2. Utilizzare il seguente comando per creare un gruppo di destinazione di VPC Lattice.

```
aws vpc-lattice create-target-group \
  --name my-lattice-target-group-ip \
  --type IP \
  --config file://tg-config.json
```

Note

Per creare un servizio in una configurazione IPv6 solo, crea gruppi target con un tipo di indirizzo IP di. IPv6 Per ulteriori informazioni, consulta [create-target-group](#) nella documentazione di riferimento dei comandi della AWS CLI .

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. Il portName nell'esempio seguente è quello definito nel campo name delle proprietà portMappings della 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"
}

```

Utilizzare il seguente comando per creare un servizio Amazon ECS e collegarlo al gruppo di destinazione 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 in Amazon ECS Anywhere.

Proteggere le attività di Amazon ECS dall'interruzione causata da eventi di riduzione orizzontale

La protezione scalabile di Amazon ECS consente di proteggere le attività dall'interruzione da parte di eventi di riduzione orizzontale derivanti da scalabilità automatica del servizio o implementazioni varie.

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. 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.
- Si dispone di un'applicazione di gioco che esegue i server di gioco come attività di 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`. Quando si imposta `ProtectionEnabled` su `true`, le attività vengono protette per 2 ore per impostazione predefinita. Poi è possibile personalizzare il periodo di protezione utilizzando l'attributo `ExpiresInMinutes`. Puoi proteggere le tue attività per un minimo di 1 minuto e fino a un massimo di 2.880 minuti (48 ore). Se stai utilizzando AWS CLI, puoi specificare l'opzione. `--protection-enabled`

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'opzione `--no-protection-enabled`.

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. Al termine dell'attività, è possibile annullare l'attributo `ProtectionEnabled` utilizzando lo stesso endpoint, rendendo l'attività idonea all'interruzione durante i successivi eventi di scalabilità.

Per ulteriori informazioni sull'endpoint dell'agente del container Amazon ECS, consultare [Endpoint di protezione scalabile di Amazon ECS](#).

- API Amazon ECS

È possibile utilizzare l'API Amazon ECS per impostare e recuperare la protezione scalabile delle attività se l'applicazione dispone di un componente che tiene traccia dello stato delle attività attive. Utilizzare `UpdateTaskProtection` per contrassegnare una o più attività come protette. Utilizzare `GetTaskProtection` per recuperare 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:

- La protezione scalabile è supportata solo con le attività implementate da un servizio.
- La protezione scalabile è supportata con attività implementate da un servizio in esecuzione su istanze gestite da Amazon ECS.
- 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 blue/green aggiornamento, la distribuzione blu con attività protette non verrà rimossa se le attività lo hanno fatto. `protectionEnabled` Il traffico verrà deviato verso le nuove attività che si presentano e le attività più vecchie verranno rimosse solo quando la `protectionEnabled` viene eliminata o scade. 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 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 di 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 container inizia il lavoro di elaborazione, 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.

Usare una richiesta PUT in questo URI dall'interno di un container per impostare la protezione scalabile. Una richiesta GET a questo URI restituisce lo stato di protezione attuale di un'attività.

Parametri di richiesta di protezione scalabile

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`

Specificare `true` per contrassegnare un'attività per la protezione. Specificare `false` per rimuovere la protezione e rendere l'attività idonea per l'interruzione.

Tipo: Booleano

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

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}'
```

Esempi di container Windows

Per i contenitori Windows, è possibile utilizzare PowerShell il `Invoke-RestMethod` cmdlet anziché `curl`. Gli esempi seguenti mostrano gli PowerShell equivalenti dei precedenti comandi `curl`.

Esempio di come proteggere un'attività del container di Windows con il periodo di tempo predefinito

Questo esempio mostra come proteggere un'attività con il periodo di tempo predefinito di 2 ore utilizzando PowerShell

```
Invoke-RestMethod -Uri $env:ECS_AGENT_URI/task-protection/v1/state -Method Put -Body '{"ProtectionEnabled":true}' -ContentType 'application/json'
```

Esempio di come proteggere un'attività del container di Windows per 60 minuti

Questo esempio mostra come proteggere un'attività per 60 minuti utilizzando il `expiresInMinutes` parametro with PowerShell.

```
Invoke-RestMethod -Uri $env:ECS_AGENT_URI/task-protection/v1/state -Method Put -Body  
'{"ProtectionEnabled":true,"ExpiresInMinutes":60}' -ContentType 'application/json'
```

Esempio di come proteggere un'attività del container di Windows per 24 ore

Questo esempio mostra come proteggere un'attività per 24 ore utilizzando il `expiresInMinutes` parametro with PowerShell.

```
Invoke-RestMethod -Uri $env:ECS_AGENT_URI/task-protection/v1/state -Method Put -Body  
'{"ProtectionEnabled":true,"ExpiresInMinutes":1440}' -ContentType 'application/json'
```

La richiesta PUT restituisce la risposta seguente:

```
{  
  "protection": {  
    "ExpirationDate": "2023-12-20T21:57:44.837Z",  
    "ProtectionEnabled": true,  
    "TaskArn": "arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"  
  }  
}
```

Parametri di risposta della protezione scalabile

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
```

Per i contenitori Windows, utilizzate il seguente PowerShell comando per ottenere lo stato di protezione:

```
Invoke-RestMethod -Uri $env:ECS_AGENT_URI/task-protection/v1/state -Method Get
```

```
{
  "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"
  }
}
```

Utilizzare l'iniezione di guasti con i 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 da utilizzare per ottimizzare le prestazioni e la resilienza dell'applicazione.

Quando l'iniezione di guasti è abilitata, l'agente del container Amazon ECS consente alle attività di accedere a nuovi endpoint di iniezione dei guasti. È necessario attivare il consenso per utilizzare l'iniezione di guasti impostando il valore del parametro di definizione dell'attività `enableFaultInjection` su `true`. Il valore predefinito è `false`.

```
{
  ...
  "enableFaultInjection": true
}
```

Note

L'iniezione di guasti funziona solo con le attività che utilizzano le modalità di rete `awsvpc` o `host`.

L'iniezione di guasti non è disponibile in Windows.

Per informazioni su come abilitare l'iniezione di errori in Console di gestione AWS, consulta [Creazione di una definizione di attività Amazon ECS utilizzando la console](#).

Sarà necessario 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`
- Modulo kernel `sch_netem`

Endpoint di iniezione di guasti di 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 l'iniezione di guasti abilitata e ogni endpoint ha un limite di frequenza di 1 richiesta ogni 5 secondi per container. Il superamento di questo limite comporta la generazione di un errore.

Note

L'agente di Amazon ECS version `1.88.0+` è necessario per utilizzare gli endpoint di iniezione di guasti.

I tre endpoint da utilizzare con l'iniezione di guasti sono:

- [Endpoint di porta blackhole di rete](#)
- [Endpoint di perdita di pacchetti di rete](#)
- [Endpoint di latenza di rete](#)

Una richiesta riuscita genera un codice di risposta di `200` con un messaggio di `running` quando si chiama l'endpoint `/start`, `stopped` per l'endpoint `/stop` e `running` o `not-running` per l'endpoint `/status`.

```
{
```



```
"Status": <string>
}
```

Una richiesta non riuscita restituisce uno dei seguenti codici di errore:

- 400: richiesta non valida
- 409: la richiesta di iniezioni di guasti è in conflitto con un altro guasto in esecuzione
- 429: la richiesta è stata sottoposta a limitazione
- 500: il server ha riscontrato un errore imprevisto

```
{
  "Error": <string message>
}
```

Note

È possibile iniettare un guasto 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 di porta blackhole di rete

L'endpoint `{ECS_AGENT_URI}/fault/v1/network-blackhole-port` elimina il traffico in entrata o in uscita per una porta e un protocollo specifici nel namespace di rete di un'attività ed è compatibile con due modalità:

- `awsvpc`: le modifiche vengono applicate al namespace della rete delle attività
- `host`: le modifiche vengono applicate all'istanza di container predefinita del namespace di rete

`{ECS_AGENT_URI}/fault/v1/network-blackhole-port/start`

Questo endpoint avvia le iniezioni di guasti nelle porte blackhole di rete e presenta i seguenti parametri:

Porta

La porta specificata da utilizzare per l'iniezione di guasti nelle porte blackhole.

Tipo: numero intero

Obbligatorio: sì

Protocollo

Il protocollo da utilizzare per l'iniezione di guasti nelle porte blackhole.

Tipo: String

Valori validi: tcp | udp

Obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione di guasti.

Tipo: String

Valori validi: ingress | egress

Obbligatorio: sì

SourcesToFilter

Un array JSON di IPv4 o IPv6 indirizzi o blocchi CIDR protetti dall'errore.

Tipo: array di stringhe

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 arresta il guasto specificato nella richiesta. Questo endpoint dispone dei seguenti parametri:

Porta

La porta interessata dal guasto che deve essere interrotta.

Tipo: numero intero

Obbligatorio: sì

Protocollo

Il protocollo da utilizzare per fermare il guasto.

Tipo: String

Valori validi: tcp | udp

Obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione di guasti.

Tipo: String

Valori validi: ingress | egress

Obbligatorio: sì

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'stopendpoint (sostituisci i valori con i tuoi): *red*

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 controllare lo stato dell'iniezione di guasti. Questo endpoint dispone dei seguenti parametri:

Porta

La porta interessata per verificare lo stato del guasto.

Tipo: numero intero

Obbligatorio: sì

Protocollo

Il protocollo da utilizzare per verificare lo stato del guasto.

Tipo: String

Valori validi: tcp | udp

Obbligatorio: sì

TrafficType

Il tipo di traffico utilizzato dall'iniezione di guasti.

Tipo: String

Valori validi: ingress | egress

Obbligatorio: sì

Di seguito è riportato un esempio di richiesta per l'utilizzo dell'`status` endpoint (sostituisci i valori con i tuoi): *red*

```
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'endpoint `{ECS_AGENT_URI}/fault/v1/network-latency` 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 della rete delle attività
- `host`: le modifiche vengono applicate all'interfaccia della rete predefinita

```
{ECS_AGENT_URI}/fault/v1/network-latency/start
```

Questo endpoint `/start` avvia le iniezioni di 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 di guasti.

Tipo: numero intero

Obbligatorio: sì

`JitterMilliseconds`

Il numero di millisecondi di jitter da aggiungere all'interfaccia di rete da utilizzare per l'iniezione di guasti.

Tipo: numero intero

Obbligatorio: sì

Origini

Un array JSON di IPv4 o IPv6 indirizzi o blocchi CIDR destinati all'uso con l'iniezione di errori.

Tipo: array di stringhe

Obbligatorio: sì

SourcesToFilter

Un array JSON di IPv4 o IPv6 indirizzi o blocchi CIDR protetti dall'errore. `SourcesToFilter` ha la priorità su. `Sources`

Tipo: array di stringhe

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` blocca il guasto e `{ECS_AGENT_URI}/fault/v1/network-latency/status` verifica lo stato del guasto.

Di seguito sono riportati due esempi di richieste per l'utilizzo degli endpoint `/stop` e degli endpoint `/status`. Entrambi utilizzano il metodo POST HTTP.

```
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'endpoint `{ECS_AGENT_URI}/fault/v1/network-packet-loss` aggiunge la perdita di pacchetti all'interfaccia di rete specificata. Questo endpoint è compatibile con due modalità:

- `awsvpc`: le modifiche vengono applicate all'interfaccia della rete delle attività
- `host`: le modifiche vengono applicate all'interfaccia della rete predefinita

`{ECS_AGENT_URI}/fault/v1/network-packet-loss/start`

Questo endpoint `/start` avvia le iniezioni di guasti di perdita di pacchetti di rete e presenta i seguenti parametri:

LossPercent

La percentuale di perdita di pacchetti

Tipo: numero intero

Obbligatorio: sì

Origini

Un array JSON di IPv4 o IPv6 indirizzi o blocchi CIDR da utilizzare per i test di iniezione dei guasti.

Tipo: array di stringhe

Obbligatorio: sì

SourcesToFilter

Un array JSON di IPv6 indirizzi IPv4 o blocchi CIDR protetti dall'errore. `SourcesToFilter` ha la priorità su `Sources`

Tipo: array di stringhe

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'endpoint `{ECS_AGENT_URI}/fault/v1/network-packet-loss/stop` blocca il guasto e `{ECS_AGENT_URI}/fault/v1/network-packet-loss/status` verifica lo stato del guasto. È supportato solo un tipo di guasto alla volta.

Di seguito sono riportati due esempi di richieste per l'utilizzo degli endpoint `/stop` e degli endpoint `/status`. Entrambi utilizzano il metodo POST HTTP.

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-packet-loss/stop
```

```
Endpoint: ${ECS_AGENT_URI}/fault/v1/network-packet-loss/status
```

Aggiornare i parametri del servizio Amazon ECS

Dopo aver creato un servizio, a volte potrebbe essere necessario aggiornare i parametri del servizio, ad esempio il numero di attività.

Quando il pianificatore di servizi avvia nuove attività, determina il posizionamento delle attività nel cluster con la seguente logica.

- Determinare quale delle istanze di container nel cluster può supportare la definizione di attività del servizio. Ad esempio, dispongono gli attributi richiesti di CPU, memoria, porte e istanza di container.
- Per impostazione predefinita, il pianificatore di servizi tenta di bilanciare le attività tra le zone di disponibilità in questo modo, anche se è possibile scegliere una strategia di posizionamento diversa.
 - Ordinare le istanze di container valide in base al numero minore di attività in esecuzione per questo servizio nella stessa zona di disponibilità dell'istanza. 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.

- Posizionare la nuova attività del servizio in un'istanza di container valida in una zona di disponibilità ottimale (in base alle fasi precedenti), favorendo le istanze di container con il minor numero di attività in esecuzione per questo servizio.

Quando il pianificatore di servizi arresta le attività in esecuzione, prova a mantenere il bilanciamento tra le zone di disponibilità nel cluster usando la seguente logica.

- Ordinare le istanze di container in base al numero maggiore di attività in esecuzione per questo servizio nella stessa zona di disponibilità dell'istanza. 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.
- Arrestare l'attività del servizio in un'istanza di container in una zona di disponibilità ottimale (in base alle fasi precedenti), favorendo le istanze di container con il maggior numero di attività in esecuzione per questo servizio.

Utilizzare l'elenco per determinare se è possibile modificare il parametro del servizio.

Ribilanciamento della zona di disponibilità

Indica se utilizzare il ribilanciamento delle zone di disponibilità per il servizio.

È possibile modificare questo parametro per le implementazioni in sequenza.

Strategia del provider di capacità

I dettagli di una strategia di un provider di capacità. È possibile impostare un provider di capacità quando si crea un cluster, si esegue un'attività o si aggiorna un servizio.

Quando si utilizza Fargate, i provider di capacità sono FARGATE o FARGATE_SPOT.

Quando utilizzi Amazon EC2, i fornitori di capacità sono gruppi di Auto Scaling.

È possibile cambiare provider di capacità per le implementazioni in sequenza e le implementazioni blu/verdi.

L'elenco seguente fornisce le transizioni valide:

- Aggiornare Fargate a un provider di capacità di un gruppo Auto Scaling.

- Aggiorna il file EC2 a un provider di capacità Fargate.
- Aggiornare il provider di capacità Fargate a un provider di capacità di un gruppo Auto Scaling.
- Aggiorna il fornitore EC2 di capacità Amazon a un fornitore di capacità Fargate.
- Aggiornare il gruppo Auto Scaling o il provider di capacità Fargate al tipo di avvio. Quando si utilizza la CLI o l'API, si passa un elenco vuoto nel parametro `capacityProviderStrategy`.

Cluster

Non è possibile cambiare il nome del cluster.

Configurazione dell'implementazione

La configurazione di implementazione include gli CloudWatch allarmi e l'interruttore automatico utilizzati per rilevare i guasti e la configurazione richiesta.

L'interruttore di implementazione determina se un'implementazione del servizio avrà esito negativo se il servizio non riesce a raggiungere uno stato stazionario. Se si utilizza l'interruttore automatico di implementazione, un'implementazione del servizio passerà a uno stato non riuscito e interromperà l'avvio di nuove attività. Se si usa l'opzione di rollback, quando l'implementazione di un servizio non riesce, viene eseguito il rollback del servizio per ripristinare l'ultima implementazione completata correttamente.

Quando viene aggiornato un servizio che utilizza l'interruttore automatico Amazon ECS, Amazon ECS crea un'implementazione e una revisione del servizio. Queste risorse consentono di visualizzare informazioni dettagliate sulla cronologia dei servizi. Per ulteriori informazioni, consultare [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi 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 implementazione.

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 quando tutte le istanze del contenitore sono nello DRAINING stato se il servizio contiene attività che utilizzano. 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. Il servizio

considera le attività integre per i servizi che non utilizzano un bilanciatore del carico se sono nello stato `RUNNING`. Il servizio considera le attività integre per i servizi che utilizzano un bilanciatore del carico se sono nello stato `RUNNING` e se sono considerate integre mediante il bilanciatore del carico. 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. EC2 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, 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. Poi, si verificano le seguenti situazioni.

- Se lo stato di integrità dell'attività di sostituzione è `HEALTHY`, il pianificatore di servizi 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](#).

Controller di implementazione

Il tipo di controller di implementazione da utilizzare per il servizio. Esistono tre tipi di controller di implementazione:

- ECS
- EXTERNAL
- CODE_DEPLOY

Quando si aggiorna un servizio, è possibile aggiornare il controller di implementazione utilizzato. L'elenco seguente fornisce le transizioni valide:

- Aggiornamento dalle distribuzioni CodeDeploy blu/verdi () alle distribuzioni in sequenza o alle distribuzioni ECS (CODE_DEPLOY). blue/green ECS
- Aggiornamento da distribuzioni CodeDeploy blu/verdi () a distribuzioni esterne ().
CODE_DEPLOY EXTERNAL
- Aggiornamento da implementazioni sequenziali o blue/green distribuzioni ECS () a distribuzioni esterne (). ECS EXTERNAL
- Aggiornamento da distribuzioni esterne () a implementazioni sequenziali o distribuzioni ECS (EXTERNAL). blue/green ECS

Quando si aggiorna il controller di implementazione di un servizio, tenere presenti le considerazioni seguenti:

- Non è possibile aggiornare il controller di implementazione di un servizio dal controller di implementazione ECS a nessuno degli altri controller se utilizza VPC Lattice o Amazon ECS Service Connect.
- Non è possibile aggiornare il controller di implementazione di un servizio durante un'implementazione del servizio in corso.
- Non è possibile aggiornare il controller di implementazione di un servizio a CODE_DEPLOY se sul servizio non sono presenti bilanciatori del carico.

- Non è possibile aggiornare il controller di implementazione di un servizio da ECS a uno qualsiasi degli altri controller se `deploymentConfiguration` include allarmi, un interruttore automatico di implementazione o una strategia di implementazione `BLUE_GREEN`. Per ulteriori informazioni, consultare [Controller e strategie di implementazione dei servizi Amazon ECS](#).
- Il valore specificato per `versionConsistency` nella definizione del container non verrà utilizzato da Amazon ECS se si aggiorna il controller di implementazione del servizio da ECS a uno qualsiasi degli altri controller.
- Se si aggiorna il controller di implementazione di un servizio da ECS a uno qualsiasi degli altri controller, le risposte `API UpdateService` e `DescribeService` continueranno a restituire `deployments` invece di `taskSets`. Per ulteriori informazioni su `UpdateService` e `CreateService`, consulta [UpdateService](#) e [CreateService](#) nell'Amazon ECS API Reference.
- Se un servizio utilizza una strategia di implementazione con aggiornamento in sequenza, l'aggiornamento del controller di implementazione da ECS a uno qualsiasi degli altri controller modificherà il modo in cui viene utilizzato il valore `maximumPercent` nel `deploymentConfiguration`. Invece di essere utilizzato solo come limite per il totale delle attività in un'implementazione con aggiornamento in sequenza, `maximumPercent` viene utilizzato per sostituire le attività non integre. Per ulteriori informazioni sul modo in cui il pianificatore sostituisce le attività non integre, consultare [Servizi Amazon ECS](#).
- Se si aggiorna il controller di implementazione di un servizio da ECS a uno qualsiasi degli altri controller di implementazione, qualsiasi `advancedConfiguration` specificato con la configurazione del bilanciatore del carico verrà ignorato. Per ulteriori informazioni, consulta [LoadBalancer](#) e [AdvancedConfiguration](#) nel riferimento alle API di Amazon ECS.

Quando aggiorni il controller di distribuzione per un servizio che utilizza CloudFormation, considera quanto segue a seconda del tipo di migrazione che stai eseguendo.

- Se disponi di un CloudFormation modello che contiene le informazioni del controller di `EXTERNAL` distribuzione, nonché `PrimaryTaskSet` le risorse `TaskSet` e rimuovi le risorse del task set dal modello durante l'aggiornamento da `EXTERNAL` a ECS, le chiamate `DescribeTaskSet` e `DeleteTaskSet` API restituiranno un errore 400 dopo l'aggiornamento del controller di distribuzione ECS. Ciò comporta un errore di CloudFormation eliminazione delle risorse del set di attività, anche se lo CloudFormation stack passa allo `UPDATE_COMPLETE` status. Per ulteriori informazioni, consultare [Resource removed from stack but not deleted](#) nella Guida per l'utente di AWS CloudFormation . Per risolvere questo problema, eliminare i set di attività utilizzando direttamente l'API `DeleteTaskSet` di Amazon ECS. Per ulteriori informazioni su come eliminare un set di attività, consulta [DeleteTaskSet](#) Amazon Elastic Container Service API Reference.

- Se stai migrando da CODE_DEPLOY a ECS con una nuova definizione di attività ed CloudFormation esegui un'operazione di rollback, la UpdateService richiama Amazon ECS ha esito negativo con il seguente errore:

```
Resource handler returned message: "Invalid request provided: Unable to update task definition on services with a CODE_DEPLOY deployment controller. Use AWS CodeDeploy to trigger a new deployment. (Service: Ecs, Status Code: 400, Request ID: 0abda1e2-f7b3-4e96-b6e9-c8bc585181ac) (SDK Attempt Count: 1)" (RequestToken: ba8767eb-c99e-efed-6ec8-25011d9473f0, HandlerErrorCode: InvalidRequest)
```

- Dopo una migrazione riuscita dal controller di implementazione da ECS a EXTERNAL, è necessario rimuovere manualmente il set di attività ACTIVE, poiché Amazon ECS non gestisce più l'implementazione. Per informazioni su come eliminare un set di attività, consulta il riferimento [DeleteTaskSet](#) all'API di Amazon Elastic Container Service.

Numero di attività desiderato

Il numero di istanze dell'attività da posizionare e mantenere in esecuzione nel servizio.

Se si desidera interrompere temporaneamente il servizio, impostare questo valore su 0. Quindi, quando si desidera avviare il servizio, aggiornarlo con il valore originale.

È possibile modificare questo parametro per le implementazioni in sequenza e le implementazioni blu/verdi.

Abilitare i tag gestiti

Determina se attivare i tag gestiti da Amazon ECS per le attività all'interno del servizio.

Solo le attività avviate dopo l'aggiornamento rifletteranno quest'ultimo. Per aggiornare i tag di tutte le attività, utilizzare l'opzione di implementazione forzata.

È possibile modificare questo parametro per le implementazioni in sequenza e quelle blu/verdi.

Abilitare ECS Exec

Determina se viene utilizzato Amazon ECS Exec.

Se non si desidera sovrascrivere il valore impostato al momento della creazione del servizio, è possibile impostarlo su null durante l'esecuzione di questa azione.

È possibile modificare questo parametro per le implementazioni in sequenza.

Periodo di tolleranza del controllo dell'integrità

Il periodo di tempo, in secondi, durante il quale il pianificatore di servizi di Amazon ECS ignora i controlli dell'integrità del bilanciamento del carico elastico, VPC Lattice e del container dopo il primo avvio di un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0. Se non si utilizza nessuno dei controlli dell'integrità, allora non viene utilizzato `healthCheckGracePeriodSeconds`.

Se le attività del servizio tardano ad avviarsi e a rispondere ai controlli dello stato, è possibile specificare un periodo di tolleranza per il controllo dell'integrità fino a un massimo di 2.147.483.647 secondi (circa 69 anni), durante i quali il pianificatore del servizio Amazon ECS ignora lo stato di questi controlli. 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.

È possibile modificare questo parametro per le implementazioni in sequenza e le implementazioni blu/verdi.

Sistemi di load balancer

È necessario utilizzare un ruolo collegato al servizio quando si aggiorna un bilanciatore del carico.

Un elenco di oggetti del bilanciatore del carico per il bilanciamento del carico elastico. Contiene il nome del bilanciatore del carico, il nome del container e la porta del container per accedere dal bilanciatore del carico. Il nome del container è come appare in una definizione del container.

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 si aggiunge, si aggiorna o si rimuove una configurazione del bilanciatore del carico, Amazon ECS avvia nuove attività con la configurazione aggiornata del bilanciamento del carico elastico, e poi interrompe le attività vecchie quando le nuove sono in esecuzione.

Per i servizi che utilizzano aggiornamenti continui, è possibile aggiungere, aggiornare o rimuovere i gruppi di destinazione di bilanciamento del carico elastico. È possibile eseguire l'aggiornamento da un singolo gruppo di destinazione a più gruppi di destinazione e da più gruppi di destinazione a un singolo gruppo di destinazione.

Per i servizi che utilizzano blue/green distribuzioni, puoi aggiornare i gruppi target Elastic Load Balancing utilizzando [through. `CreateDeployment`](#) CodeDeploy. Tieni presente che non sono

supportati più gruppi target per le distribuzioni. blue/green Per ulteriori informazioni, consultare [Register multiple target groups with a service](#).

Per i servizi che utilizzano il controller di distribuzione esterno, è possibile aggiungere, aggiornare o rimuovere i sistemi di bilanciamento del carico utilizzando. [CreateTaskSet](#) Per le implementazioni esterne, non sono supportati più gruppi di destinazione. Per ulteriori informazioni, consultare [Register multiple target groups with a service](#).

Passare un elenco vuoto per rimuovere i bilanciatori del carico

È possibile modificare questo parametro per le implementazioni in sequenza.

Configurazione della rete

La configurazione di rete del servizio.

È possibile modificare questo parametro per le implementazioni in sequenza.

Vincoli di posizionamento

Una serie di oggetti di vincolo di posizionamento delle attività per aggiornare il servizio da utilizzare. Se non viene specificato alcun valore, i vincoli di posizionamento esistenti per il servizio rimarranno invariati. Se specificato, questo valore sovrascriverà tutti i vincoli di posizionamento esistenti definiti per il servizio. Per rimuovere tutti i vincoli di posizionamento esistenti, specificare una serie vuota.

Puoi specificare un massimo di 10 vincoli per ogni attività. Questo limite include i vincoli nella definizione di attività e quelli specificati in fase di runtime.

È possibile modificare questo parametro per le implementazioni in sequenza e quelle blu/verdi.

Strategia di posizionamento

Gli oggetti di strategia di posizionamento delle attività per aggiornare il servizio da utilizzare. Se non viene specificato alcun valore, la strategia di posizionamento esistente per il servizio rimarrà invariata. Se specificato, questo valore sovrascriverà la strategia di posizionamento esistente definita per il servizio. Per rimuovere una strategia di posizionamento esistente, specificare un oggetto vuoto.

È possibile modificare questo parametro per le implementazioni in sequenza e quelle blu/verdi.

Versione della piattaforma

La versione della piattaforma Fargate su cui viene eseguito il servizio.

Un servizio che utilizza una versione della piattaforma Linux non può essere aggiornato per utilizzare una versione della piattaforma Windows e viceversa.

È possibile modificare questo parametro per le implementazioni in sequenza.

Propaga i tag

Determina se propagare i tag dalla definizione dell'attività o dal servizio all'attività. Se non viene specificato alcun valore, i tag non vengono propagati.

Solo le attività avviate dopo l'aggiornamento rifletteranno quest'ultimo. Per aggiornare i tag su tutte le attività, impostare `forceNewDeployment` su `true`, in modo tale che Amazon ECS avvii nuove attività con i tag aggiornati.

È possibile modificare questo parametro per le implementazioni in sequenza e le implementazioni blu/verdi.

Configurazione di Service Connect

La configurazione per Amazon ECS Service Connect. Questo parametro determina il modo in cui il servizio si connette ad altri servizi all'interno dell'applicazione.

È possibile modificare questo parametro per le implementazioni in sequenza.

Registri del servizio

È necessario utilizzare un ruolo collegato ai servizi quando si aggiornano i registri di servizio.

I dettagli per i registri di rilevamento servizi da assegnare a questo servizio. Per ulteriori informazioni, consultare [Service Discovery](#).

Quando si aggiunge, si aggiorna o si rimuove la configurazione dei registri del servizio, Amazon ECS avvia nuove attività con la configurazione aggiornata dei registri del servizio, e poi interrompe le attività vecchie quando le nuove attività sono in esecuzione.

Passare un elenco vuoto per rimuovere i registri di servizio.

È possibile modificare questo parametro per le implementazioni in sequenza.

Definizione di attività

La definizione di attività e la revisione da utilizzare per il servizio.

Se si modificano 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 e la porta del container specificati nella configurazione del bilanciatore del carico devono rimanere nella definizione di attività.

Il comportamento di estrazione delle immagini del container è diverso a seconda delle opzioni di calcolo. Per ulteriori informazioni, consultare uno dei seguenti argomenti:

- [Progettazione per AWS Fargate per Amazon ECS](#)
- [Architetto per la EC2 capacità di Amazon ECS](#)
- [Esterno \(Amazon ECS Anywhere\) per Amazon ECS](#)

È possibile modificare questo parametro per le implementazioni in sequenza.

Configurazione del volume

I dettagli del volume che era `configuredAtLaunch`. Quando `configuredAtLaunch` è impostato su `true` nella definizione dell'attività, questo parametro di servizio configura un volume Amazon EBS per ogni attività nel servizio da creare e allegare durante l'implementazione. [È possibile configurare le dimensioni, il tipo di volume, l'IOPS, la velocità effettiva, l'istantanea e la crittografia in Configuration. ServiceManaged EBSVolume](#) Il name del volume deve corrispondere al name dalla definizione dell'attività. Se impostato su `null`, non viene attivata alcuna nuova implementazione. In caso contrario, se questa configurazione è diversa da quella esistente, attiva una nuova implementazione.

È possibile modificare questo parametro per le implementazioni in sequenza.

Configurazione di VPC Lattice

La configurazione di VPC Lattice per il servizio. Questo definisce il modo in cui il servizio si integra con VPC service-to-service Lattice per la comunicazione.

È possibile modificare questo parametro per le implementazioni in sequenza.

AWS CDK considerazioni

AWS CDK Non tiene traccia degli stati delle risorse. Non sa se si sta creando o aggiornando un servizio. I clienti devono utilizzare l'escape hatch per accedere direttamente al costruito L1 Service di ECS.

Per informazioni sugli escape hatch, consulta [Personalizza i costrutti dalla AWS Construct Library](#) nella Guida per sviluppatori AWS Cloud Development Kit (AWS CDK) v2.

Per eseguire la migrazione del servizio esistente al costrutto `ecs.Service`, effettuare le seguenti operazioni:

1. Utilizzare l'escape hatch per accedere al costrutto L1 del Service.
2. Impostare manualmente le seguenti proprietà nel costrutto L1 del Service.

Se il tuo servizio utilizza la EC2 capacità di Amazon:

- `daemon?`
- `placementConstraints?`
- `placementStrategies?`
- Se si utilizza la modalità di rete `awsvpc`, è necessario impostare le `vpcSubnets?` e i costrutti `securityGroups?`.

Se il servizio utilizza Fargate:

- `FargatePlatformVersion`
- Le `vpcSubnets?` e i costrutti `securityGroups?`.

3. Impostare `launchType` nel modo seguente:

```
const cfnEcsService = service.node.findChild('Service') as ecs.CfnService;  
cfnEcsService.launchType = "FARGATE";
```

Per eseguire la migrazione da un tipo di avvio a un provider di capacità, procedere come segue:

1. Utilizzare l'escape hatch per accedere al costrutto L1 del Service.
2. Aggiungere il costrutto `capacityProviderStrategies?`.
3. Distribuire il servizio.

Aggiornamento di un servizio Amazon ECS

Dopo aver creato un servizio, a volte potrebbe essere necessario aggiornare i parametri del servizio, ad esempio il numero di attività.

Quando viene aggiornato un servizio che utilizza l'interruttore automatico Amazon ECS, Amazon ECS crea un'implementazione e una revisione del servizio. Queste risorse consentono di visualizzare

informazioni dettagliate sulla cronologia dei servizi. Per ulteriori informazioni, consultare [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).

Prerequisiti

Prima di aggiornare un servizio, verificare quali parametri del servizio possono essere modificati in base al tipo di implementazione. Per un elenco completo dei parametri modificabili, consultare [Aggiornare i parametri del servizio Amazon ECS](#).

Procedura

Console

1. [Apri la console nella versione 2](https://console.aws.amazon.com/ecs/)<https://console.aws.amazon.com/ecs/>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare la casella di spunta accanto al servizio e poi scegliere 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 di definizioni di processi e la revisione selezionate siano compatibili con la configurazione di calcolo definita. Se viene visualizzato un avviso, verificare che siano selezionate sia la compatibilità delle definizioni di processo che la configurazione di calcolo.

6. Se hai scelto Replica, per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e mantenere nel servizio.
7. Se è stato 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 Ribilanciamento del servizio delle zone di disponibilità, selezionare Ribilanciamento del servizio della zona di disponibilità.
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, consultare [Deployment configuration](#).

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 le attività vengono implementate per il servizio, espandere le Opzioni di implementazione e quindi configurare le opzioni.
 - a. Per Tipo di controller di implementazione, specificare il controller di implementazione del servizio. La console Amazon ECS supporta i seguenti tipi di controller: ECS.
 - b. Per Strategia di implementazione, scegliere la strategia utilizzata da Amazon ECS per implementare nuove versioni del servizio.
 - c. A seconda della scelta della Strategia di implementazione, eseguire le seguenti operazioni:

Strategia di implementazione	Fasi	
Aggiornamento continuo	a. Per Percentuale minima di attività in esecuzione, specificare un valore percentuale minimo di attività che devono essere eseguite durante l'implementazione di un servizio. Per ulteriori informazioni, consultare Distribui i servizi Amazon ECS sostituendo le attività. b. Per Percentuale massima di attività in esecuzione, specificare un valore percentua	

Strategia di implementazione	Fasi	
	le massimo di attività che possono essere eseguite durante l'implementazione di un servizio. Per ulteriori informazioni, consultare e Distribuisci i servizi Amazon ECS sostituendo le attività.	
Blu/verde	Per Tempo di incorporamento, specificare la durata, in minuti, durante la quale le revisioni del servizio blu e verde devono essere eseguite contemporaneamente. Per ulteriori informazioni, consultare Implementazioni Amazon ECS blue/green.	

- d. Per eseguire le funzioni Lambda per una fase del ciclo di vita, in Hook del ciclo di vita di implementazione, svolgere le seguenti operazioni per ogni funzione Lambda unica:

- i. Scegliere Aggiungi.

Ripetere l'operazione per ogni funzione unica che si desidera eseguire.

- ii. Per Funzione Lambda, immettere il nome della funzione.
- iii. Per Ruolo, scegli il ruolo che hai creato nei prerequisiti con le blue/green autorizzazioni.

Per ulteriori informazioni, consulta [Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#).

- iv. Per Fasi del ciclo di vita, selezionare le fasi eseguite dalla funzione Lambda.

- v. (Facoltativo) Per Dettagli dell'hook, inserire una coppia chiave-valore che fornisca informazioni sull'hook.
11. 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 l'implementazione all'ultimo stato di implementazione completata quando l'interruttore automatico di implementazione imposta l'implementazione su uno stato di errore, selezionare Rollback in caso di errore.

- 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.

12. Per modificare le opzioni di calcolo, espandere Configurazione di calcolo, quindi effettuare le seguenti operazioni:
- a. Per i servizi attivi AWS Fargate, per la versione della piattaforma, scegli la nuova versione.
 - b. Per i servizi che utilizzano una strategia del provider di capacità, in Strategia del provider di capacità, eseguire le operazioni seguenti:
 - 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 un gruppo Auto Scaling non può essere aggiornato per utilizzare un provider di capacità Fargate. Un servizio che utilizza un provider di capacità Fargate che non può essere aggiornato per utilizzare un provider di capacità di un gruppo Auto Scaling.

13. (Facoltativo) Per configurare il servizio di dimensionamento automatico, espandere Dimensionamento automatico del servizio e quindi specificare i seguenti parametri. Per utilizzare il dimensionamento automatico predittivo, che esamina i dati di caricamento precedenti provenienti dai flussi di traffico, è necessario configurarlo dopo aver creato il servizio. Per ulteriori informazioni, consultare [Usa modelli storici per scalare i servizi Amazon ECS con scalabilità predittiva](#).
 - a. Per utilizzare il dimensionamento automatico, selezionare Dimensionamento automatico del servizio.
 - b. Per Numero minimo di attività, inserire il limite inferiore del numero di attività che devono essere utilizzate dal servizio di dimensionamento automatico. Il numero desiderato non scenderà al di sotto di questo conteggio.
 - c. In Numero massimo di processi, specificare il limite superiore del numero di processi che devono essere utilizzati dal servizio di dimensionamento automatico. Il numero desiderato non sarà superiore a questo conteggio.
 - d. Scegli il tipo di policy. In Tipo di policy di dimensionamento, scegliere una delle opzioni seguenti.

Per utilizzare questo tipo di policy	Esegui questa operazione	
Monitoraggio degli obiettivi	a. In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. b. In Nome policy, inserire il nome della policy. c. Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. • ECSServiceMediaCPUUtilization: utilizzo medio della CPU del servizio. • ECSServiceAverageMemoryUtilization— Utilizzo medio della memoria del servizio. • ALBRequestCountPerTarget— Numero di richieste completate per destinazione in un gruppo target di Application Load Balancer. d. Per Target value (Valore di destinazione), inserisci il valore mantenuto dal servizio	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	per il parametro selezionato. e. Per Tempo di raffreddamento di aumento orizzontale, immettere la quantità di tempo che deve passare, in secondi, tra un'attività di aumento (aggiungere attività) e l'inizio di un'altra attività di questo tipo. f. Per Tempo di raffreddamento di riduzione orizzontale, immettere la quantità di tempo che deve passare, in secondi, tra un'attività di riduzione (rimuovere attività) e l'inizio di un'altra attività di questo tipo. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Selezionare Disattiva riduzione orizzontale se si desidera che la politica di scalabilità	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	venga ridimensionata in base all'aumento del traffico, ma non è necessario che venga ampliata quando il traffico diminuisce.	

Per utilizzare questo tipo di policy	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.	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	i. In Operazioni di dimensionamento, esegui queste operazioni: • In Operazioni di dimensionamento, scegliere se aggiungere o rimuovere o impostare un determinato numero desiderato per il servizio. • Se si sceglie di aggiungere o rimuovere attività, immettere nel campo Valore il numero di attività (o la percentuale di attività esistenti) da aggiungere o rimuovere quando l'operazione di dimensionamento viene attivata. 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 percentua	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	le del conteggio desiderato esistente. • Per Limite inferiore e Limite superiore, inserisci i relativi limiti della regolazione di dimensionamento per fasi. Per impostazione 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. Scegliere Aggiungi nuova operazione e di dimensionamento, quindi ripetere i passaggi di Operazioni di dimensionamento. • In Tempo di raffreddamento, inserire	

Per utilizzare questo tipo di policy	Esegui questa operazione	
	la quantità di tempo, espressa in secondi, necessari a per rendere effettiva un'attività di dimensionamento precedente. Per una policy di aggiunta, si tratta del periodo successivo a un'attività di aumento orizzontale in cui la politica di scalabilità blocca le attività con aumento orizzontale e limita il numero di attività che possono essere aumentate orizzontalmente alla volta. Per una policy di rimozione, questo è il tempo che deve passare tra un'attività di riduzione e l'inizio di un'altra attività di questo tipo.	

14. (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 nel namespace, scegliere 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 nel namespace, scegliere Client e server.
- b. Per utilizzare un namespace differente da quello del cluster predefinito, per Namespace, scegliere il namespace del servizio. Può trattarsi di uno spazio dei nomi creato separatamente Regione AWS nello stesso spazio dell'utente Account AWS o di uno spazio dei nomi nella stessa regione condiviso con l'account utilizzando (). AWS Resource Access Manager AWS RAM [Per ulteriori informazioni sugli spazi dei AWS Cloud Map nomi condivisi, consulta Condivisione dello spazio dei nomi tra account nella Guida per gli sviluppatori AWS Cloud Map AWS Cloud Map](#)
- c. (Facoltativo) Specificare una configurazione del log. Selezionare Usa la raccolta di log. L'opzione predefinita invia i log dei contenitori a Logs. CloudWatch 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 AWS 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 del container a Firehose. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
 - Flusso di dati Amazon Kinesis: configura il processo per inviare log di container a Kinesis Data Streams. Vengono fornite le opzioni di driver di log predefinite che inviano i log a un flusso di consegna 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 log di container a un bucket Amazon S3. Vengono fornite le opzioni del driver di log predefinito, ma è necessario specificare un nome del bucket Amazon S3 valido.
- d. Per abilitare i log di accesso, segui questi passaggi:

- i. Espandi la configurazione del registro di accesso. Per Format, scegli JSON oTEXT.
- ii. Per includere i parametri di interrogazione nei log di accesso, selezionate Includi parametri di interrogazione.

 Note

Per disabilitare i log di accesso, in Formato, scegli Nessuno.

15. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento dell'implementazione, è possibile configurare il volume espandendo Volume.

Il nome e il tipo di volume vengono configurati durante la creazione di una revisione della definizione di 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 Tipo di volume EBS, scegliere il tipo di volume EBS da collegare all'attività. b. Per Dimensione (GiB), inserire 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 snapshot.	

Per configurare questo tipo di volume	Esegui questa operazione	
	c. Per IOPS, inserite il numero massimo di input/output operazioni (IOPS) che il volume deve fornire. Questo valore è configurabile solo per i tipi di volume <code>io1</code>, <code>io2</code> e <code>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 ID Snapshot, scegliere uno snapshot del volume Amazon EBS esistente o inserire l'ARN di uno snapshot se si desidera creare un volume da uno snapshot. È possibile creare un nuovo volume vuoto anche non scegliendo o inserendo un ID di snapshot. f. Se si specifica un ID di snapshot, è possibile specificare una Velocità di inizializzazione del volume (MiB/s). Inserire	

Per configurare questo tipo di volume	Esegui questa operazione	
	un valore compreso tra 100 e 300 in MiB/s, che determinerà la velocità di caricamento dei dati dallo snapshot specificato utilizzando ID Snapshot per la creazione del volume. g. Per Tipo di file system, scegliere 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 una snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione della snapshot. Se c'è una mancata corrispondenza del tipo di file system, l'attività non verrà avviata. h. Per il Ruolo di infrastruttura, scegliere un ruolo IAM con le autorizza	

Per configurare questo tipo di volume	Esegui questa operazione	
	zioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. È possibile allegare la policy gestita da AmazonECS InfrastructureRolePolicyForVolumes al ruolo, oppure utilizzare la policy come guida per creare e allegare la policy con autorizzazioni che soddisfino le esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Ruolo IAM dell'infrastruttura Amazon ECS. i. Per Crittografia, scegliere Predefinita se si desidera utilizzare le impostazioni della crittografia Amazon EBS predefinita. Se sul tuo account è configurata la crittografia per impostazione predefinita, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specifica	

Per configurare questo tipo di volume	Esegui questa operazione	
	ta nell'impostazione. Se si sceglie Predefinita 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 si sceglie Nessuna, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se si crea un volume da uno snapshot crittografato. j. Se hai scelto Personalizzato per la crittografia, devi specificare AWS KMS key quello che desideri utilizzare. Per la Chiave KMS, scegliere una AWS KMS key o inserire un ARN della chiave. Se scegli di crittografare il volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di	

Per configurare questo tipo di volume	Esegui questa operazione	
	disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazioni, consultare Data encryption for Amazon EBS volumes. k. (Facoltativo) In Tag, è possibile aggiungere tag al volume di Amazon EBS propagando i tag dalla definizione dell'attività o del servizio, o fornendo tag personalizzati. Se si desidera propagare i tag dalla definizione dell'attività, scegliere Definizione dell'attività per Propaga tag da. Se si desidera propagare i tag dal servizio, scegliere Servizio per Propagare i tag da. Se si sceglie Non propagare, o se non si sceglie un valore, i tag non vengono propagati. Se si desidera fornire i propri tag, scegliere Aggiungi tag, e poi fornire la chiave e il	

Per configurare questo tipo di volume	Esegui questa operazione	
	valore per ogni tag aggiunto. Per ulteriori informazioni sulla menzione di volumi Amazon EBS, consultar e Tagging Amazon EBS volumes.	

16. (Facoltativo) Per identificare il tuo servizio, espandi la sezione Tags (Tag), quindi configura i tag.

- [Aggiungere un tag] Scegliere Aggiungi tag e procedere come segue:
 - 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).

17. Scegliere Aggiorna.

AWS CLI

- Esegui `update-service`. Per informazioni sull'esecuzione del comando, consulta [update-service](#) nella Guida di riferimento. AWS Command Line Interface

L'esempio `update-service` seguente aggiorna il numero di attività desiderato del servizio `my-http-service` su 2.

Sostituisci il *`user-input`* con i tuoi valori.

```
aws ecs update-service \
  --cluster MyCluster \
  --service my-http-service \
  --desired-count 2
```

Fasi successive

Tracciare l'implementazione e visualizzare la cronologia dei servizi dell'interruttore automatico di Amazon ECS. Per ulteriori informazioni, consultare [Visualizza la cronologia dei servizi utilizzando le distribuzioni dei servizi Amazon ECS](#).

Aggiornamento di un servizio Amazon ECS per utilizzare un provider di capacità

Se disponi di un servizio esistente che utilizza il tipo di avvio Amazon EC2 o Fargate e desideri utilizzare Amazon ECS Managed Instances, devi aggiornare il servizio per utilizzare il tuo provider di capacità Amazon ECS Managed Instances.

Prerequisiti

Creare un provider di capacità per le istanze gestite da Amazon ECS. Per ulteriori informazioni, consultare [Creazione di un provider di capacità per le istanze gestite da Amazon ECS](#).

Procedura

Console

1. <https://console.aws.amazon.com/ecs/Api> la console nella versione 2.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare la casella di spunta accanto al servizio e poi scegliere Aggiorna.
4. Selezionare Forza una nuova implementazione.
5. In Configurazione di calcolo, scegliere Strategia del provider di capacità. Poi, scegliere una delle opzioni seguenti:
 - Se il provider di capacità delle istanze gestite da Amazon ECS è il provider di capacità predefinito, scegliere Usa impostazione predefinita del cluster.
 - Se il provider di capacità delle istanze gestite da Amazon ECS non è il provider di capacità predefinito, scegliere Usa personalizzato (avanzate). Scegliere il provider di capacità delle istanze gestite da Amazon ECS, quindi per Peso, scegliere 1.
6. Scegliere Aggiorna.

AWS CLI

- Esegui `update-service`. Per informazioni sull'esecuzione del comando, consulta [update-service](#) nella Guida di riferimento. AWS Command Line Interface

Sostituisci il *user-input* con i tuoi valori.

```
aws ecs update-service \
  --cluster my-cluster \
  --service my-service \
  --capacity-provider-strategy capacityProvider=my-managed-instance-capacity-provider,weight=1 \
  --force-new-deployment
```

Eliminazione di un servizio Amazon ECS utilizzando la console

Di seguito sono riportati alcuni dei motivi per cui eliminare un servizio:

- L'applicazione non è più necessaria
- Si sta eseguendo la migrazione del servizio in un nuovo ambiente
- L'applicazione non viene utilizzata attivamente
- L'applicazione utilizza più risorse del necessario e si sta 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 si elimina un servizio, Amazon ECS elimina tutte le implementazioni e le revisioni del servizio.

Quando si elimina un servizio, se vi sono ancora attività in esecuzione che richiedono pulizia, lo stato del servizio passa da ACTIVE a DRAINING e il servizio non è più visibile nella console o nell'operazione 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 stato DRAINING o INACTIVE possono ancora essere visualizzati con l'operazione API `DescribeServices`.

 Important

Se si tenta di creare un nuovo servizio con lo stesso nome di un servizio esistente nello stato ACTIVE o DRAINING, si riceverà un errore.

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 Cluster, scegliere 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. Al prompt di conferma, inserire elimina, quindi scegliere Elimina.

Eeguire la migrazione di un ARN di servizio breve di Amazon ECS in un ARN lungo

Amazon ECS assegna un nome della risorsa Amazon (ARN) univoco a ciascun servizio. I servizi creati prima del 2021 hanno un formato ARN corto:

```
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 assegnare tag al servizio.

È possibile migrare un servizio con un formato ARN corto al formato ARN lungo senza dover ricreare il servizio. È possibile usare l'API, la CLI o la console. L'operazione di migrazione non può essere annullata.

Il processo di migrazione è semplice e garantisce zero tempi di inattività per il servizio. Durante la migrazione:

- **Disponibilità del servizio:** il servizio continua a funzionare normalmente senza interruzioni del traffico o delle funzionalità.
- **Attività in esecuzione:** le attività esistenti continuano a essere eseguite senza interruzioni. Le nuove attività avviate dopo la migrazione utilizzeranno il formato ARN lungo se l'impostazione dell'account `taskLongArnFormat` è abilitata.
- **Istanze di container:** le istanze di container non sono interessate dalla migrazione ARN del servizio e continuano a funzionare normalmente.
- **Configurazione del servizio:** tutte le impostazioni del servizio, tra cui la definizione delle attività, la rete e le configurazioni del bilanciatore del carico, rimangono invariate.

Se desideri utilizzare per 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 CloudFormation per etichettare il servizio.

Se si desidera utilizzare Terraform per etichettare un servizio con un formato ARN corto, è necessario migrare il servizio utilizzando l'API, la CLI o la console. Una volta completata la migrazione, è possibile utilizzare Terraform per etichettare il servizio.

Al termine della 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 quota di tag.

- Quando `PhysicalResourceId` in uno 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 si dispone di un ARN di servizio corto, visualizzare i dettagli del servizio nella console Amazon ECS (appare un avviso quando il servizio ha il formato ARN corto) o il

parametro `return` di `describe-services`. Quando l'ARN non include il nome del cluster, si dispone di un ARN corto. Il formato di un ARN corto è il seguente:

```
arn:aws:ecs:region:aws_account_id:service/service-name
```

2. Si noti la data di creazione.
3. Se si dispone di policy IAM che utilizzano il formato ARN corto, aggiornarlo 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 si dispone di strumenti che utilizzano il formato ARN corto, aggiornarlo 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. Abilitare 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.

Eseguire il comando come utente root quando il servizio ha una data di `createdAt` 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. Abilitare il formato ARN lungo dell'attività. Questa impostazione dell'account controlla il formato ARN per le nuove attività che vengono avviate dopo il completamento della migrazione del

servizio. 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.

Eseguire il comando come utente root quando il servizio ha una data di `createdAt` 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"
  }
}
```

Note

L'impostazione `taskLongArnFormat` non migra direttamente le attività esistenti. Influisce solo sul formato ARN delle nuove attività create dopo l'attivazione dell'impostazione. Le attività in esecuzione esistenti mantengono il formato dell'ARN attuale fino a quando non vengono sostituite tramite le normali operazioni di servizio (come implementazioni o attività di scalabilità).

Procedura

Usare quanto segue per eseguire la migrazione dell'ARN del servizio.

Console

1. Apri la console nella <https://console.aws.amazon.com/ecs/versione 2>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella sezione Servizi, scegliere un servizio con un avviso nella colonna ARN.

Si apre la pagina dei dettagli del servizio.

4. Scegliere Migra a un ARN lungo.

Appare la finestra di dialogo Migra servizio.

5. Scegliere Migrate (Migrazione).

CLI

Dopo aver soddisfatto tutti i prerequisiti, è possibile etichettare il servizio. Esegui il comando seguente:

Amazon ECS valuta la possibilità di passare il formato ARN lungo in una richiesta API `tag-resource` per un servizio con un ARN corto come segnale per eseguire la migrazione del 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 "TestService" e un valore impostato su ": 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 soddisfatto tutti i prerequisiti, è possibile etichettare il servizio. Creare una risorsa `aws_ecs_service` e impostare il riferimento tags. Per ulteriori informazioni, consultare [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"  
  }  
}
```

```
}
```

Fasi successive

È possibile aggiungere tag al servizio. Per ulteriori informazioni, consultare [Aggiunta di tag alle risorse Amazon ECS](#).

Se si desidera che Amazon ECS propaghi i tag dalla definizione dell'attività o dal servizio all'attività, eseguire `update-service` con il parametro `propagateTags`. Per ulteriori informazioni, vedere [update-service nel Reference](#). AWS Command Line Interface

Risoluzione dei problemi

Alcuni utenti potrebbero riscontrare il seguente errore durante la migrazione dal formato ARN corto 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 l'impostazione dell'account `serviceLongArnFormat` è già stata abilitata ma questo errore continua a verificarsi, 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. Identificare 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 la 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. Abilitare le impostazioni dell'account richieste per quel principale specifico. Questa operazione può essere eseguita in uno dei seguenti modi:
 - a. Assumere il ruolo o l'utente IAM per quel principale. Quindi eseguire `put-account-setting`.

- b. Usare 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 dell'account `serviceLongArnFormat` richiesta sul principale che ha creato il servizio, il che consente di procedere con la migrazione dell'ARN.

Logica di limitazione del servizio Amazon ECS

Il pianificatore di servizi Amazon ECS include la logica protettiva che limita i lanci delle attività quando queste non riescono ripetutamente a essere avviate. Questo aiuta a prevenire il consumo inutile di risorse e riduce i costi.

Quando le attività di un servizio non riescono a passare da uno stato PENDING a RUNNING e invece passano direttamente a STOPPED, il pianificatore del servizio:

- Aumenta in modo incrementale il tempo tra i tentativi di riavvio
- Continua ad aumentare i ritardi fino a un massimo di 27 minuti tra un tentativo e l'altro
- Genera un messaggio relativo all'evento del servizio per avvisare l'utente del problema

Note

Il periodo di ritardo massimo di 27 minuti potrebbe cambiare nei futuri aggiornamenti.

Quando la limitazione è attivata, si riceve questo messaggio relativo all'evento di servizio:

```
(service service-name) is unable to consistently start tasks successfully.
```

Caratteristiche importanti della logica di limitazione:

- I servizi continuano a ripetere i tentativi a tempo indeterminato

- L'unica modifica è l'aumento del tempo tra i riavvii
- Non ci sono parametri configurabili dall'utente

Risoluzione dei problemi di limitazione della larghezza di banda della rete

Per risolvere il problema della limitazione, è possibile:

- Aggiornare il servizio per usare una nuova definizione di attività, che riporta immediatamente il servizio a un funzionamento normale e non limitato. Per ulteriori informazioni, consultare [Aggiornamento di un servizio Amazon ECS](#).
- Affrontare la causa alla base degli errori delle attività.

Le cause più comuni degli errori delle attività che innescano la limitazione includono:

- Risorse del cluster (porte, memoria o CPU) insufficienti
 - Indicato da un [messaggio di evento relativo al servizio di risorse insufficiente](#)
- Errori di estrazione dell'immagine del container
 - Possono essere causati da nomi di immagini non validi, tag o autorizzazioni insufficienti
 - Determina `CannotPullContainerError` in [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#)
- Spazio su disco insufficiente
 - Determina `CannotCreateContainerError` in [errori di attività interrotte](#)
 - Per le fasi di risoluzione, consultare [Risolvi i problemi relativi a API error \(500\): devmapper di Docker in Amazon ECS](#)

Important

I seguenti scenari NON attivano la logica di limitazione:

- Attività che si interrompono dopo aver raggiunto lo stato `RUNNING`
- Attività interrotte a causa di controlli dell'integrità non riusciti del bilanciamento del carico elastico
- Attività in cui il comando del container termina con un codice diverso da zero dopo aver raggiunto lo stato `RUNNING`

Parametri di definizione del servizio di Amazon ECS

Una definizione di servizio stabilisce come eseguire il servizio Amazon ECS. I seguenti parametri possono essere specificati in una definizione di servizio.

Tipo di lancio

`launchType`

Tipo: String

Valori validi: EC2 | FARGATE | EXTERNAL

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

Strategia del fornitore di capacità

`capacityProviderStrategy`

Tipo: matrice di oggetti

Obbligatorio: no

La strategia del fornitore di capacità da utilizzare per il servizio.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del 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

Obbligatorio: sì

Il nome breve o il nome della risorsa Amazon (ARN) del provider di capacità.

`weight`

Tipo: numero intero

Intervallo valido: numeri interi compresi tra 0 e 1.000.

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: numero intero

Intervallo valido: numeri interi compresi tra 0 e 100.000.

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: stringa

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Una definizione di attività deve essere specificata quando si utilizza il controller di implementazione dell'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: stringa

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

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: stringa

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 del servizio

serviceName

Tipo: stringa

Obbligatorio: sì

Il nome del servizio. Sono consentite fino a 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: String

Valori validi: REPLICA | DAEMON

Obbligatorio: no

La strategia di pianificazione da utilizzare. Se non viene specificata alcuna strategia di pianificazione, viene utilizzata la strategia REPLICA. Per ulteriori informazioni, consultare [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, consultare [Strategia di pianificazione delle repliche](#).

- **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, consultare [Strategia di pianificazione dei daemon](#).

**Note**

I processi Fargate non supportano la strategia di pianificazione DAEMON

Conteggio desiderato

`desiredCount`

Tipo: Integer

Obbligatorio: no

Il numero di istanze della definizione di attività specificata da posizionare e mantenere in esecuzione nel servizio.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del 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 dimensionamento automatico dei servizi, quando si aggiorna un servizio attualmente in esecuzione con un `desiredCount` inferiore al numero di attività attualmente in esecuzione, il servizio viene ridotto verticalmente fino al `desiredCount` specificato.

Configurazione dell'implementazione

`deploymentConfiguration`

Tipo: oggetto

Obbligatorio: no

I parametri di implementazione opzionali determinano quante attività vengono eseguite durante l'implementazione e l'ordine di arresto e di avvio delle attività.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

`maximumPercent`

Tipo: Integer

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 il pianificatore di servizi REPLICA e ha un `desiredCount` di quattro attività e un valore `maximumPercent` pari al 200%, il pianificatore avvia quattro nuove attività prima di arrestare le quattro attività più vecchie, 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%.

Il pianificatore 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 sul modo in cui il pianificatore sostituisce le attività non integre, consultare [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 blue/green (CODE_DEPLOY) o di EXTERNAL distribuzione e le attività che utilizzano il tipo di EC2 avvio, il valore percentuale massimo viene impostato sul valore predefinito. Il valore viene utilizzato solo per definire il limite superiore sul numero di

attività nel servizio che rimangono nello stato RUNNING mentre le istanze di container sono nello stato DRAINING.

 Note

Non è possibile specificare un `maximumPercent` valore personalizzato per un servizio che utilizza il tipo blue/green (`CODE_DEPLOY`) o di `EXTERNAL` distribuzione e presenta attività che utilizzano il. EC2

Se il servizio utilizza i tipi di implementazione blu/verde (`CODE_DEPLOY`) o `EXTERNAL`, e le attività del servizio utilizzano Fargate, il valore percentuale massimo non viene utilizzato. Il valore viene comunque restituito quando si descrive il servizio.

`minimumHealthyPercent`

Tipo: Integer

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 un `desiredCount` di quattro attività, un `minimumHealthyPercent` del 50% e un `maximumPercent` del 100%, il pianificatore di servizi arresta due attività esistenti per liberare capacità del cluster prima di avviare 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 sul modo in cui il pianificatore sostituisce le attività non integre, consultare [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, consultare [Controllo dell'integrità](#).

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 e 50% per. APIs Console di gestione AWS

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 di `EXTERNAL` distribuzione e sta eseguendo attività che utilizzano il EC2, il valore della percentuale di integrità minima viene impostato sul valore predefinito. Il valore viene utilizzato solo per definire il limite inferiore sul numero di attività nel servizio che rimangono nello stato RUNNING mentre le istanze di container sono nello stato DRAINING.

 Note

Non è possibile specificare un `maximumPercent` valore personalizzato per un servizio che utilizza il tipo `blue/green` (`CODE_DEPLOY`) o di `EXTERNAL` distribuzione e presenta attività che utilizzano il. EC2

Se un servizio utilizza i tipi di implementazione blu/verde (`CODE_DEPLOY`) o `EXTERNAL` ed esegue processi che utilizzano 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

Obbligatorio: no

Il tipo di controller di implementazione da utilizzare per il servizio. Se non viene specificato alcun controller di implementazione, viene utilizzato il controller ECS. Per ulteriori informazioni, consultare [Servizi Amazon ECS](#).

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

`type`

Tipo: String

Valori validi: `ECS` | `CODE_DEPLOY` | `EXTERNAL`

Obbligatorio: sì

Il tipo di controller di implementazione da utilizzare. Esistono tre tipi di controller di implementazione:

`ECS`

Il controller di distribuzione Amazon ECS supporta diverse strategie di implementazione: aggiornamento progressivo, `blue/green`, `linear`, and `canary`. The rolling update deployment

type involves replacing the current running version of the container with the latest version. Blue/green implementazioni che creano un nuovo ambiente e spostano il traffico contemporaneamente. Le implementazioni lineari spostano gradualmente il traffico con incrementi percentuali uguali. Le implementazioni Canary spostano prima una piccola percentuale del traffico, quindi spostano il traffico rimanente. 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 implementazione blue/green (CODE_DEPLOY) utilizza il modello di blue/green implementazione fornito da 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

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 si utilizza Fargate, i vincoli di posizionamento dei processi non sono supportati.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

type

─Tipo: stringa

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: stringa

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, consultare [Creare espressioni per definire istanze di container per le attività di Amazon ECS](#).

`placementStrategy`

Tipo: matrice di oggetti

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.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

`type`

Tipo: String

Valori validi: `random` | `spread` | `binpack`

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: stringa

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

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, consultare [Aggiunta di tag alle risorse Amazon ECS](#).

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

key

Tipo: String

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

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

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

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`

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, consultare [Uso dei tag per la risoluzione dei problemi](#).

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

`propagateTags`

Tipo: String

Valori validi: `TASK_DEFINITION` | `SERVICE`

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`.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

Configurazione della rete

`networkConfiguration`

Tipo: oggetto

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. Se si utilizza Fargate, la modalità di rete `awsvpc` è obbligatoria. Per ulteriori informazioni sulla rete per EC2, vedere [Opzioni di task networking di Amazon ECS per EC2](#). Per maggiori informazioni sulla rete per Fargate, consultare [Amazon ECS task networking options for Fargate](#).

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

`awsvpcConfiguration`

Tipo: oggetto

Obbligatorio: no

Un oggetto che rappresenta le sottoreti e i gruppi di sicurezza per un'attività o un servizio.

`subnets`

Tipo: array di stringhe

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: array di stringhe

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: String

Valori validi: ENABLED | DISABLED

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

Obbligatorio: no

Il periodo di tempo, in secondi, durante il quale il pianificatore di servizi di Amazon ECS ignora i controlli dell'integrità del bilanciamento del carico elastico, VPC Lattice e del container dopo il primo avvio di un'attività. Se non si specifica un valore per il periodo di tolleranza per il controllo dell'integrità, viene utilizzato il valore predefinito di 0. Se non si utilizza nessuno dei controlli dell'integrità, allora non viene utilizzato healthCheckGracePeriodSeconds.

Se le attività del servizio tardano ad avviarsi e a rispondere, puoi specificare un periodo di tolleranza per il controllo dell'integrità fino a un massimo di 2.147.483.647 secondi (circa 69 anni), durante i quali il pianificatore del servizio Amazon ECS ignora lo stato di questi controlli. 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.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

loadBalancers

Tipo: matrice di oggetti

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla Console di gestione AWS. È possibile utilizzare AWS Copilot AWS CLI o SDK per modificare

la configurazione del load balancer solo per il ECS Rolling Deployment Controller, 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.

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: stringa

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 di destinazione viene specificato solo quando di utilizza un Application Load Balancer o un Network Load Balancer.

`loadBalancerName`

─Tipo: stringa

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 omissso.

`containerName`

─Tipo: stringa

Obbligatorio: no

Il nome del container (così come appare nella definizione di quest'ultimo) da associare al load balancer.

`containerPort`

Tipo: Integer

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 EC2, l'istanza del contenitore deve consentire il traffico in entrata sulla mappatura `hostPort` delle porte.

`role`

─Tipo: stringa

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`.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

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, consultare [Friendly Names and Paths](#) 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, consultare [Uso di ruoli collegati ai servizi per Amazon ECS](#).

serviceConnectConfiguration

Tipo: oggetto

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Per ulteriori informazioni, consultare [Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi](#).

enabled

Tipo: Booleano

Obbligatorio: sì

Specifica se utilizzare Service Connect con questo servizio.

namespace

─Tipo: stringa

Obbligatorio: no

Il nome breve o il nome completo Amazon Resource Name (ARN) dello spazio dei AWS Cloud Map nomi da utilizzare con Service Connect. Il namespace deve trovarsi nella stessa Regione AWS del cluster e del servizio Amazon ECS. Il tipo di namespace non influisce su Service Connect. Per ulteriori informazioni AWS Cloud Map, consulta [Working with Services](#) nella Developer Guide.AWS Cloud Map

services

Tipo: matrice di oggetti

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 un namespace utilizza solo per connettersi ad altri servizi nel namespace. Un esempio può essere un'applicazione frontend che accetta richieste in arrivo da un bilanciatore 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

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: stringa

Obbligatorio: no

`discoveryName` È il nome del nuovo AWS Cloud Map servizio creato da Amazon ECS per questo servizio Amazon ECS. Deve essere univoco nel namespace AWS Cloud Map .

Se questo campo non è specificato, viene utilizzato `portName`.

`clientAliases`

Tipo: matrice di oggetti

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 nel namespace.

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: numero intero

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 namespace.

Per evitare di modificare le applicazioni nei servizi client Amazon ECS, imposta la stessa porta utilizzata dall'applicazione client per impostazione predefinita.

`dnsName`

■Tipo: stringa

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

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](#)

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 `r`.

`accessLogConfiguration`

Tipo: [ServiceConnectAccessLogConfiguration](#) oggetto

Obbligatorio: no

La configurazione per la registrazione degli accessi di Service Connect, incluso il formato del registro e se i registri devono includere stringhe di query. I registri di accesso raccolgono informazioni dettagliate sulle richieste effettuate al servizio, inclusi i modelli di richiesta, il codice di risposta e i dati temporali. Per abilitare i registri di accesso, è inoltre necessario specificare un `inlogConfiguration`. `serviceConnectConfiguration`

`serviceRegistries`

Tipo: matrice di oggetti

Obbligatorio: no

I dettagli della configurazione dell'individuazione per il tuo servizio. Per ulteriori informazioni, consultare [Usare il rilevamento servizi per connettere i servizi Amazon ECS con nomi DNS](#).

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

`registryArn`

─Tipo: stringa

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, consultare [Working with Services](#) nella Guida per gli sviluppatori di AWS Cloud Map .

`port`

Tipo: Integer

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: stringa

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

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: stringa

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: stringa

Obbligatorio: no

Indica se il servizio utilizza il ribilanciamento della zona di disponibilità. I valori validi sono `ENABLED` e `DISABLED`.

Quando si aggiorna un servizio, questo parametro non innesca una nuova implementazione del servizio.

Comportamento predefinito:

- Per i nuovi servizi: se non viene specificato alcun valore, il valore predefinito è `DISABLED`.
- Per i servizi esistenti: se non viene specificato alcun valore, Amazon ECS imposta il valore predefinito sul valore esistente. Se in precedenza non è stato impostato alcun valore, Amazon ECS lo imposta su `DISABLED`.

Per ulteriori informazioni sul ribilanciamento delle zone di disponibilità, consultare [Bilanciamento di un servizio Amazon ECS tra zone di disponibilità](#).

Configurazioni del volume

volumeConfigurations

Tipo: oggetto

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.

Quando si aggiorna un servizio, questo parametro innesca una nuova implementazione del servizio.

Questo oggetto è necessario per collegare i volumi Amazon EBS alle attività gestite da un servizio. Per ulteriori informazioni, consultare [Usare i volumi Amazon EBS con Amazon ECS](#).

name

Tipo: stringa

Obbligatorio: sì

Il nome di un volume configurato durante la creazione o l'aggiornamento di un servizio. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini (`_`) e caratteri di sottolineatura (`-`). Questo valore deve corrispondere al nome di volume specificato nella definizione dell'attività.

managedEBSVolume

Tipo: oggetto

Obbligatorio: no

La configurazione del volume utilizzata per creare volumi Amazon EBS collegati alle attività mantenute da un servizio al momento della creazione o dell'aggiornamento del servizio. Viene allegato un volume per ogni attività.

encrypted

Tipo: Booleano

Obbligatorio: no

Valori validi: `true|false`

Specifica se crittografare ciascun 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, consultare [Enable Amazon EBS encryption by default](#) nella Guida per l'utente di Amazon EBS. Per ulteriori informazioni sulla crittografia dei volumi Amazon EBS collegati alle attività di Amazon ECS, consultare [Crittografia dei dati archiviati nei volumi Amazon EBS allegati alle attività Amazon ECS](#).

`kmsKeyId`

■Tipo: stringa

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 dell'archiviazione gestita di Amazon ECS che potrebbe essere stata specificata. Per ulteriori informazioni, consultare [Crittografia dei dati archiviati nei volumi Amazon EBS allegati alle attività Amazon ECS](#).

È possibile specificare la chiave KMS utilizzando una delle opzioni seguenti:

- ID della chiave: ad esempio, `1234abcd-12ab-34cd-56ef-1234567890ab`.
- Alias della chiave: ad esempio, `alias/ExampleAlias`.
- ARN della chiave: ad esempio, `arn:aws:kms:us-east-1:012345678910:key/1234abcd-12ab-34cd-56ef-1234567890ab`.
- ARN dell'alias: 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, alias o ARN non valido, l'azione sembra essere completata, ma alla fine ha

esito negativo. Per ulteriori informazioni, consultare [Troubleshooting Amazon EBS volume attachment issues](#).

volumeType

─Tipo: stringa

Obbligatorio: no

Valori validi: | | | | gp2 gp3 io1 io2 sc1 st1 standard

Il tipo di volume di Amazon EBS. Per ulteriori informazioni sui tipi di volume, consultare [Amazon EBS volume types](#) nella Guida per l'utente di Amazon EBS. Il tipo di volume di default è gp3.

Note

Il tipo di volume standard non è supportato per le attività Fargate.

sizeInGiB

Tipo: Integer

Obbligatorio: no

Intervallo valido: numeri interi compresi tra 1 e 16.384.

La dimensione del volume EBS in gibibyte (GiB). Se non viene fornito un ID di snapshot per configurare un volume per l'allegato, è necessario fornire un valore di dimensione. Se viene configurato un volume per l'allegato utilizzando uno snapshot, il valore predefinito è la dimensione dello snapshot. È possibile specificare una dimensione maggiore o uguale a quella dello snapshot.

Per i tipi di volume gp2 e gp3, l'intervallo valido è 1-16.384.

Per i tipi di volume io1 e io2, l'intervallo valido è 4-16.384.

Per i tipi di volume st1 e sc1, l'intervallo valido è 125-16.384.

Per il tipo di volume standard, l'intervallo valido è 1-1.024.

snapshotId

■Tipo: stringa

Obbligatorio: no

L'ID di snapshot di un volume Amazon EBS esistente che Amazon ECS utilizza per creare nuovi volumi per il collegamento. È necessario specificare `snapshotId` o `sizeInGiB`.

volumeInitializationRate

Tipo: Integer

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 per il collegamento. Questa proprietà può essere specificata solo se si indica un `snapshotId`. Per ulteriori informazioni su questa velocità di inizializzazione dei volumi, compreso l'intervallo di velocità supportate, consultare [Initialize Amazon EBS volumes](#) nella Guida per l'utente di Amazon EBS.

iops

Tipo: Integer

Obbligatorio: no

Il numero di I/O operazioni 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 la performance di base del volume e la velocità con cui il volume accumula I/O crediti per il bursting. Questo parametro è obbligatorio per i volumi `io1` e `io2`. Questo parametro non è supportato per volumi `gp2`, `st1`, `sc1` o `standard`.

Per i volumi `gp3`, l'intervallo di valori valido è compreso tra 3.000 e 16.000.

Per i volumi `io1`, l'intervallo di valori valido è compreso tra 100 e 64.000.

Per i volumi `io2`, l'intervallo di valori valido è compreso tra 100 e 64.000.

throughput

Tipo: Integer

Obbligatorio: no

Il throughput per il provisioning dei volumi collegati alle attività mantenute da un servizio.

 Important

Questo parametro è supportato solo per i volumi gp3.

roleArn

Tipo: stringa

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

Obbligatorio: no

La specifica per i tag da applicare a ciascun volume di Amazon EBS.

resourceType

Tipo: stringa

Obbligatorio: sì

Valori validi: volume

Il tipo di risorsa cui applicare tag al momento della creazione.

tags

Tipo: matrice di oggetti

Obbligatorio: no

I metadati applicati ai volumi per aiutare a categorizzarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili.

AmazonECSCreated e AmazonECSManaged sono tag riservati che vengono aggiunti

da Amazon ECS, quindi è possibile specificare un massimo di 48 tag personalizzati. Quando un volume viene eliminato, vengono eliminati anche i tag. Per ulteriori informazioni, consultare [Aggiunta di tag alle risorse Amazon ECS](#).

key

Tipo: String

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

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

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

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).

propagateTags

Tipo: String

Valori validi: TASK_DEFINITION | SERVICE | NONE

Obbligatorio: no

Specificare se copiare i tag dalla definizione di attività o dal servizio in un volume. Se NONE è specificato o non viene specificato alcun valore, i tag non vengono copiati.

fileSystemType

─Tipo: stringa

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 una snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione della snapshot. Se c'è una mancata corrispondenza del tipo di file system, l'attività non verrà avviata.

I valori validi per Linux sono `xfs`, `ext3`, 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.

EC2

```
{
  "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": ""
    }
  ],
  "enableECSManagedTags": 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": ""
      }
    }
  ]
}

```

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
```

Aggiunta di tag alle 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 relativi 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 elimini una risorsa, verranno eliminati anche tutti i tag a essa associati.

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' Console di gestione AWS 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 EC2 container Amazon per 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 .	Sì, solo dai fornitori di capacità per le istanze gestite da Amazon ECS. Puoi propagare i tag dal provider di capacità per Amazon ECS Managed Instances a tutte le risorse gestite dal provider, come EC2 istanze Amazon, modello di lancio, interfacce di rete elastiche e volumi.	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
- Fornitori di capacità Amazon ECS

Amazon ECS offre la possibilità di utilizzare l'autorizzazione all'assegnazione di tag per la creazione di risorse. Quando l' Account AWS è configurato per l'autorizzazione dei tag, gli utenti devono

disporre delle autorizzazioni per le azioni che creano la risorsa, come `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 ulteriori 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 il tuo schema di tagging, 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 i tag gestiti da Amazon ECS, il servizio assegna automaticamente a tutte le attività appena avviate e a tutti i volumi Amazon EBS associati alle attività i tag relativi alle 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 sulla menzione di volumi Amazon EBS, consultare [Utilizzo dei tag per i 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 associato a un'attività che fa parte di un servizio riceverà un tag con una Chiave come `aws:ecs:clusterName` e un Valore impostato sul nome del cluster, nonché un tag con una Chiave come `aws:ecs:serviceName` e un Valore impostato sul nome del servizio. Per ulteriori informazioni sulla menzione di volumi Amazon EBS, consultare [Utilizzo dei tag per i 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 si utilizza il APIs per creare un servizio o eseguire un'attività, è necessario `enableECSTags` impostare su `true` for `run-task` e `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. Consigliamo di non modificare manualmente i tag in modo che Amazon ECS possa gestire efficacemente le funzionalità.

Uso dei tag per la risoluzione dei problemi

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.

Aggiunta di tag alle risorse Amazon ECS

Puoi applicare tag ad attività, servizi, definizioni delle attività o cluster nuovi o esistenti. Per informazioni sull'etichettatura delle istanze dei container, consulta. [Aggiungere tag a un'istanza di EC2 container Amazon per 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.

Operazione	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'implementazione di aggiornamenti continui di Amazon ECS	create-service	CreateService
Crea un set di attività.	Implementare 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 i carichi di lavoro Fargate	create-cluster	CreateCluster
Esegui una o più istanze di container.	Avvio di un'istanza di container Linux di Amazon ECS	run-instances	RunInstances

Operazione	Console	AWS CLI	Operazione API
Crea un provider di capacità per le istanze gestite da Amazon ECS.	Creazione di un provider di capacità per le istanze gestite da Amazon ECS	create-capacity-provider	CreateCapacityProvider

Assegnare tag alle risorse esistenti (console Amazon ECS)

Puoi aggiungere o eliminare i tag associati a cluster, servizi, attività e definizioni di attività direttamente dalla pagina della risorsa.

Modifica di un tag per una singola risorsa

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
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).

Assegnazione di tag a risorse AWS CLI esistenti

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](#)

Tag per le istanze gestite da Amazon ECS

Le istanze gestite di Amazon ECS supportano una combinazione di tag personalizzati e tag aggiunti AWS che possono essere utilizzati per l'ottimizzazione dei costi. Per ulteriori informazioni sull'utilizzo dei tag per la risoluzione dei problemi, consulta [Uso dei tag per la risoluzione dei problemi](#).

Tag aggiunti da AWS

AWS aggiunge i seguenti tag a ogni istanza gestita di Amazon ECS creata dal provider di capacità:

- Amazon ECS assegna automaticamente i tag riservati AmazonECSCreated e AmazonECSManaged alle istanze gestite da Amazon ECS.
- Amazon ECS assegna i seguenti tag di sistema a ogni istanza:
 - 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:capacityProviderName` e un Valore impostato sul nome del provider di capacità.
 - Un tag con una Chiave come `aws:ecs:containerInstanceId` e un Valore come ID dell'istanza di container per l'istanza gestita da Amazon ECS.
- Amazon EC2 aggiunge il tag di sistema `aws:ec2:managed-launch` con il valore `ecs-managed-instances`. Inoltre, Amazon EC2 aggiunge tag di sistema che indicano il modello di avvio utilizzato per creare l'istanza gestita e la EC2 flotta Amazon di cui l'istanza gestita fa parte.

Tag personalizzati

Puoi assegnare tag personalizzati aggiuntivi alle istanze gestite da Amazon ECS assegnando tag al provider di capacità e abilitando la propagazione dei tag usando la proprietà `propagateTags`.

La seguente definizione di provider di capacità mostra come è possibile specificare e propagare i tag dal provider di capacità quando si crea il provider di capacità utilizzando il valore `CAPACITY_PROVIDER` per `propagateTags`.

```
{
  "name": "my-cluster-managed-instances-cp",
  "cluster": "my-cluster",
  "tags": [
    {
      "key": "tag_key",
      "value": "tag_value"
    }
  ],
  "managedInstancesProvider": {
    "infrastructureRoleArn": "arn:aws:iam::123456789012:role/ecsInfrastructureRole",
    "propagateTags": "CAPACITY_PROVIDER",
    "instanceLaunchTemplate": {
      "ec2InstanceProfileArn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceProfile",
      "networkConfiguration": {
        "subnets": [
          "subnet-abcdef01234567",
          "subnet-bcdefa98765432"
        ],
        "securityGroups": [
          "sg-0123456789abcdef"
        ]
      }
    }
  }
}
```

 Note

Quando si assegnano nuovi tag a un provider di capacità, i tag appena assegnati non verranno propagati alle istanze esistenti ma verranno propagati a tutte le istanze appena create.

Per ulteriori informazioni sui provider di capacità delle istanze gestite da Amazon ECS, consulta [Provider di capacità di Istanze gestite da Amazon ECS](#).

Aggiungere tag a un'istanza di EC2 container Amazon per Amazon ECS

Puoi associare tag alle tue istanze di EC2 container Amazon per Amazon ECS utilizzando uno dei seguenti metodi:

- Metodo 1: quando crei l'istanza del contenitore utilizzando l' EC2 API, 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' EC2API di Amazon. Per ulteriori informazioni, consulta [Bootstrap di istanze di container Linux Amazon ECS per il trasferimento dei 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeTags"
      ],
      "Resource": "*"
    }
  ]
}
```

Aggiunta di tag alle istanze di container esterne per Amazon ECS

Puoi associare i tag alle istanze di container esterne per Amazon ECS 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 utilizzare il per aggiungere tag. Console di gestione AWS Per ulteriori informazioni, consulta [Assegnare 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. Il modo in cui tali dati vengono presentati dipende dall'opzione di elaborazione dell'attività.

Per le attività che utilizzano Fargate, la colonna `lineItem/Operation` mostrerà `FargateTask` e verrà visualizzato il costo associato a ciascuna attività.

Per le attività che utilizzano EC2, la `lineItem/Operation` colonna mostra `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 [Uso dei tag per la risoluzione dei problemi](#).

 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 del servizio
- Tag delle risorse
- Tipo di lancio
- 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'splitLineItem intestazione del rapporto. 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.

Di seguito sono indicati i prerequisiti:

- Imposta il parametro di configurazione dell'agente Amazon ECS `ECS_DISABLE_METRICS` su `false`.

Quando questa impostazione è impostata `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 il monitoraggio di Amazon ECS, è opportuno creare 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 dall'opzione di elaborazione dei processi e dei servizi nei cluster. Se utilizzi Fargate per i tuoi servizi, vengono forniti i parametri di utilizzo della memoria e della CPU per semplificare il monitoraggio dei servizi. Infatti EC2, possiedi e devi monitorare le EC2 istanze che costituiscono l'infrastruttura sottostante. Parametri di prenotazione e di utilizzo della CPU e della memoria vengono resi disponibili a livello di cluster, servizio e processo.

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 il monitoraggio di Amazon ECS.

- Rendi il monitoraggio una priorità per risolvere i piccoli problemi prima che diventino grandi problemi
- 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?
- Automatizza il monitoraggio il più possibile.
- Controlla i file di log di Amazon ECS. Per ulteriori informazioni, consulta [Visualizzare i log dell'agente 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 il Monitoraggio del runtime](#).

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 Fargate, è possibile 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 [Scalabilità automatica del servizio Amazon ECS](#).

Per i cluster che utilizzano attività o servizi EC2, 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 di 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 per accedere ai log può essere utilizzato dai contenitori. 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.
- Approfondimenti sui container: raccogli, aggrega e riepiloga parametri e log dalle applicazioni e dai microservizi containerizzati. Container Insights raccoglie dati come eventi di log delle prestazioni tramite Embedded Metric Format. Questi eventi di log delle prestazioni sono elementi che usano uno schema JSON strutturato che permette ai dati ad alta cardinalità di essere acquisiti e archiviati

su larga scala. Da questi dati, CloudWatch crea metriche aggregate a livello di cluster, attività e servizio come CloudWatch metriche. 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, [Registrazione delle chiamate API Amazon ECS tramite 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 i cluster e i container all'interno dell'ambiente AWS . 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 EC2
 - Parametri del servizio
 - Stato di integrità dei servizi
 - Eventi di implementazione dei servizi
- 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 dell'integrità del container: si tratta di comandi che vengono eseguiti localmente su un container e convalidano lo stato e la disponibilità delle applicazioni. Devi configurarli per container nella definizione dell'attività.
- AWS Trusted Advisor può aiutarti a monitorare AWS le tue 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 ai guasti che indica che un servizio è in esecuzione in un'unica zona di disponibilità.
- Una tolleranza ai guasti che indica che non è stata utilizzata la strategia di collocazione sparsa per più zone di disponibilità.
- AWS Compute Optimizer è un servizio che analizza i parametri 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](#).

- Server Amazon ECS MCP: utilizza gli assistenti AI per monitorare le distribuzioni, risolvere i problemi dei container e ispezionare le configurazioni utilizzando il linguaggio naturale. Il server MCP offre visibilità in tempo reale sullo stato del servizio, analizza gli errori delle attività e accelera la risoluzione dei problemi. Per ulteriori informazioni, consulta [Server MCP 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 container di 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 container e sulla policy 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. Puoi misurare la prenotazione di CPU e memoria, così come l'utilizzo di CPU, memoria e file system

EBS, nell'intero cluster e l'utilizzo della CPU, della memoria e del file system EBS sui servizi nei cluster. Per i carichi di lavoro GPU, puoi misurare la prenotazione GPU nel cluster.

L'infrastruttura su cui sono ospitate le attività Amazon ECS nei tuoi cluster determina quali parametri sono disponibili. Per le attività ospitate sull'infrastruttura Fargate, Amazon ECS fornisce parametri di utilizzo di CPU, memoria e file system EBS per semplificare il monitoraggio dei 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 gli approfondimenti sui container](#)

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 [Parametri 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 i parametri da visualizzare.

Per visualizzare i parametri da	Fasi	
Cluster	Nella pagina sui dettagli del cluster, scegli la scheda Parametri. È inoltre disponibile 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. I parametri sono disponibili nella scheda Stato e parametri.	

CloudWatch console

Per 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 i fornitori di EC2 capacità, i parametri dei cluster e dei servizi di 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.

Amazon ECS invia i parametri a CloudWatch intervalli di un minuto. Questi parametri vengono raccolti per le risorse che svolgono attività nello stato RUNNING. Se un cluster, un servizio o un'altra risorsa non hanno attività in esecuzione, non verrà riportato alcun parametro per quella risorsa durante quel periodo. Ad esempio, se hai un cluster con un servizio ma quel servizio non ha attività in uno RUNNING stato, non verranno inviate metriche a CloudWatch. Allo stesso modo, se disponi di due servizi e uno di essi ha processi in esecuzione mentre l'altro no, verranno inviati solo i parametri per il servizio con i processi in esecuzione.

Metrica	Description	Dimensione valida	Statistiche utili	Unità
CPUReservation	La percentuale di unità CPU che viene riservata nel cluster o nel servizio.	ClusterName	Media, Minimo, Massimo	Percentuale
	La prenotazione della CPU (filtrata per ClusterName) 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.			

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	e 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 ospitate su un' EC2 istanza Amazon.			

Metrica	Description	Dimensione valida	Statistiche utili	Unità
CPUUtilization	La percentuale di unità CPU che viene 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 EC2 le 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, Minimo, Massimo	Percentuale

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	attività ospitate su un'istanza Amazon. EC2 L'utilizzo della CPU a livello di servizio (filtrato solo per <code>ClusterName</code> e <code>ServiceName</code>) viene misurato come il totale delle unità CPU in uso dai processi che appartengono al servizio, diviso il numero totale delle unità CPU prenotate per tutti i processi che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su EC2 istanze Amazon e Fargate.			

Metrica	Description	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 nel cluster. Questo parametro può essere filtrato solo per <code>ClusterName</code>. Solo EC2 le istanze Amazon in <code>ACTIVE</code> o in <code>DRAINING</code> stato influiranno sui parametri di prenotazione della memoria.	<code>ClusterName</code>	Media, Minimo, Massimo	Percentuale

Metrica	Description	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 specificato, MemoryReservation viene utilizzato nel calcolo al posto della memoria totale.			

Metrica	Description	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, Minimo, Massimo	Percentuale

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	su un'istanza Amazon. EC2 L'utilizzo della memoria a livello di servizio (filtrato per <code>ClusterName</code> e <code>ServiceName</code>) viene misurato come il totale della memoria in uso dai processi che appartengono al servizio, diviso il totale della memoria prenotata per tutti i processi che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su EC2 istanze Amazon e Fargate.			

Metrica	Description	Dimensione valida	Statistiche utili	Unità
EBSFilesystemUtilization	La percentuale del file system Amazon EBS utilizzata dalle attività in un servizio. Il parametro di utilizzo del file system EBS a livello di servizio (filtrato per <code>ClusterName</code> e <code>ServiceName</code>) viene misurato come il totale del file system EBS in uso nei processi che appartengono al servizio, diviso il totale dell'archiviazione del file system EBS allocato per tutti i processi che appartengono al servizio. La metrica di utilizzo del file system EBS a livello di servizio è disponibile solo per le	<code>ClusterName</code> , <code>ServiceName</code>	Media, Minimo, Massimo	Percentuale

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	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 lo spazio sul disco che viene utilizzato solo da Fargate. Non ci sono costi associati allo			

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	spazio utilizzat o da Fargate, ma vedrai questo spazio di archiviaz ione aggiuntiv o utilizzan do strumenti come df.			

Metrica	Description	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, Minimo, Massimo	Percentuale

Metrica	Description	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 DiscoveryName selezionato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect. Dimensioni valide: DiscoveryName e DiscoveryName, ServiceName, ClusterName .	DiscoveryName e, DiscoveryName ServiceName ClusterName	Media, Minimo, Massimo, Somma	Conteggio

Metrica	Description	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 Discovery Name selezionato. Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.	Discovery Name e Discovery Name, ServiceName, ClusterName	Media, Minimo, Massimo, Somma	Conteggio

Metrica	Description	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

Metrica	Description	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 nella mappatura delle porte della definizione dell'attività.	DiscoveryName e DiscoveryName, ServiceName, ClusterName	Media, Minimo, Massimo, Somma	Conteggio

Metrica	Description	Dimensione valida	Statistiche utili	Unità
GrpcRequestCount	Il numero di richieste di traffico in entrata gRPC elaborato 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

Metrica	Description	Dimensione valida	Statistiche utili	Unità
HTTPCode_Target_2XX_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

Metrica	Description	Dimensione valida	Statistiche utili	Unità
	Dimensioni valide:			

Metrica	Description	Dimensione valida	Statistiche utili	Unità
HTTPCode_Target_3XX_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

Metrica	Description	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

Metrica	Description	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

Metrica	Description	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

Metrica	Description	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, Minimo, Massimo	Millisecondi

Metrica	Description	Dimensione valida	Statistiche utili	Unità
ClientTLSNegotiationErrorCount	Numero totale di volte in cui la connessione TLS non è riuscita. Questo parametro viene utilizzato 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

Metrica	Description	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. Questo parametro viene utilizzato 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 il namespace 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
<code>ClusterName</code>	Questa dimensione filtra i dati richiesti per tutte le risorse in un cluster specificato. Tutti i parametri di Amazon ECS vengono filtrati per <code>ClusterName</code> .
<code>ServiceName</code>	Questa dimensione filtra i dati richiesti per tutte le risorse in un servizio specificato all'interno di un cluster specificato.
<code>DiscoveryName</code>	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.
<code>DiscoveryName, ServiceName, ClusterName</code>	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.

Dimensione	Definizione
TargetDiscoveryName	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. 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 namespace. 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 <code>DiscoveryName</code>, <code>ServiceName</code>, <code>ClusterName</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 namespace. 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 avvertono quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sulle quote di servizio di Fargate, consulta [Endpoint e quote di Amazon ECS](#) nel Riferimenti generali di Amazon Web Services.

AWS Fargate pubblica le seguenti metriche nel namespace. `AWS/Usage`

Metrica	Description
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	Description
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 User Guide.Quote di servizio

Parametri di prenotazione dei cluster di 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 supportato su cluster con processi ospitati su AWS Fargate.

$$\text{Cluster CPU reservation} = \frac{(\text{Total CPU units reserved by tasks in cluster}) \times 100}{(\text{Total CPU units registered by container instances in cluster})}$$

$$\text{Cluster memory reservation} = \frac{(\text{Total MiB of memory reserved by tasks in cluster} \times 100)}{(\text{Total MiB of memory registered by container instances in cluster})}$$

$$\text{Cluster GPU reservation} = \frac{(\text{Total GPUs reserved by tasks in cluster} \times 100)}{(\text{Total GPUs registered by container instances in cluster})}$$

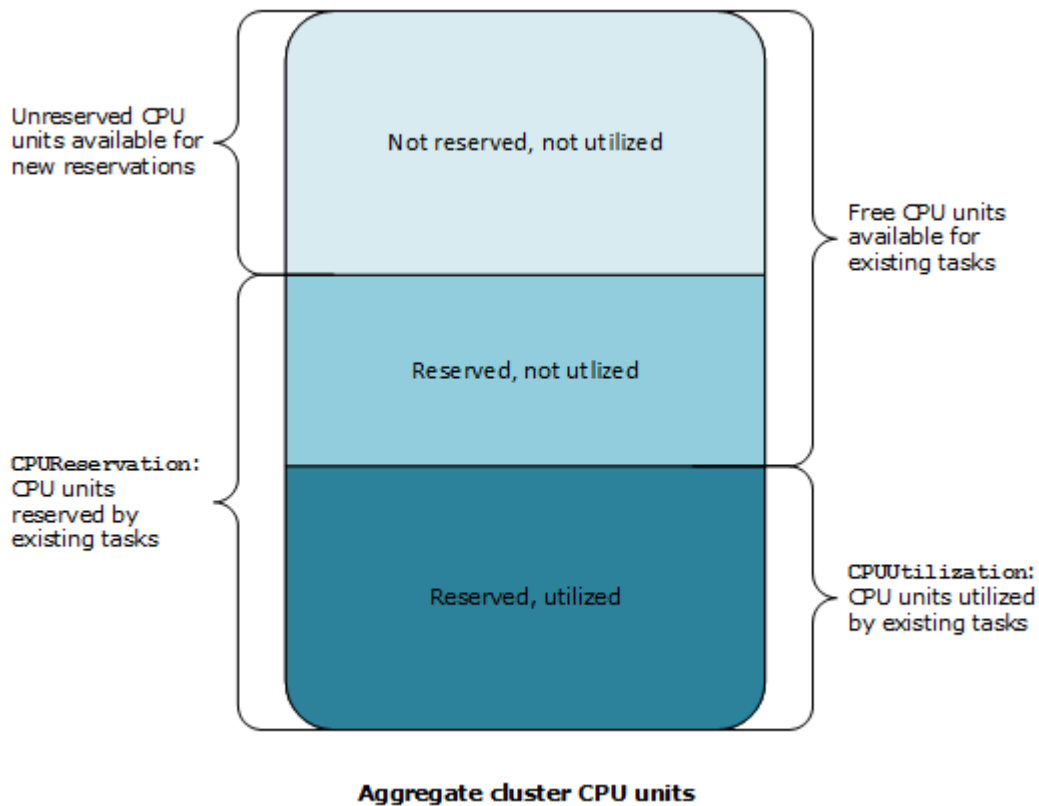
Quando esegui un'attività in un cluster, Amazon ECS analizza la definizione dell'attività e riserva le unità CPU aggregate, i MiB di memoria e GPUs quelle specificate nelle definizioni dei contenitori. Ogni minuto, Amazon ECS calcola il numero di unità CPU, MiB di memoria e GPUs che sono attualmente 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](#).



Parametri di utilizzo dei cluster di 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 su cluster con processi ospitati su AWS Fargate.

Parametri di utilizzo di CPU e 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'`c4.4xlarge` istanza e un'`c4.large` istanza. 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 dieci attività sono in esecuzione su questo cluster e ogni attività consuma 1,024 unità di CPU e 2,048 MiB di memoria, un totale di 10,240 unità di CPU e 20,480 MiB di memoria viene utilizzato nel cluster. Ciò corrisponde al 55% CloudWatch 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

Il parametro di utilizzo del file system EBS a livello di cluster viene misurato come il totale del file system EBS in uso nei processi in esecuzione sul cluster, diviso il totale dell'archiviazione del file system EBS allocato per tutti i processi nel cluster.

$$\text{Cluster EBS filesystem utilization} = \frac{(\text{Total GB of EBS filesystem used by tasks in cluster} \times 100)}{(\text{Total GB of EBS filesystem allocated to tasks in cluster})}$$

Parametri di utilizzo del servizio Amazon ECS

I parametri di utilizzo del servizio sono disponibili per CPU, memoria e, quando è presente un volume EBS collegato alle attività, per 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 di CPU e memoria è misurato come la percentuale di CPU e memoria utilizzata dai processi di Amazon ECS che appartengono a un servizio in un cluster rispetto alla CPU e alla memoria specificate nella definizione di attività del servizio.

Quando visualizzi queste metriche in CloudWatch, puoi scegliere diverse statistiche:

- **Media:** l'utilizzo medio di tutte le attività del servizio, calcolato utilizzando la formula seguente.
- **Minimo:** l'utilizzo dell'attività con il minor uso di risorse nel servizio, che rappresenta la percentuale di CPU o memoria utilizzata dall'attività che richiede meno risorse rispetto a quella specificata nella definizione dell'attività.
- **Massimo:** l'utilizzo dell'attività con il massimo uso di risorse nel servizio, che rappresenta la percentuale di CPU o memoria utilizzata dall'attività che richiede più risorse rispetto a quella specificata nella definizione dell'attività.

Le seguenti formule mostrano come viene calcolata la statistica Media:

$$\text{Service CPU utilization} = \frac{(\text{Total CPU units used by tasks in service}) \times 100}{(\text{Total CPU units specified in the task definition}) \times (\text{Number of tasks in the service})}$$

$$\text{Service memory utilization} = \frac{(\text{Total MiB of memory used by tasks in service} \times 100)}{(\text{Total MiB of memory specified in the task definition}) \times (\text{Number of tasks in the service})}$$

 Note

Le formule precedenti si applicano solo alla statistica Media. Le statistiche Minimo e Massimo rappresentano la singola attività con rispettivamente il minore e il massimo utilizzo delle risorse, anziché un calcolo aggregato per tutte le attività.

Amazon ECS raccoglie i parametri ogni 20 secondi. Ogni minuto, l'agente di container di Amazon ECS calcola il numero di unità di CPU e MiB di memoria attualmente in uso per ogni processo 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, consulta [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 il totale del file system EBS in uso nei processi che appartengono al servizio, diviso il totale dell'archiviazione del file system EBS allocato per tutti i processi che appartengono al servizio.

Conteggio dei processi **RUNNING** del servizio

Puoi utilizzare i CloudWatch parametri 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.

Casi d'uso di utilizzo del servizio Amazon ECS

L'elenco seguente fornisce informazioni su quando utilizzare i parametri di Amazon ECS.

- **Monitoraggio dell'utilizzo delle risorse:** utilizza statistiche medie per monitorare i modelli di consumo di CPU e memoria, stabilire linee di base per le prestazioni e individuare tendenze di graduale peggioramento delle prestazioni.
- **Ottimizzazione dei costi:** utilizza le statistiche medie per monitorare l'utilizzo delle risorse, dimensionare correttamente i container, regolare le prenotazioni in base ai modelli di utilizzo e implementare la scalabilità pianificata.
- **Benchmarking delle prestazioni:** utilizza le statistiche medie per confrontare le metriche tra le revisioni dei servizi, stabilire le prestazioni KPIs e convalidare i miglioramenti di ottimizzazione.
- **Rilevamento delle risorse:** utilizza le statistiche medie per identificare il fabbisogno minimo di risorse durante i periodi di inattività, impostare le prenotazioni appropriate e rilevare cali anomali.
- **Rilevamento delle anomalie:** utilizza le statistiche minime per individuare cali insoliti nell'utilizzo delle risorse che indicano potenziali problemi, come errori di inizializzazione o periodi di inattività imprevisti.
- **Perfezionamento delle policy di scalabilità:** utilizza le statistiche minime per stabilire soglie di riduzione orizzontale ottimali basate su un utilizzo minimo praticabile per prevenire un ridimensionamento aggressivo.
- **Pianificazione della capacità:** utilizza le statistiche massime per impostare le dimensioni appropriate delle attività e pianificare l'infrastruttura con margini di crescita sufficienti per i picchi di traffico.
- **Identificazione dei punti critici in termini di prestazioni:** utilizza le statistiche massime per rilevare i punti di saturazione delle risorse, identificare i colli di bottiglia e determinare quando aumentare le dimensioni delle attività.
- **Configurazione delle policy di scalabilità:** utilizza le statistiche massime per impostare soglie di aumento orizzontale ottimali basate sui modelli di picco e configurare la capacità di espansione.
- **Monitoraggio della conformità agli SLA:** utilizza il massimo delle statistiche per tenere traccia dei tempi di risposta di picco e dei tassi di errore per garantire che le prestazioni del servizio soddisfino gli standard. SLAs

Informazioni sui parametri correlati

Per informazioni su Container Insights, consulta i [casi d'uso di Container Insights](#) nella Guida CloudWatch per l'utente.

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 sono in uso EC2, 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 i tuoi processi utilizzano Fargate, puoi visualizzare lo stato delle istanze di container.

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 di container
- Modifica dello stato dell'attività
- Deployment state change (Cambio stato distribuzione)
- Operazioni di servizio

Note

Amazon ECS potrebbe aggiungere in futuro altri tipi di eventi, origini e dettagli. Se deserializzi i dati JSON di eventi nel codice, assicurati che l'applicazione sia in grado di gestire proprietà sconosciute al fine di evitare problemi se e quando queste proprietà aggiuntive verranno aggiunte.

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 operazioni di servizio specificano il servizio in 2 campi diversi:

- Per gli eventi generati da `create-service`, il servizio è sul campo `serviceName`.
- Per gli eventi generati da `update-service`, il servizio è sul campo `service`.

Se utilizzi strumenti automatizzati per gli eventi di servizio, devi codificare per entrambi i campi.

Per informazioni su come creare una regola per gli eventi di operazioni di servizio, consulta [Eventi di operazioni di servizi Amazon ECS](#).

Per ulteriori informazioni su come integrare Amazon ECS e EventBridge, consulta [Integrating Amazon and EventBridge Amazon ECS](#).

Eventi di modifica dello stato delle istanze di container di Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle istanze di container.

Puoi chiamare le operazioni `StartTask` `RunTask`, o `StopTask` API, direttamente o con o. Console di gestione AWS 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. Console di gestione AWS 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 backend 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 dei processi di Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle attività:

Puoi chiamare `StartTaskRunTask`, o le operazioni `StopTask` API, direttamente o con Console di gestione AWS 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`

Infatti EC2, l'agente container Amazon ECS monitora lo stato delle tue attività sulle tue istanze di container. L'agente del container di Amazon ECS segnala eventuali modifiche allo stato. Le modifiche dello stato potrebbero includere modifiche da `PENDING` a `RUNNING` o da `RUNNING` a `STOPPED`.

È possibile forzare l'annullamento della registrazione dell'istanza del contenitore sottostante con l'operazione `DeregisterContainerInstance` API e il `force` flag, direttamente o con l'opzione `o`.
Console di gestione AWS 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. Nota quanto segue in relazione ai campi:

- Lo stato di integrità dell'evento non è disponibile nell'evento di modifica dello stato del processo. Se è necessario lo stato di integrità del processo, è possibile eseguire [describe-tasks](#).
- Se i container utilizzano un'immagine ospitata con Amazon ECR, viene restituito il campo `imageDigest`.
- I valori per i campi `createdAt`, `connectivityAt`, `pullStartedAt`, `startedAt`, `pullStoppedAt` e `updatedAt` sono timestamp di stringhe ISO.
- Il valore `detail-type` è "ECS Task State Change".
- Quando l'evento viene generato per un processo interrotto, i campi `stoppedReason` e `stopCode` forniscono informazioni aggiuntive sul motivo per cui l'attività è stata interrotta (ad esempio, "User initiated").

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 gli eventi di attività Amazon ECS arrestate](#)

```
{
  "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": "ecs-fargate"
            },
            {
              "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. EC2

```

{
  "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 operazioni di servizi 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'attività in esecuzione.

SERVICE_DEPLOYMENT_IN_PROGRESS

L'implementazione di un servizio è in corso. L'implementazione del servizio può essere un rollback o la revisione di un nuovo servizio.

SERVICE_DEPLOYMENT_COMPLETED

L'implementazione di un servizio è in stato costante ed è completa. L'implementazione del servizio può essere un rollback o la distribuzione della revisione di un nuovo 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 il rilevamento 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 una delle destinazioni di VPC Lattice non è integra.

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

Lo scheduler del servizio è stato limitato a causa dei limiti di limitazione delle AWS Cloud Map API. Può verificarsi nei servizi configurati per utilizzare il rilevamento 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 sull'integrità delle 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 EC2, è stato tentato di avviare l'attività 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 operazioni di servizi 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-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 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 [Distribuisci 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 [Distribuisci 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 è possibile 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
        )

```

Nell'esempio seguente di Fargate viene riportata una funzione Lambda scritta in Python 3.9, la quale acquisisce eventi di modifica dello stato del processo e li salva nella tabella Amazon DynamoDB seguente:

```

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 container di Amazon ECS utilizzando Container Insights con osservabilità migliorata.

Container Insights raccoglie, aggrega e riepiloga parametri e log di applicazioni e microservizi containerizzati. Container Insights con osservabilità migliorata fornisce tutte le metriche di Container Insights, oltre a varie metriche aggiuntive per attività e container.

Container Insights scopre tutti i container in esecuzione in un cluster e raccoglie i dati sulle prestazioni a ogni livello dello stack delle 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.

I parametri riflettono solo le risorse con le attività in esecuzione durante l'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 che utilizzano Amazon EC2 e Fargates. Dopo aver configurato Container Insights con l'osservabilità

migliorata su Amazon ECS, Container Insights raccoglie automaticamente la telemetria dettagliata dell'infrastruttura, dal livello del cluster fino al livello del container nel tuo ambiente, e visualizza questi dati critici sulle prestazioni in pannelli di controllo curati, eliminando la complessità della configurazione dell'osservabilità. Per informazioni su come configurare Container Insights con osservabilità migliorata, consulta [Container Insights con osservabilità migliorata](#).

Ti consigliamo di utilizzare Container Insights con osservabilità avanzata invece di Container Insights, in quanto fornisce una visibilità dettagliata nell'ambiente del 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.](#)

Parametri aggiuntivi per le istanze gestite Amazon ECS

Nella tabella seguente sono elencati i parametri aggiuntivi disponibili per le istanze gestite da Amazon ECS quando si utilizza Container Insights.

Metrica	Description	Dimensioni	Unità
Instance0 SFilesyst emUtiliza tion	La percentuale di spazio totale del disco usata (volume).	ClusterName - quando ContainerInsights è abilitato ClusterName , CapacityP roviderName - quando ContainerInsights è abilitato ClusterName ,CapacityP roviderName ,Container InstanceId , EC2Instan	Percentuale

Metrica	Description	Dimensioni	Unità
		ceId - quando Enhanced Container Insights è abilitato	
InstanceDataFileSystemUtilization	La percentuale di spazio totale del disco usata (volume di dati).	ClusterName - quando Container Insights è abilitato ClusterName , CapacityProviderName - quando Container Insights è abilitato ClusterName , CapacityProviderName , ContainerInstanceId , EC2InstanceId - quando Enhanced Container Insights è abilitato	Percentuale

Monitoraggio delle istanze gestite da Amazon ECS

Il monitoraggio è una parte importante per mantenere l'affidabilità, la disponibilità e le prestazioni dei carichi di lavoro di Amazon ECS Managed Instances. AWS fornisce diversi strumenti e servizi per aiutarti a monitorare le applicazioni e l'infrastruttura containerizzate.

Monitoraggio di Container Insights

CloudWatch Container Insights offre un monitoraggio completo per le applicazioni e i microservizi containerizzati. Infatti, raccoglie, aggrega e riepiloga automaticamente parametri e log dalle applicazioni e dai microservizi containerizzati in esecuzione sulle istanze gestite da Amazon ECS.

Container Insights raccoglie i parametri a livello di cluster, attività e servizio, fornendo visibilità su:

- Utilizzo della CPU e della memoria
- Parametri sulle prestazioni di rete
- Utilizzo dell'archiviazione
- Prestazioni di processi e servizi

Le metriche sono disponibili nei CloudWatch dashboard e possono essere utilizzate per creare allarmi e risposte automatiche a problemi di prestazioni. Container Insights offre anche funzionalità di monitoraggio avanzate che aiutano a identificare e risolvere rapidamente i problemi.

Note

Container Insights viene fornito a un costo aggiuntivo. Per ulteriori informazioni sui prezzi, consulta [Prezzi di CloudWatch](#).

Monitoraggio delle istanze

Per monitorare l'infrastruttura sottostante che supporta i carichi di lavoro di Amazon ECS Managed Instances, puoi utilizzare i EC2 parametri di Amazon disponibili tramite CloudWatch

Le istanze gestite di Amazon ECS sono dotate di due volumi Amazon EBS:

- Volume root utilizzato per il filesystem del sistema operativo
- Volume di dati utilizzato dalle applicazioni

Quando Container Insights è abilitato, Amazon ECS pubblica automaticamente i parametri di utilizzo del sistema operativo e del file system di dati a livello di istanza.

I parametri disponibili per le istanze gestite di Amazon ECS includono:

- EC2 Metriche di Amazon: utilizzo della CPU, prestazioni di rete, operazioni su disco e controlli dello stato. Per ulteriori informazioni, consulta [Monitora le tue istanze utilizzando CloudWatch](#)
- Metriche di Amazon ECS (quando Container Insights è abilitato): utilizzo del sistema operativo e del volume di dati del file system. Per ulteriori informazioni, consulta i parametri di [Amazon ECS Container Insights](#)
- Metriche di Amazon EBS: IOPS, velocità effettiva, latenza di lettura e scrittura. Per ulteriori informazioni, consulta i parametri di [Amazon EBS CloudWatch](#)

Note

L' CloudWatch agente non può essere eseguito come daemon perché i daemon non sono supportati su Amazon ECS Managed Instances. Ciò significa che non sono disponibili

parametri aggiuntivi a livello di sistema che richiedono l'esecuzione dell' CloudWatch agente come daemon.

Queste metriche sono disponibili automaticamente senza l'installazione manuale dell'agente.

Monitoraggio dettagliato delle istanze gestite da Amazon ECS

CloudWatch offre due categorie di monitoraggio: monitoraggio di base e monitoraggio dettagliato. Per impostazione predefinita, l'istanza gestita è configurata con il monitoraggio base. Facoltativamente, è possibile abilitare il monitoraggio dettagliato per identificare e intervenire più rapidamente in caso di problemi operativi. Puoi attivare o disabilitare il monitoraggio dettagliato quando crei o aggiorni un provider di capacità delle istanze gestite da Amazon ECS.

L'abilitazione del monitoraggio dettagliato su un'istanza gestita non influisce sul monitoraggio dei volumi Amazon EBS collegati.

La seguente tabella evidenzia le differenze tra il monitoraggio base e il monitoraggio dettagliato delle istanze gestite.

Tipo di monitoraggio	Descrizione	Costi
Monitoraggio base	I parametri di verifica dello stato sono disponibili in periodi di 1 minuto. Tutti gli altri parametri sono disponibili in periodi di 5 minuti.	Nessun costo.
Monitoraggio dettagliato	Tutti i parametri, inclusi i parametri di controllo dello stato, sono disponibili in periodi di 1 minuto. Per ottenere questo tipo di dati, devi abilitarne esplicitamente la ricezione per l'istanza gestita. Per le istanze gestite per le quali hai abilitato il monitoraggio dettagliato, puoi ricevere inoltre i dati aggregati sui gruppi di istanze gestite simili.	Ti viene addebitato o un costo in base ai parametri a cui vengono inviate le istanze gestite di Amazon ECS. CloudWatch Non verrà addebitato o alcun costo per l'archiviazione dei dati. Per ulteriori informazioni, consulta

Tipo di monitoraggio	Descrizione	Costi
		il livello a pagamento e l'esempio 1 - Monitoraggio EC2 dettagliato nella CloudWatch pagina dei prezzi.

Autorizzazioni richieste

Per abilitare il monitoraggio dettagliato per un'istanza gestita, l'utente deve disporre dell'autorizzazione per utilizzare l'operazione API `MonitorInstances`. Per disabilitare il monitoraggio dettagliato per un'istanza gestita, l'utente deve disporre dell'autorizzazione per utilizzare l'operazione API `UnmonitorInstances`.

Determina lo stato delle attività di Amazon ECS utilizzando i controlli dell'integrità dei container

Quando crei la definizione di un processo, puoi configurare un controllo dell'integrità per i container. I controlli dell'integrità sono comandi che vengono eseguiti localmente su un container 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 dell'integrità Docker presente nell'immagine del container.

Quando un controllo dell'integrità è definito in una definizione di attività, il container esegue il processo di controllo dell'integrità all'interno del container, quindi valuta il codice di uscita per determinare lo stato dell'applicazione.

Il controllo dell'integrità comprende i seguenti parametri:

- **Comando:** 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. Utilizza `CMD-SHELL` quando hai bisogno

di funzionalità della shell come pipe, reindirizzamenti, concatenazione di comandi o espansione delle variabili di ambiente. Ad esempio, CMD-SHELL consente di utilizzare comandi come `curl -f http://localhost/ || exit 1` nel punto in cui la shell interpreta l'operatore `||`. Utilizza CMD per comandi semplici che non richiedono l'interpretazione della shell.

- Intervallo: periodo di tempo (in secondi) tra ogni controllo dell'integrità.
- Timeout: periodo di tempo (in secondi) per cui attendere che un controllo dell'integrità venga superato prima di considerarlo un errore.
- Tentativi: il numero di volte in cui riprovare un controllo dell'integrità non riuscito prima che il container venga considerato non integro.
- Periodo di inizio: 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.

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 dell'integrità nella definizione di un processo, consulta [Controllo dell'integrità](#).

Di seguito vengono descritti i valori dello stato di integrità possibili per un container:

- 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 per il controllo dell'integrità vengono eseguiti nel container. Pertanto, è necessario includere i comandi nell'immagine del container.

Il controllo dell'integrità si connette all'applicazione tramite l'interfaccia di loopback del container all'indirizzo `localhost` o `127.0.0.1`. Il codice di uscita `0` indica che il processo ha avuto esito positivo, mentre un codice di uscita diverso da zero indica che ha avuto esito negativo.

Quando utilizzi i controlli dell'integrità del container, 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 dell'integrità dei container sono supportati per le attività Fargate se utilizzi una piattaforma Linux versione 1.1.0 o successiva o una piattaforma Windows versione 1.1.0 o successiva.

In che modo Amazon ECS determina lo stato delle attività

I container essenziali e con il comando per il controllo dell'integrità nella definizione dell'attività sono gli unici considerati per determinare lo stato dell'attività.

Le seguenti regole sono valutate in ordine:

1. Se lo stato di un container essenziale è UNHEALTHY, lo stato dell'attività è UNHEALTHY.
2. Se lo stato di un container essenziale è UNKNOWN, lo stato dell'attività è UNKNOWN.
3. Se lo stato di tutti i container essenziali è HEALTHY, lo stato dell'attività è HEALTHY.

Considera il seguente esempio di integrità di un'attività con 2 container essenziali.

Integrità del container 1	Integrità del container 2	Integrità delle attività
UNHEALTHY	UNKNOWN	UNHEALTHY
UNHEALTHY	HEALTHY	UNHEALTHY
HEALTHY	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY

Considera il seguente esempio di integrità di un'attività con 3 container.

Integrità del container 1	Integrità del container 2	Integrità del container 3	Integrità delle attività
UNHEALTHY	UNKNOWN	UNKNOWN	UNHEALTHY
UNHEALTHY	UNKNOWN	HEALTHY	UNHEALTHY
UNHEALTHY	HEALTHY	HEALTHY	UNHEALTHY

Integrità del container 1	Integrità del container 2	Integrità del container 3	Integrità delle attività
HEALTHY	UNKNOWN	HEALTHY	UNKNOWN
HEALTHY	UNKNOWN	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY	HEALTHY

In che modo il controllo dell'integrità viene influenzato dalle disconnessioni degli agenti

Se l'agente del container Amazon ECS si disconnette dal servizio Amazon ECS, non causerà la transizione di un container a uno stato UNHEALTHY. Infatti, è stato progettato per garantire che i container rimangano in funzione durante il riavvio degli agenti o la temporanea indisponibilità. Lo stato del controllo dell'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 dell'integrità. Non ci sono ipotesi sullo stato dei controlli dell'integrità del container.

Visualizzazione dell'integrità dei container di Amazon ECS

Puoi visualizzare lo stato del container nella console e utilizzare l'API nella risposta `DescribeTasks`. 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. Ricorda di utilizzare `2&1` per ottenere sia le informazioni `stdout` che `stderr`. L'esempio seguente utilizza `CMD-SHELL` perché richiede funzionalità di shell come gli operatori pipe (`>>`) e gli operatori logici OR (`||`):

```
"command": [
  "CMD-SHELL",
  "curl -f http://localhost/ >> /proc/1/fd/1 2>&1 || exit 1"
],
```

Monitora l'integrità delle istanze di 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 una versione successiva o una AWS CLI versione 2.3.6 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 di container, esegui `describe-container-instances` con l'opzione `CONTAINER_INSTANCE_HEALTH`.

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 delle istanze di container

Quando `overallStatus` è in uno stato diverso da OK, prova quanto segue:

- Attendi e poi esegui `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. Dashboard AWS Health

Individua 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](#).

Dopo aver creato il ruolo, collega una policy con le seguenti autorizzazioni e allegala a tale ruolo.

- `logs:PutLogEvents`
- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:DescribeLogStreams`
- `logs:DescribeLogGroups`
- `logs:PutRetentionPolicy`
- `xray:PutTraceSegments`
- `xray:PutTelemetryRecords`
- `xray:GetSamplingRules`
- `xray:GetSamplingTargets`
- `xray:GetSamplingStatisticSummaries`
- `ssm:GetParameters`

Specificando il AWS Distro for OpenTelemetry sidecar da integrare nella definizione dell'attività AWS X-Ray

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 di Amazon ECS attraverso la nuova 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 di 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. L'esperienza della console Amazon ECS semplifica questo processo aggiungendo l'opzione Utilizza raccolta parametri durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta [Creazione di una definizione di attività di Amazon ECS attraverso la nuova console](#).

Le metriche dell'applicazione vengono esportate in CloudWatch Logs con il nome del gruppo di log `/aws/ecs/application/metrics` e le metriche possono essere visualizzate 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 container di 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 di Amazon ECS si occupa del ruolo IAM di esecuzione dei processi per tuo conto, ma il ruolo IAM del processo deve essere creato manualmente e aggiunto alla tua definizione di 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 l' OpenTelemetry integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta [Individua 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 Console di gestione AWS 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 Tipo di entità attendibile, seleziona Servizio AWS.
 4. Per Servizio o caso d'uso, seleziona Elastic Container Service e poi il caso d'uso Attività Elastic Container Service.
 5. Scegli Next (Successivo).
 6. Nella sezione Aggiungi autorizzazioni AWSDistroOpenTelemetryPolicyForXray, cerca e quindi seleziona la politica.
 7. (Facoltativo) Impostare un [limite delle autorizzazioni](#). Questa è una funzionalità 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. Specifica un nome del ruolo o un suffisso del nome del ruolo per facilitare l'identificazione dello 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 account 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 il 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 di Amazon ECS attraverso la nuova 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. L'esperienza della console Amazon ECS semplifica questo processo aggiungendo l'opzione Utilizza raccolta parametri durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta [Creazione di una definizione di attività di Amazon ECS attraverso la nuova 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. Il ruolo IAM di questo processo deve essere creato manualmente prima di registrare la definizione di attività. Per ulteriori informazioni sulla creazione di un ruolo del processo, consulta [Ruolo IAM dell'attività Amazon ECS](#).

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. L'esperienza della console di Amazon ECS si occupa del ruolo IAM di esecuzione dei processi per tuo conto, ma il ruolo IAM del processo 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 l' OpenTelemetry integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta [Individua le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni](#).

Le seguenti autorizzazioni sono necessarie per l' OpenTelemetry integrazione di AWS Distro con Amazon Managed Service for Prometheus:

- registri: PutLogEvents
- registri: CreateLogGroup
- registri: CreateLogStream
- registri: DescribeLogStreams
- registri: DescribeLogGroups

- cloudwatch: PutMetricData

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 di Amazon ECS attraverso la nuova 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"
}

```

Registrazione delle chiamate API Amazon ECS tramite 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 le chiamate dalla console Amazon ECS e le chiamate di codice alle operazioni API di 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 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 Console di gestione AWS 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 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](#).

Eventi relativi ai dati di Amazon ECS in CloudTrail

[Gli eventi relativi ai dati](#) forniscono informazioni sulle operazioni eseguite sulle risorse o all'interno di una risorsa. Per impostazione predefinita, CloudTrail non registra gli eventi relativi ai dati. La cronologia CloudTrail degli eventi non registra gli eventi relativi ai dati.

Per gli eventi di dati sono previsti costi aggiuntivi. Per ulteriori informazioni sui CloudTrail prezzi, consulta la sezione [AWS CloudTrail Prezzi](#).

Puoi registrare gli eventi relativi ai dati per i tipi di risorse Amazon ECS utilizzando la CloudTrail console o AWS CLI le operazioni CloudTrail API. Per ulteriori informazioni su come registrare gli eventi relativi ai dati, consulta [Registrazione di eventi relativi ai dati con Console di gestione AWS](#) e [Registrazione di eventi di dati con AWS Command Line Interface](#) nella Guida per l'utente AWS CloudTrail .

La tabella seguente elenca i tipi di risorse Amazon ECS per i quali è possibile registrare gli eventi relativi ai dati. La colonna Data event type (console) mostra il valore da scegliere dall'elenco Data event type (console) sulla CloudTrail console. La colonna del valore resources.type mostra il resources.type valore, da specificare durante la configurazione dei selettori di eventi avanzati utilizzando o. AWS CLI CloudTrail APIs La CloudTrail colonna Dati APIs registrati mostra le chiamate API registrate per il tipo di risorsa. CloudTrail

Tipo di evento di dati (console)	valore resources.type	Dati registrati APIs su CloudTrail
AwsApiCall	AWS::ECS::ContainerInstance	- ecs:Poll(per EC2 e Amazon ECS Managed Instances) - ecs:StartTelemetrySession (per EC2 e Amazon ECS Managed Instances) - ecs:PutSystemLogEvents (per istanze gestite Amazon ECS)

È possibile configurare selettori di eventi avanzati per filtrare i campi eventName, readOnly e resources.ARN per registrare solo gli eventi importanti per l'utente. Per ulteriori informazioni su questi campi, consulta [AdvancedFieldSelector](#) in Riferimento API AWS CloudTrail.

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'CreateServiceazione.

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.

Visualizzazione dei CloudWatch log Live Tail per i servizi e le attività di Amazon ECS

CloudWatch Logs Live Tail ti aiuta a risolvere rapidamente gli incidenti visualizzando un elenco in streaming dei nuovi eventi di registro man mano che vengono inseriti. È possibile visualizzare gli eventi delle attività e dei servizi nella console Amazon ECS. Questa visualizzazione offre una visione coerente dello stato dei processi e dei servizi.

Ogni attività in un servizio Amazon ECS ha flussi di log dedicati per ogni container. Se un servizio aumenta verticalmente, ogni istanza del processo ha il suo set di flussi di log. La convenzione di denominazione per i flussi di log segue lo schema:

```
log-group-prefix/container-name/task-id
```

Un singolo processo scrive sullo stesso flusso di log durante il suo ciclo di vita. Il flusso di log contiene i messaggi provenienti dai container di quel processo e anche qualsiasi output del codice dell'applicazione. Ogni messaggio ha un timestamp, compresi i log personalizzati.

Note

Le sessioni Live Tail comportano costi al minuto in base al tempo di utilizzo della sessione. Per ulteriori informazioni sui prezzi, consulta la pagina [CloudWatch dei prezzi di Amazon](#).

Autorizzazioni richieste

Le seguenti autorizzazioni sono necessarie all'utente IAM della console per avviare e interrompere le sessioni di CloudWatch Logs Live Tail:

- `logs:DescribeLogGroups`

- `logs:StartLiveTail`
- `logs:StopLiveTail`

Procedura

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Determina la risorsa che si desidera visualizzare

Risorsa	Fasi	
Processi	a. Nella pagina Cluster, scegliere il cluster. Si apre la pagina dei dettagli del cluster. b. Selezionare la scheda Tasks (Attività). Si apre la pagina dei dettagli del processo. c. Scegli la scheda Log.	
Servizi	a. Nella pagina Cluster, scegliere il cluster. Si apre la pagina dei dettagli del cluster. b. Scegli il servizio. Si apre la pagina dei dettagli del servizio. c. Scegli la scheda Log.	

3. Scegli CloudWatch Logs Live Tail, quindi scegli Start.
4. (Facoltativo) Per filtrare i flussi, in Filtro, per Seleziona gruppi di log, scegli il gruppo di log.

AWS CLI riferimenti

Puoi anche usare il AWS CLI per avviare le sessioni di Live Tail per CloudWatch Logs. Le seguenti risorse forniscono informazioni dettagliate sull'utilizzo di AWS CLI con Live Tail:

- [start-live-tail command reference](#): sintassi completa del comando, parametri ed esempi per il `aws logs start-live-tail` comando.
- CloudWatch Guida per [l'utente di Logs Live Tail - Guida](#) completa che include AWS CLI l'utilizzo con modalità di sola stampa e interattiva.
- StartLiveTail Esempi [SDK - Esempi](#) programmatici per l'utilizzo dell'API con vari. StartLiveTail AWS SDKs

Monitora i carichi di lavoro utilizzando i metadati Amazon ECS

È possibile utilizzare i metadati delle attività e dei container 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 container che forniscono l'ID Docker, il nome e i dettagli dell'immagine.

Questo fornisce visibilità all'interno del container.

- Impostazioni di rete come indirizzi IP, sottoreti e modalità di rete.

Questo facilita la configurazione e la risoluzione dei problemi della rete.

- Stato e integrità dell'attività

Questo consente di capire se le attività sono in esecuzione.

È possibile visualizzare i metadati utilizzando uno dei metodi seguenti:

- 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 container

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 attività:

- `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI`: l'endpoint URL HTTP completo per l'SDK da utilizzare quando si effettua una richiesta di credenziali. Sono inclusi sia lo schema che l'host. Per ulteriori informazioni, consulta [Container credential provider nella Guida di riferimento AWS SDKs](#).
- `ECS_CONTAINER_METADATA_URI_V4`: l'indirizzo della versione 4 dei metadati dell'attività. Per ulteriori informazioni, consulta [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#).
- `ECS_CONTAINER_METADATA_URI`: l'indirizzo della versione 3 dei metadati dell'attività. Per ulteriori informazioni, consulta [Versione 3 degli endpoint dei metadati delle attività di Amazon ECS](#).
- `ECS_AGENT_URI`: l'indirizzo di base per i diversi endpoint supportati da Fargate. Per ulteriori informazioni, consulta:
 - [Endpoint di protezione scalabile di Amazon ECS](#)
 - [Endpoint di iniezione di guasti di Amazon ECS](#)
- `AWS_EXECUTION_ENV`: le informazioni sull'opzione di calcolo 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 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 dell'immagine e dell'attività Amazon ECS](#).

Argomenti

- [Posizioni dei file di metadati dei container](#)
- [Attivazione dei metadati dei container Amazon ECS](#)
- [Formato del file di metadati di container di 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 container 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 del file di metadati di container di 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 di Amazon ECS disponibili per le attività sulle istanze gestite da Amazon ECS

Amazon ECS sulle istanze gestite da Amazon ECS 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. La versione 4 dell'endpoint dei metadati delle attività è disponibile per le attività Amazon ECS sulle istanze gestite da Amazon ECS.

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

- [La versione 4 dell'endpoint dei metadati delle attività di Amazon ECS per le attività sulle istanze gestite da Amazon ECS](#)

La versione 4 dell'endpoint dei metadati delle attività di Amazon ECS per le attività sulle istanze gestite da Amazon ECS

Important

Se utilizzi attività Amazon ECS ospitate su EC2 istanze Amazon, consulta l'endpoint di metadati delle [attività Amazon ECS](#).

A partire dalle istanze gestite da Amazon ECS, 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 delle attività sono per impostazione predefinita per tutte le attività Amazon ECS eseguite sulle istanze gestite da Amazon ECS.

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 della versione 4 dell'endpoint dei metadati delle attività per le istanze gestite da Amazon ECS

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 [La risposta JSON della V4 dei metadati delle attività di Amazon ECS sulle istanze gestite da Amazon ECS](#).

`${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

I processi di Amazon ECS sulle istanze gestite da Amazon ECS richiedono che il container venga eseguito per circa 1 secondo prima di restituire le statistiche del container.

`${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

I processi di Amazon ECS sulle istanze gestite da Amazon ECS richiedono che il container venga eseguito per circa 1 secondo prima di restituire le statistiche del container.

La risposta JSON della V4 dei metadati delle attività di Amazon ECS sulle istanze gestite da Amazon ECS

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.

ServiceName

Il nome del servizio a cui appartiene l'attività. ServiceName verrà visualizzato se l'attività è associata a un servizio.

VPCID

L'ID VPC dell'istanza di container.

TaskARN

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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.

LaunchType

Il tipo di avvio utilizzato dall'attività. Per le attività in esecuzione sulle istanze gestite da Amazon ECS, il tipo di avvio è `MANAGED_INSTANCES`.

Containers

Un elenco di metadati dei container per ogni container associato all'attività.

DockerId

L'ID Docker per il container.

Quando utilizzi le istanze gestite da Amazon ECS, 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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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.

LogOptions

Le opzioni del driver di log definite per il container.

ContainerARN

Il nome della risorsa Amazon (ARN) del container.

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 omesso.

Snapshotter

Il snapshotter usato da containerd per scaricare l'immagine di container. I valori validi sono `overlayfs`, che è l'impostazione predefinita, e `soci`, ovvero il valore utilizzato durante il caricamento lento con un indice SOCI.

RestartCount

Il numero di volte in cui il container è stato riavviato.

Note

I metadati `RestartCount` sono inclusi solo se è abilitata una policy di riavvio per il container. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questa funzionalità utilizza il servizio di sincronizzazione oraria di Amazon per misurare la precisione dell'orologio e fornire il suo limite di errore per i container. 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.

FaultInjectionEnabled

Indica se l'iniezione di guasti è abilitata per l'attività. Per ulteriori informazioni, consulta [Utilizzare l'iniezione di guasti con i carichi di lavoro Amazon ECS e Fargate..](#)

Esempi della V4 dei metadati delle attività di Amazon ECS per le attività sulle istanze gestite da Amazon ECS

Gli esempi seguenti mostrano gli output di esempio degli endpoint dei metadati dei processi per i processi di Amazon ECS in esecuzione sulle istanze gestite da Amazon ECS.

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": "400c0466d1e54c0691aae0f86e0f9dc6-2531612879",
  "Name": "nginx",
  "DockerName": "nginx",
  "Image": "public.ecr.aws/nginx/nginx:latest",
  "ImageID": "sha256:d5f28ef21aabddd098f3dbc21fe5b7a7d7a184720bc07da0b6c9b9820e97f25e",
  "Labels": {
    "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-west-2:111122223333:cluster/managed-
instances-cluster",
    "com.amazonaws.ecs.container-name": "curl",
    "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/managed-
instances-cluster/400c0466d1e54c0691aae0f86e0f9dc6",
    "com.amazonaws.ecs.task-definition-family": "ecs-managed-instances-task-def",
    "com.amazonaws.ecs.task-definition-version": "7"
  },
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": { "CPU": 2 },
}
```

```

"CreatedAt": "2025-09-24T19:39:04.88336233Z",
"StartedAt": "2025-09-24T19:39:04.88336233Z",
"Type": "NORMAL",
"LogDriver": "awslogs",
"LogOptions": {
  "awslogs-create-group": "true",
  "awslogs-group": "/ecs/managed-instances-task-def",
  "awslogs-region": "us-west-2",
  "awslogs-stream": "ecs/nginx/400c0466d1e54c0691aae0f86e0f9dc6"
},
"ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/managed-instances-
cluster/400c0466d1e54c0691aae0f86e0f9dc6/3703f0e4-a351-4f1e-a0e7-6981ea7adc8b",
"Networks": [
  {
    "NetworkMode": "awsvpc",
    "IPv4Addresses": ["172.31.62.223"],
    "AttachmentIndex": 0,
    "MACAddress": "0e:41:0b:1e:f2:fb",
    "IPv4SubnetCIDRBlock": "172.31.48.0/20",
    "PrivateDNSName": "ip-172-31-62-223.us-west-2.compute.internal",
    "SubnetGatewayIpv4Address": "172.31.48.1/20"
  }
],
"Snapshotter": "overlayfs"
}

```

Esempi della V4 dei metadati delle attività di Amazon ECS per le attività sulle istanze gestite da Amazon ECS

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-west-2:111122223333:cluster/managed-instances-cluster",
  "TaskARN": "arn:aws:ecs:us-west-2:111122223333:task/managed-instances-
cluster/400c0466d1e54c0691aae0f86e0f9dc6",
  "Family": "managed-instances-task-def",
  "Revision": "7",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Limits": { "CPU": 1, "Memory": 3072 },
  "PullStartedAt": "2025-09-24T19:38:58.682942001Z",

```



```

"PullStoppedAt": "2025-09-24T19:39:03.091597524Z",
"AvailabilityZone": "us-west-2d",
"LaunchType": "MANAGED_INSTANCES",
"Containers": [
  {
    "DockerId": "400c0466d1e54c0691aae0f86e0f9dc6-2531612879",
    "Name": "curl",
    "DockerName": "curl",
    "Image": "nginx",
    "ImageID":
"sha256:d5f28ef21aabddd098f3dbc21fe5b7a7d7a184720bc07da0b6c9b9820e97f25e",
    "Labels": {
      "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-west-2:111122223333:cluster/
managed-instances-cluster",
      "com.amazonaws.ecs.container-name": "nginx",
      "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/managed-
instances-cluster/400c0466d1e54c0691aae0f86e0f9dc6",
      "com.amazonaws.ecs.task-definition-family": "managed-instances-task-def",
      "com.amazonaws.ecs.task-definition-version": "7"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": { "CPU": 2 },
    "CreatedAt": "2025-09-24T19:39:04.88336233Z",
    "StartedAt": "2025-09-24T19:39:04.88336233Z",
    "Type": "NORMAL",
    "LogDriver": "awslogs",
    "LogOptions": {
      "awslogs-create-group": "true",
      "awslogs-group": "/ecs/managed-instances-task-def",
      "awslogs-region": "us-west-2",
      "awslogs-stream": "ecs/nginx/400c0466d1e54c0691aae0f86e0f9dc6"
    },
    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/managed-instances-
cluster/400c0466d1e54c0691aae0f86e0f9dc6/3703f0e4-a351-4f1e-a0e7-6981ea7adc8b",
    "Networks": [
      {
        "NetworkMode": "awsvpc",
        "IPv4Addresses": ["172.31.62.223"],
        "AttachmentIndex": 0,
        "MACAddress": "0e:41:0b:1e:f2:fb",
        "IPv4SubnetCIDRBlock": "172.31.48.0/20",
        "PrivateDNSName": "ip-172-31-62-223.us-west-2.compute.internal",
        "SubnetGatewayIpv4Address": "172.31.48.1/20"
      }
    ]
  }
]

```

```

    }
  ],
  "Snapshotter": "overlayfs"
}
],
"ServiceName": "exec-service-2",
"ClockDrift": {
  "ClockErrorBound": 0.14120749999999999,
  "ReferenceTimestamp": "2025-09-24T19:48:37Z",
  "ClockSynchronizationStatus": "SYNCHRONIZED"
},
"FaultInjectionEnabled": false
}

```

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.

```

{
  "400c0466d1e54c0691aae0f86e0f9dc6-2531612879": {
    "read": "2025-09-24T19:51:54.899736614Z",
    "preread": "2025-09-24T19:51:44.901199024Z",
    "pids_stats": {},
    "blkio_stats": {
      "io_service_bytes_recursive": [
        { "major": 259, "minor": 1, "op": "read", "value": 0 },
        { "major": 259, "minor": 1, "op": "write", "value": 131072 },
        { "major": 259, "minor": 0, "op": "read", "value": 48173056 },
        { "major": 259, "minor": 0, "op": "write", "value": 0 },
        { "major": 252, "minor": 0, "op": "read", "value": 48173056 },
        { "major": 252, "minor": 0, "op": "write", "value": 0 }
      ],
      "io_serviced_recursive": null,
      "io_queue_recursive": null,
      "io_service_time_recursive": null,
      "io_wait_time_recursive": null,
      "io_merged_recursive": null,
      "io_time_recursive": null,
      "sectors_recursive": null
    },
    "num_procs": 0,
  }
}

```

```

"storage_stats": {},
"cpu_stats": {
  "cpu_usage": {
    "total_usage": 670462000,
    "usage_in_kernelmode": 276072000,
    "usage_in_usermode": 394389000
  },
  "system_cpu_usage": 787660000000,
  "online_cpus": 1,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"precpu_stats": {
  "cpu_usage": {
    "total_usage": 663809000,
    "usage_in_kernelmode": 273333000,
    "usage_in_usermode": 390476000
  },
  "system_cpu_usage": 777710000000,
  "online_cpus": 1,
  "throttling_data": {
    "periods": 0,
    "throttled_periods": 0,
    "throttled_time": 0
  }
},
"memory_stats": {
  "usage": 83562496,
  "stats": {
    "active_anon": 24576,
    "active_file": 258048,
    "anon": 32264192,
    "anon_thp": 0,
    "file": 48533504,
    "file_dirty": 0,
    "file_mapped": 36286464,
    "file_writeback": 0,
    "inactive_anon": 32243712,
    "inactive_file": 48271360,
    "kernel_stack": 442368,
    "pgactivate": 6,

```

```

    "pgdeactivate": 0,
    "pgfault": 18936,
    "pglazyfree": 0,
    "pglazyfreed": 0,
    "pgmajfault": 311,
    "pgrefill": 0,
    "pgscan": 0,
    "pgsteal": 0,
    "shmem": 4096,
    "slab": 1598960,
    "slab_reclaimable": 1080040,
    "slab_unreclaimable": 518920,
    "sock": 0,
    "thp_collapse_alloc": 0,
    "thp_fault_alloc": 0,
    "unevictable": 0,
    "workingset_activate": 0,
    "workingset_nodereclaim": 0,
    "workingset_refault": 0
  },
  "limit": 18446744073709551615
},
"name": "nginx",
"id": "400c0466d1e54c0691aae0f86e0f9dc6-2531612879"
}
}

```

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 [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#).
- 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 awsipc rete, l'agente contenitore Amazon ECS deve avere almeno una versione. 1.54.0 Per ulteriori informazioni, consulta [Versione 3 degli endpoint dei metadati delle attività di Amazon ECS](#).

- 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 awsipc rete, l'agente contenitore Amazon ECS deve avere almeno una versione. 1.54.0 Per ulteriori informazioni, consulta [Versione 2 degli endpoint dei metadati delle attività di Amazon ECS](#).

Se la tua attività Amazon ECS è ospitata su Amazon EC2 o se utilizza la modalità di host rete ed è ospitata su Amazon ECS Managed Instances, puoi anche accedere ai metadati dell'host dell'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
```

Se la tua attività Amazon ECS è ospitata su Amazon EC2 e in una configurazione IPv6 -only, puoi accedere ai metadati dell'host delle attività utilizzando l' IPv6 endpoint IMDS. Il comando seguente, se eseguito dall'istanza che ospita l'attività, elenca l'ID dell'istanza host su cui si trova. IPv6

```
curl http://[fd00:ec2::254]/latest/meta-data/instance-id
```

Per accedere all'endpoint IPv6 IMDS, abilita l'endpoint IPv6 IMDS sull'istanza del contenitore e configura anche la modalità endpoint del servizio di metadati utilizzando il provider di credenziali IMDS per l'SDK scelto. IPv6 Per ulteriori informazioni sull'abilitazione dell'endpoint IPv6 IMDS per l'istanza del contenitore, consulta le opzioni del [servizio di metadati dell'istanza nella](#) Amazon EC2 User Guide. Per ulteriori informazioni sul provider di credenziali IMDS per SDKs, consulta il provider di credenziali [IMDS nella and Tools Reference Guide](#).AWS SDKs

Note

L'endpoint IPv6 IMDS non è accessibile quando l'impostazione dell'account è abilitata. awsipcTrunking Per accedere alle credenziali del ruolo IAM dell'istanza di container quando awsipcTrunking è abilitato, puoi invece utilizzare un ruolo IAM dell'attività. Per ulteriori informazioni sui ruoli IAM delle attività, consulta [Ruolo IAM dell'attività Amazon ECS](#).

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 è possibile ottenere utilizzando l'endpoint, consulta [Categorie di metadati dell'istanza](#).

Versione 4 degli endpoint dei metadati delle attività di Amazon ECS

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 Console di gestione AWS. Per ulteriori informazioni, consulta [Monitora i container di 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 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 [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](#).

Per l'output di esempio dei metadati delle attività, consulta [Esempi della V4 dei metadati delle attività di 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 relativa alla 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`.

Important

Quando usi l'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags`, tieni presente che ogni chiamata invia fino a due richieste API a `ecs:ListTagsForResource` (una per i tag delle istanze di container e una per i tag delle attività) e che tutti i container sidecar dell'attività possono effettuare queste chiamate per tuo conto. Le chiamate frequenti agli endpoint possono comportare una limitazione delle API.

Valuta la possibilità di implementare strategie di caching o batch per ridurre la frequenza delle chiamate, specialmente nelle applicazioni ad alto traffico, e di eseguire il debugging dei problemi di limitazione delle API utilizzando AWS CloudTrail. Per informazioni sulle

limitazioni per l'API `ListTagsForResource`, consulta [Richiedi la limitazione per l'API di Amazon ECS](#) nella Documentazione di riferimento delle API di Amazon Elastic Container Service. Per ulteriori informazioni sul debug delle chiamate API di Amazon ECS utilizzando, consulta AWS CloudTrail. [Registrazione delle chiamate API Amazon ECS tramite AWS CloudTrail](#)

`${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 la documentazione dell'API [ContainerStats](#) Docker.

Per le attività di Amazon ECS che utilizzano le modalità di `bridge` rete `awsvpc` 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 `awsvpc` 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 relativa alla 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

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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 (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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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 container è stato riavviato.

 Note

I metadati `RestartCount` sono inclusi solo se è abilitata una policy di riavvio per il container. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

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 della V4 dei metadati delle attività Amazon ECS

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": "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 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-a6bcc3dbadf9cfe85300",
"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
}
}
}

```

Versione 3 degli endpoint dei metadati delle attività di Amazon ECS

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 Fargate sulla versione della piattaforma v1.3.0 o successiva e per le attività che utilizzano EC2 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 container Amazon ECS e utilizza 1.54.0 la modalità di rete. awsipc 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 Fargate e versioni della piattaforma precedenti alla v1.3.0, è supportato l'endpoint dei metadati dei processi versione 2. Per ulteriori informazioni, consulta [Versione 2 degli endpoint dei metadati delle attività di Amazon ECS](#).

Percorsi dell'endpoint dei metadati delle attività 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 per tutti i contenitori associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta [Risposta JSON relativa alla 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 relativa alla 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

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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 il servizio di sincronizzazione oraria di Amazon per misurare la precisione dell'orologio e fornire il suo limite di errore per i container. 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 della V3 dei metadati delle attività Amazon ECS

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"  
}
```

Versione 2 degli endpoint dei metadati delle attività di Amazon ECS

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 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 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

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 awsvpc rete ricevono un indirizzo locale all'interno di un intervallo di IPv4 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 con Fargate che utilizzano la versione della piattaforma 1.1.0 o successive. Per ulteriori informazioni, consulta [Versioni della piattaforma Fargate per Amazon ECS](#).
- Le attività EC2 che lo utilizzano utilizzano anche la modalità di awsvpc 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

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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 il servizio di sincronizzazione oraria di Amazon per misurare la precisione dell'orologio e fornire il suo limite di errore per i container. 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 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 metadati delle attività Amazon ECS versione 4 per le attività su Fargate](#)
- [Endpoint metadati delle attività Amazon ECS versione 3 per le attività su Fargate](#)

Endpoint metadati delle attività Amazon ECS versione 4 per le 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 relativa alla V4 dei metadati delle attività Amazon ECS 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 relativa alla V4 dei metadati delle attività Amazon ECS 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.

ServiceName

Il nome del servizio a cui appartiene l'attività. ServiceName apparirà per Fargate le attività associate a un servizio.

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

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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 usato da containerd per scaricare l'immagine di container. 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 container è stato riavviato.

Note

I metadati `RestartCount` sono inclusi solo se è abilitata una policy di riavvio per il container. Per ulteriori informazioni, consulta [Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container](#).

ClockDrift

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questa funzionalità utilizza il servizio di sincronizzazione oraria di Amazon per misurare la precisione dell'orologio e fornire il suo limite di errore per i container. 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 relativi alla V4 dei metadati delle attività Amazon ECS per le 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 relativi alla V4 dei metadati delle attività Amazon ECS per le 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,
    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_pgpout": 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 metadati delle attività Amazon ECS versione 3 per le 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 relativa alla V3 dei metadati delle attività Amazon ECS 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 relativa alla V3 dei metadati delle attività Amazon ECS 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

Il nome della risorsa Amazon (ARN) dell'attività a cui appartiene il container.

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 file digest SHA-256 del manifesto delle immagini. Questo è il digest che può essere utilizzato per estrarre l'immagine utilizzando il formato `repository-url/image@sha256:digest`.

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 il servizio di sincronizzazione oraria di Amazon per misurare la precisione dell'orologio e fornire il suo limite di errore per i container. 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 relativi alla V3 dei metadati delle attività Amazon ECS per le 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 del container di 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 il Monitoraggio del runtime

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 gestisce 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 CSPM, 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 delle seguenti chiamate API di Amazon ECS:

- [DescribeClusters](#)

I parametri di risposta includono il tag del Monitoraggio del runtime (quando è impostato) quando si utilizza l'opzione `--include TAGS`.

- [DescribeTasks](#)

Per Fargate, i parametri di risposta includono il contenitore GuardDuty sidecar.

- [ListAccountSettings](#)

I parametri di risposta includono l'impostazione dell'account del Monitoraggio del runtime, impostata dall'amministratore della sicurezza.

- I dati di introspezione dell'agente del container. Per ulteriori informazioni, consulta [Introspezione del container di Amazon ECS](#).
- L'endpoint dei metadati dei processi per l'opzione di calcolo:
 - [Versione 4 degli endpoint dei metadati delle attività di Amazon ECS](#)
 - [Endpoint metadati delle attività Amazon ECS versione 4 per le attività su Fargate](#)

Considerazioni

Quando utilizzi il Monitoraggio del runtime, tieni presenti le considerazioni che seguono:

- Il Monitoraggio del runtime 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 Monitoraggio del runtime a livello di cluster deve disporre delle autorizzazioni IAM adeguate 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.
- I processi Fargate devono utilizzare un ruolo per l'esecuzione dell'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.
- Il Monitoraggio del runtime non è supportato per le applicazioni in esecuzione sulle istanze gestite Amazon ECS.

Utilizzo delle risorse

Il tag che aggiungi al cluster viene conteggiato ai fini della quota di tag del cluster.

Il contenitore GuardDuty secondario dell'agente 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 Fargate di Amazon ECS

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.

L'implementazione di un nuovo servizio avviene la prima volta che si avvia il servizio o quando si aggiorna quello esistente con l'opzione Forza nuova implementazione.

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 indicati i prerequisiti per l'utilizzo di Monitoraggio del runtime:

- La versione della piattaforma Fargate deve essere 1.4.0 o successive per Linux.
- Ruoli IAM e autorizzazioni per Amazon ECS:
 - I processi Fargate devono utilizzare un ruolo per l'esecuzione dell'attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per tuo conto. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
 - Puoi controllare il Monitoraggio del runtime 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 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 Network Address Translation (NAT) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

Per ulteriori informazioni, consulta [Sottorete 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 il Monitoraggio del runtime 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 di container: EC2

- È necessario utilizzare la versione 20230929 o successive dell'AMI Amazon ECS.
- È necessario eseguire l'agente Amazon ECS nella versione 1.77 o successive sulle istanze di container.
- È necessario utilizzare la versione 5.10 o successive.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta [Quali modelli operativi e carichi di lavoro supporta Runtime Monitoring](#). GuardDuty
- Puoi utilizzare Systems Manager per gestire le istanze di 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 Fargate esistenti di Amazon ECS

Quando attivi il Monitoraggio del runtime, tutte le nuove attività autonome e le implementazioni di nuovi servizi nel cluster vengono protette automaticamente. Per 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. I processi Fargate devono utilizzare un ruolo per l'esecuzione dell'attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per tuo conto. Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Procedura

- Per proteggere immediatamente un'attività, devi eseguire una delle operazioni riportate di seguito:
 - Per le attività autonome, interrompile e riavviale. Per ulteriori informazioni, consulta [Interruzione di un'attività di 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 nuova implementazione". Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).

Rimozione di 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.

Per preservare il vincolo di immutabilità, le attività e le implementazioni esistenti con Monitoraggio del runtime non vengono influenzate.

- Interrompe la fatturazione e non accettare più eventi di runtime per le attività.

Procedura

Esegui i seguenti passaggi per rimuovere il Monitoraggio del runtime 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 con la CLI o l'API](#). Utilizza 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 un endpoint VPC, consulta [Eliminazione di un endpoint di interfaccia](#) nella Guida per l'utente di AWS PrivateLink .

Rimozione di Monitoraggio del runtime per Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disabilita 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 accettare più eventi di runtime 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.

Effettua il provisioning dei cluster per il Monitoraggio del runtime aggiungendo un tag predefinito.

Per le istanze di EC2 container, si scarica, si installa e si gestisce il security agent. GuardDuty

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 indicati i prerequisiti per l'utilizzo di Monitoraggio del runtime:

- La versione della piattaforma Fargate deve essere 1.4.0 o successive per Linux.
- Ruoli IAM e autorizzazioni per Amazon ECS:
 - I processi Fargate devono utilizzare un ruolo per l'esecuzione dell'attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il security agent per tuo conto. GuardDuty Per ulteriori informazioni, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).
 - Puoi controllare il Monitoraggio del runtime 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 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 Network Address Translation (NAT) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

Per ulteriori informazioni, consulta [Sottorete 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 il Monitoraggio del runtime 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 di container: EC2

- È necessario utilizzare la versione 20230929 o successive dell'AMI Amazon ECS.
- È necessario eseguire l'agente Amazon ECS nella versione 1.77 o successive sulle istanze di container.
- È necessario utilizzare la versione 5.10 o successive.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta [Quali modelli operativi e carichi di lavoro supporta Runtime Monitoring](#). GuardDuty
- Puoi utilizzare Systems Manager per gestire le istanze di 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.

Aggiunta di Monitoraggio del runtime in 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 Monitoraggio del runtime 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 il Monitoraggio del runtime 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 con la CLI o l'API](#). Utilizza 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 Amazon ECS esistenti

Quando attivi il Monitoraggio del runtime, tutte le nuove attività autonome e le implementazioni di nuovi servizi nel cluster vengono protette automaticamente. Per 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, devi eseguire una delle operazioni riportate di seguito:

- Per le attività autonome, interrompile e riavviate. Per ulteriori informazioni, consultare [Interruzione di un'attività di 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 nuova implementazione". Per ulteriori informazioni, consulta [Aggiornamento di un servizio Amazon ECS](#).

Rimozione di Monitoraggio del runtime da un cluster Amazon ECS

Puoi rimuovere il Monitoraggio del runtime da un cluster. Ciò GuardDuty causa l'interruzione del monitoraggio di tutte le risorse nel cluster.

Per rimuovere il Monitoraggio del runtime 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 con 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'utente di GuardDuty.

3. Elimina l'endpoint GuardDuty VPC per ogni VPC del cluster. Per ulteriori informazioni su come eliminare un endpoint VPC, consulta [Eliminazione di un endpoint di interfaccia](#) nella Guida per l'utente di 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.

Rimozione di Monitoraggio del runtime per Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disabilita la funzionalità in GuardDuty. Per informazioni su come disabilitare la funzionalità, consulta [Enabling Runtime Monitoring](#) nella Amazon GuardDuty User Guide.

Rimuovi il Monitoraggio del runtime da tutti i cluster. Per ulteriori informazioni, consulta [Rimozione di 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 Monitoraggio del runtime sia abilitato e in esecuzione sulle attività e sui container.

Argomenti

- [Come posso sapere se Monitoraggio del runtime è attivo sul mio account?](#)
- [Come posso sapere se Monitoraggio del runtime 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 di EC2 container?](#)
- [Cosa succede quando non esiste un ruolo di esecuzione dell'attività per un'attività in esecuzione sul cluster?](#)
- [Come posso sapere se dispongo delle autorizzazioni corrette per etichettare i cluster per il Monitoraggio del runtime?](#)
- [Cosa succede in assenza di connessione Amazon ECR?](#)
- [Come posso risolvere gli errori relativi alla memoria nelle mie attività di Fargate dopo aver abilitato il Monitoraggio del runtime?](#)

Come posso sapere se Monitoraggio del runtime è attivo sul mio account?

Nella console Amazon ECS, le informazioni si trovano nella pagina Impostazioni dell'account.

Inoltre, puoi eseguire `list-account-settings` con l'opzione `effective-settings`.

```
aws ecs list-account-settings --effective-settings
```

Output

Le impostazioni con nome su `guardDutyActivate` e valore su `on` indicano che l'account è configurato. Devi verificare con il tuo GuardDuty 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 Monitoraggio del runtime è attivo su un cluster?

Puoi rivedere le statistiche di copertura nella GuardDuty console. Queste statistiche includono informazioni sulle risorse di Amazon ECS associate ai tuoi account o ai tuoi account membro e consistono nella percentuale dei cluster integri rispetto a tutti i cluster nella Regione AWS selezionata. Ciò include la copertura per i cluster che utilizzano EC2s Fargate e. 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 Container nella pagina dei Dettagli dell'attività.

Puoi eseguire `describe-tasks` e cercare il container 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 container denominato `gd-agent` si trova nello stato `RUNNING`.

```
"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 di EC2 container?

Esegui il seguente comando per visualizzare lo stato:

```
sudo systemctl status amazon-guardduty-agent
```

Il file di log 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 sul 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 Monitoraggio del runtime?

Per etichettare un cluster, è necessario disporre dell'operazione `ecs:TagResource` per `CreateCluster` e `UpdateCluster`.

Il seguente frammento è un esempio di policy:

```
{
  "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 relativi alla memoria nelle mie attività di Fargate dopo aver abilitato il Monitoraggio del runtime?

L'agente GuardDuty di sicurezza 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 GuardDuty le distribuzioni nel cluster. Questi strumenti ti aiutano a scoprire il profilo di consumo del GuardDuty security agent per le tue applicazioni. Dopodiché, è possibile modificare le

dimensioni dell'attività Fargate, se necessario, per evitare potenziali condizioni di esaurimento della memoria.

Monitora i container 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, è possibile utilizzarlo per ottenere un accesso ininterrotto ai container per eseguire il debug dei 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 i dettagli sull'uso di ECS Exec e per una guida video sull'utilizzo della AWS CLI di Copilot, consulta la documentazione di Copilot. GitHub](#)

È inoltre possibile utilizzare ECS Exec per mantenere policy di controllo degli accessi più severe. 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, è possibile utilizzare ECS Exec per visualizzare quali attività sono state eseguite e per verificare chi ha effettuato l'accesso a un contenitore. CloudTrail

Considerazioni

Considera le informazioni seguenti durante l'utilizzo di ECS Exec:

- ECS Exec potrebbe non funzionare come previsto quando viene eseguito su sistemi operativi non supportati da Systems Manager. Per ulteriori informazioni sui sistemi operativi supportati, consulta [Tipi di sistemi operativi](#) nella Guida per l'utente di AWS Systems Manager .
- ECS Exec è supportato per le attività eseguite sulla seguente infrastruttura:
 - Contenitori Linux su Amazon EC2 su qualsiasi AML ottimizzata per Amazon ECS, incluso Bottlerocket
 - Container 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 ambiente `NO_PROXY` 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 ambiente `NO_PROXY`, possono verificarsi errori durante il recupero dei metadati dell'istanza o delle credenziali del ruolo IAM dall'endpoint dei metadati all'interno del container. L'impostazione della variabile di ambiente `NO_PROXY` come consigliato filtra i metadati e il traffico IAM in modo che le richieste a `169.254.169.254` and `169.254.170.2` non passino attraverso il proxy 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 l'interfaccia Amazon VPC con Amazon ECS e la utilizzi AWS KMS key per la crittografia, devi creare l'interfaccia per cui l'endpoint Amazon VPC. 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 puoi utilizzare un gateway NAT), devi creare gli endpoint dell'interfaccia Amazon VPC per Gestione sessione di Systems 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](#)
- Amazon ECS Exec non è supportato per le attività eseguite in una configurazione di IPv6 sola configurazione. Per ulteriori informazioni sull'esecuzione di attività in una configurazione IPv6 - only, consulta e. [Opzioni di rete di attività di Amazon ECS per Fargate](#) [Opzioni di task networking di Amazon ECS per EC2](#)

- 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 che il file system del container possa essere scritto 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 container sidecar. Pertanto, è necessario specificare il nome del container 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 di CPU e 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 AWS CLI la 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 namespace dell'ID processo (PID). Se [condividi un namespace 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 utilizza `run-task` per avviare un'attività su un cluster che utilizza la scalabilità gestita con collocamento asincrono (avviare un'attività senza istanza).

- Non è possibile eseguire ECS Exec su container Nano Server di Microsoft.

Architecture

ECS Exec utilizza AWS Systems Manager (SSM) Session Manager per stabilire una connessione con il contenitore in esecuzione e utilizza le politiche 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).

Configurazione di ECS Exec

Per utilizzare ECS Exec, devi prima attivare la funzionalità per le tue attività e i tuoi servizi e poi eseguire i comandi nei container.

Modifiche facoltative alla definizione di attività

Se imposti il parametro di definizione dell'attività da `initProcessEnabled` a `true`, questo avvia il processo di `init` all'interno del container. In questo modo, vengono rimossi 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

È possibile attivare la funzionalità ECS Exec per i servizi e le attività autonome specificando il `--enable-execute-command` flag quando si utilizza uno dei seguenti AWS CLI comandi: [create-service](#), o. [update-service](#) [start-task](#) [run-task](#)

Ad esempio, se esegui il seguente comando, 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=ENABLED}" \
  --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

Registrazione tramite ECS Exec

Puoi configurare la registrazione per le sessioni ECS Exec per acquisire i comandi e il relativo output per scopi di controllo e risoluzione dei problemi.

Attivazione della registrazione nelle attività e nei 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 utilizzando ECS Exec. L'impostazione predefinita prevede l'invio dei log ai CloudWatch registri utilizzando il driver di awslogs registro 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 log.
- **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:us-east-1:111122223333: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](#).

È necessario aggiungere la seguente policy in linea al ruolo IAM del processo che richiede le autorizzazioni AWS KMS . Per ulteriori informazioni, consulta [Autorizzazioni ACS Exec](#).

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": "arn:aws:kms:us-east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    }
  ]
}
```

```
}
```

Utilizzo delle policy IAM per limitare l'accesso a ECS Exec

Puoi limitare l'accesso utente all'operazione API `execute-command` utilizzando una o più delle seguenti chiavi di condizione delle 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`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:ExecuteCommand",
        "ecs:DescribeTasks"
      ],
      "Resource": [
        "arn:aws:ecs:us-east-1:111122223333:task/cluster-name/*",
        "arn:aws:ecs:us-east-1:111122223333: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`.

JSON

```

{
  "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.

JSON

```

{
  "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*.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ssm:StartSession",
      "Resource": [
        "arn:aws:ecs:us-east-1:111122223333: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`.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ssm:StartSession",
      "Resource": "arn:aws:ecs:*:*:task/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Task-Tag-Key": "Exec-Task"
        }
      }
    }
  ]
}
```

Risoluzione dei problemi relativi a ECS Exec

Per ulteriore supporto nella risoluzione dei problemi, consulta [Risoluzione dei problemi con Exec](#).

Esecuzione di comandi tramite ECS Exec

Puoi utilizzare Amazon ECS Exec per raccogliere informazioni diagnostiche relative ai tuoi container e risolvere gli errori che si verificano durante l'intero ciclo di vita dei container.

Prerequisiti

Prima di iniziare a utilizzare ECS Exec, assicurati di aver completato le seguenti operazioni:

- Rivedi le considerazioni. Per ulteriori informazioni, consulta [Considerazioni](#)
- Attiva ECS Exec per attività e servizi. Per ulteriori informazioni, consulta [Configurazione di ECS Exec](#)
- Installa e configura il. AWS CLI Per ulteriori informazioni, consulta [Nozioni di base su AWS CLI](#).
- Installa il plug-in Session Manager per AWS CLI. Per ulteriori informazioni, consulta [Installazione del plug-in Session Manager per AWS CLI](#).
- Configura un ruolo dell'attività con le autorizzazioni appropriate. Devi utilizzare un ruolo attività con le autorizzazioni appropriate per ECS Exec. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività](#).
- Verifica i requisiti della versione. 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](#).

Utilizzo della console per le attività del servizio

Puoi utilizzare la console per eseguire i comandi utilizzando ECS Exec.

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.

3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. Nella pagina dei dettagli del servizio, scegli Elimina. Dopodiché, seleziona il ruolo.
5. In Container, scegli il container in cui desideri utilizzare ECS Exec.
6. Per eseguire i comandi, segui uno di questi passaggi:

- Scegli Connetti.

Viene visualizzata una CloudShell sessione in cui è possibile eseguire i comandi.

- Seleziona la freccia, quindi scegli il comando Copia AWS CLI .

In questo modo, è possibile eseguire i comandi localmente.

Risultato previsto

Se la connessione è riuscita, dovresti vedere un prompt interattivo della shell dal tuo container. Ora puoi eseguire i comandi direttamente nell'ambiente del container. Per terminare la sessione, scegli Interrompi sessione.

Utilizzo della console per processi autonomi

Puoi utilizzare la console per eseguire i comandi utilizzando ECS Exec.

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Attività, seleziona l'attività.

Si apre la pagina dei dettagli del processo.

4. In Container, scegli il container in cui desideri utilizzare ECS Exec.
5. Per eseguire i comandi, segui uno di questi passaggi:

- Scegli Connetti.

Viene visualizzata una CloudShell sessione in cui è possibile eseguire i comandi.

- Seleziona la freccia, quindi scegli il comando Copia AWS CLI .

In questo modo, è possibile eseguire i comandi localmente.

Risultato previsto

Se la connessione è riuscita, dovresti vedere un prompt interattivo della shell dal tuo container. Ora puoi eseguire i comandi direttamente nell'ambiente del container. Per terminare la sessione, scegli **Interrompi sessione**.

Utilizzo della shell di comando

Puoi utilizzare la shell di comando per eseguire i comandi utilizzando 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"
```

Risultato previsto

Se il comando è riuscito, dovresti vedere un prompt interattivo della shell dal tuo container. Ora puoi eseguire i comandi direttamente nell'ambiente del container. Per uscire dalla sessione, digita `exit` o premi `Ctrl+D`.

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 .

Server MCP Amazon ECS

Il server Amazon ECS MCP è un servizio completamente gestito che consente esperienze basate sull'intelligenza artificiale per lo sviluppo e le operazioni. [Model Context Protocol \(MCP\)](#) fornisce un'interfaccia standardizzata che arricchisce gli agenti e le applicazioni AI con una conoscenza contestuale e in tempo reale dei carichi di lavoro Amazon ECS, consentendo risposte più accurate e sensibili al contesto e flussi di lavoro basati sull'intelligenza artificiale per ispezionare, risolvere i problemi e ottimizzare i carichi di lavoro Amazon ECS.

Note

Il server Amazon ECS MCP è in versione di anteprima ed è soggetto a modifiche.

Panoramica

Il server Amazon ECS MCP può essere facilmente integrato con qualsiasi assistente di codifica AI compatibile con MCP, come Kiro, o strumenti di terze parti come Claude Code, Cursor e Cline, per migliorare il flusso di lavoro di sviluppo.

Durante l'implementazione, fornisce visibilità in tempo reale sulle implementazioni dei servizi e sullo stato di salute delle attività. Per lo sviluppo e le operazioni, il server semplifica la gestione di Amazon ECS fornendo flussi di lavoro di alto livello per il monitoraggio dello stato dei container, l'analisi degli eventi di servizio e l'analisi dei problemi di distribuzione. Per il debug e la risoluzione dei problemi, il server accelera la risoluzione dei problemi attraverso strumenti diagnostici completi che analizzano gli errori delle attività, i log dei container e le configurazioni di rete. Queste funzionalità sono accessibili tramite interazioni in linguaggio naturale, che consentono di eseguire operazioni complesse sui container in modo più intuitivo ed efficiente.

Il server Amazon ECS MCP offre diversi strumenti che puoi utilizzare per:

- Monitora distribuzioni e servizi

Controlla lo stato dell'implementazione, monitora gli eventi di servizio e monitora lo stato dei container in tempo reale.

- Risolvi i problemi relativi ai container

Analizza gli errori delle attività, analizza gli errori di estrazione delle immagini ed esamina i log dei contenitori con finestre temporali personalizzabili.

- Ispeziona le configurazioni di rete

Esamina le configurazioni di VPC, sottorete e gruppi di sicurezza per i tuoi servizi Amazon ECS.

- Risolvi le dipendenze tra le risorse

Identifica i blocchi che impediscono l'eliminazione delle definizioni delle attività e comprendi le relazioni tra le risorse.

Il server MCP Amazon ECS completamente gestito è ospitato nel AWS cloud, eliminando la necessità di installazione e manutenzione locali. Fornisce funzionalità di livello aziendale come aggiornamenti e patch automatici, sicurezza centralizzata tramite l'integrazione IAM, registrazione completa degli audit e la comprovata scalabilità CloudTrail, affidabilità e supporto di AWS.

Nozioni di base

Per iniziare, consulta [Guida introduttiva al server MCP Amazon ECS](#).

Guida introduttiva al server MCP Amazon ECS

Questa guida illustra i passaggi per configurare e utilizzare il server MCP Amazon ECS con i tuoi assistenti di codice AI. Imparerai come configurare il tuo ambiente, connetterti al server e iniziare a gestire i cluster Amazon ECS attraverso interazioni in linguaggio naturale.

Note

Il server Amazon ECS MCP è in versione di anteprima ed è soggetto a modifiche.

Prerequisiti

Prima di iniziare, assicurati di disporre di:

- [Creato un AWS account con accesso ad Amazon ECS](#)
- [Installato e configurato AWS CLI con credenziali](#)
- [Python 3.10+ installato](#)

- [Uv installato](#)

Configurazione

Verifica dei prerequisiti di

Verifica che la tua versione di Python sia 3.10 o superiore

```
python3 --version
```

Controlla l'installazione di uv

```
uv --version
```

Verifica la AWS CLI configurazione

```
aws configure list
```

Configura le autorizzazioni IAM

Avrai bisogno delle autorizzazioni [IAM](#) per effettuare richieste di sola lettura ai AWS servizi e interagire con il server MCP. Puoi sfruttare [le policy AWS gestite o una singola policy](#) personalizzata per un controllo più granulare.

Opzione 1: policy combinata gestita e personalizzata

1. Allega la policy AWS gestita ReadOnlyAccessper l'accesso in sola lettura a tutti i servizi AWS
2. Crea e allega una politica personalizzata aggiuntiva per le autorizzazioni MCP (vedi le autorizzazioni MCP JSON di seguito)

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MCPServerAccess",
      "Effect": "Allow",
      "Action": [
        "ecs-mcp:InvokeReadOnlyTools",
        "ecs-mcp:UseMcp"
      ]
    }
  ]
}
```

```

    ],
    "Resource": "*"
  }
]
}

```

Opzione 2: policy personalizzata singola (controllo più granulare)

In alternativa, puoi creare e allegare un'unica policy JSON personalizzata che includa sia le autorizzazioni di AWS servizio che le autorizzazioni MCP:

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "MCPServerAccess",
      "Effect": "Allow",
      "Action": [
        "ecs-mcp:InvokeReadOnlyTools",
        "ecs-mcp:UseMcp"
      ],
      "Resource": "*"
    },
    {
      "Sid": "ECSReadOnlyAccess",
      "Effect": "Allow",
      "Action": [
        "ecs:ListAccountSettings",
        "ecs:ListClusters",
        "ecs:ListContainerInstances",
        "ecs:ListTaskDefinitions",
        "ecs:ListServices",
        "ecs:ListServiceDeployments",
        "ecs:ListTasks",
        "ecs:DescribeClusters",
        "ecs:DescribeCapacityProviders",
        "ecs:DescribeContainerInstances",
        "ecs:DescribeTaskDefinition",
        "ecs:DescribeServices",
        "ecs:DescribeServiceDeployments",
        "ecs:DescribeServiceRevisions",
        "ecs:DescribeTaskSets",
        "ecs:DescribeTasks"
      ]
    }
  ]
}

```

```
    ],
    "Resource": "*"
  },
  {
    "Sid": "CloudWatchLogsReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "logs:DescribeLogGroups",
      "logs:DescribeLogStreams",
      "logs:GetLogEvents",
      "logs:FilterLogEvents"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ELBReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "elasticloadbalancing:DescribeLoadBalancers",
      "elasticloadbalancing:DescribeTargetGroups",
      "elasticloadbalancing:DescribeListeners"
    ],
    "Resource": "*"
  },
  {
    "Sid": "EC2ReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "ec2:DescribeVpcs",
      "ec2:DescribeSubnets",
      "ec2:DescribeSecurityGroups",
      "ec2:DescribeRouteTables",
      "ec2:DescribeNetworkInterfaces"
    ],
    "Resource": "*"
  },
  {
    "Sid": "ECRReadOnlyAccess",
    "Effect": "Allow",
    "Action": [
      "ecr:DescribeRepositories",
      "ecr:DescribeImages"
    ],
    "Resource": "*"
  }
```



```
}  
]  
}
```

Scegli e configura un assistente AI

Installa assistenti AI compatibili con MCP o qualsiasi strumento compatibile con MCP. [Ad esempio, puoi usare Kiro, Cline, Cursor o Claude Code](#). Quindi configura il tuo assistente di codice AI per utilizzare il server Amazon ECS MCP tramite MCP Proxy for AWS, necessario per l'accesso sicuro e autenticato al server Amazon ECS MCP. Il proxy funge da bridge lato client, gestendo l'autenticazione SigV4 utilizzando le credenziali locali. AWS AWS L'esempio seguente utilizza Kiro CLI. Segui questo [link](#) per saperne di più sulla configurazione di MCP in Kiro.

Individua il file di configurazione MCP

- macOS/Linux:

```
~/.kiro/settings/mcp.json
```

- Windows:

```
%USERPROFILE%\kiro\settings\mcp.json
```

Crea il file di configurazione se non esiste.

Aggiungi la configurazione del server MCP

Assicuratevi di sostituire il segnaposto region (`{region}`) con la regione desiderata (ad es.). `us-west-2` Fate riferimento a [the section called “Contenitori Linux su AWS Fargate”](#) per un elenco completo delle regioni. Assicuratevi inoltre di sostituire il `{profile}` segnaposto con il [nome del tuo AWS CLI profilo](#), ad es. `default`

Per Mac/Linux:

```
{  
  "mcpServers": {  
    "ecs-mcp": {  
      "disabled": false,  
      "type": "stdio",
```

```

    "command": "uvx",
    "args": [
      "mcp-proxy-for-aws@latest",
      "https://ecs-mcp.{region}.api.aws/mcp",
      "--service",
      "ecs-mcp",
      "--profile",
      "{profile}",
      "--region",
      "{region}"
    ]
  }
}

```

Per Windows:

```

{
  "mcpServers": {
    "ecs-mcp": {
      "disabled": false,
      "type": "stdio",
      "command": "uvx",
      "args": [
        "--from",
        "mcp-proxy-for-aws@latest",
        "mcp-proxy-for-aws.exe",
        "https://ecs-mcp.{region}.api.aws/mcp",
        "--service",
        "ecs-mcp",
        "--profile",
        "{profile}",
        "--region",
        "{region}"
      ]
    }
  }
}

```

Verifica la configurazione

Riavvia Kiro CLI `kiro-cli`, verifica che il server MCP sia caricato e controlla gli `/mcp` strumenti disponibili. `/tools`

Verifica la tua configurazione

Prova la connessione

Fai una semplice domanda al tuo assistente AI per verificare la connessione:

```
List all ECS clusters in my AWS account
```

Dovresti vedere un elenco dei tuoi cluster Amazon ECS.

Parla con il tuo assistente AI che utilizza il server Amazon ECS MCP

Esempio 1: monitoraggio delle distribuzioni

```
Check deployment status for my web-service in production-cluster  
Show me the ALB URL for my deployed service  
Get service events for the last hour
```

Esempio 2: analisi dello stato dei container

```
Show me all tasks that failed in the last 2 hours  
Why are my containers failing health checks?  
Display container logs for my api-service
```

Esempio 3: risoluzione dei problemi

```
Analyze task failures in my production cluster  
Check for image pull errors in the last 30 minutes  
Why is my task definition stuck in DELETE_IN_PROGRESS state?
```

Esempio 4: ispezionare le configurazioni

```
Show me the network configuration for my web-service  
What security groups are attached to my service?  
List all VPC and subnet details for my ECS service
```

Configurazioni e best practice comuni

Profili multipli AWS

Se lavori con più AWS account, crea configurazioni server MCP separate.

Per Mac/Linux:

```
{
  "mcpServers": {
    "ecs-mcp-prod": {
      "disabled": false,
      "type": "stdio",
      "command": "uvx",
      "args": [
        "mcp-proxy-for-aws@latest",
        "https://ecs-mcp.{region}.api.aws/mcp",
        "--service",
        "ecs-mcp",
        "--profile",
        "production",
        "--region",
        "us-west-2"
      ]
    },
    "ecs-mcp-dev": {
      "disabled": false,
      "type": "stdio",
      "command": "uvx",
      "args": [
        "mcp-proxy-for-aws@latest",
        "https://ecs-mcp.{region}.api.aws/mcp",
        "--service",
        "ecs-mcp",
        "--profile",
        "development",
        "--region",
        "us-east-1"
      ]
    }
  }
}
```

Best practice di sicurezza

Non passate informazioni segrete o sensibili tramite meccanismi di input consentiti:

- Non includete segreti o credenziali in alcun file di configurazione
- Non passate informazioni sensibili direttamente nei prompt al modello

- Non includete segreti nelle definizioni delle attività o nelle configurazioni dei servizi
- Evita di registrare informazioni riservate nei registri delle applicazioni
- Utilizzate il nostro Parameter Store per archiviare informazioni sensibili

Configurazioni degli strumenti

Per un elenco completo degli strumenti e delle configurazioni, vedere. [Configurazioni degli strumenti del server Amazon ECS MCP](#)

Configurazioni degli strumenti del server Amazon ECS MCP

Questa guida mostra tutte le configurazioni disponibili per il server MCP Amazon ECS completamente gestito.

Note

Il server Amazon ECS MCP è in versione di anteprima ed è soggetto a modifiche.

Variabili di ambiente

AWS_PROFILE (facoltativo)

AWS nome del profilo delle credenziali da utilizzare; può essere sovrascritto dall'argomento della riga di comando. `--profile`

Ad esempio: `export AWS_PROFILE=dev`

AWS_REGION (facoltativo)

AWS regione per la firma di SigV4; il valore predefinito è se non è impostato. `us-west-2`

Ad esempio: `export AWS_REGION=us-west-2`

Arguments (Argomenti)

URL dell'endpoint MCP SigV4 (richiesto)

L'URL dell'endpoint MCP a cui connettersi.

Ad esempio: `https://ecs-mcp.us-west-2.api.aws/mcp`

`--service` (opzionale)

AWS nome del servizio per la firma SigV4; rilevato automaticamente dal nome host dell'endpoint se non fornito.

Ad esempio: `--service ecs-mcp`

`--profile` (opzionale)

AWS profilo di credenziali da usare. Il valore predefinito è la variabile di `AWS_PROFILE` ambiente se non specificata.

Ad esempio: `--profile dev`

`--region`

AWS regione da usare. Utilizza la variabile di `AWS_REGION` ambiente se non è impostata, il valore predefinito è. `us-east-1`

Ad esempio: `--region us-west-2`

`--read-only` (opzionale)

Disabilita gli strumenti che potrebbero richiedere permessi di scrittura (gli strumenti che NON richiedono permessi di scrittura sono annotati con). `readOnlyHint=true` Per impostazione predefinita, tutti gli strumenti sono abilitati.

Ad esempio: `--read-only`

Per ulteriori opzioni di configurazione, vedere [Parametri di configurazione](#).

Tools (Strumenti)

[Il server Amazon ECS MCP espone i seguenti strumenti MCP.](#)

Strumenti di sola lettura

`get_deployment_status`

Lo strumento verifica lo stato della distribuzione di Amazon ECS per un determinato cluster e servizio Amazon ECS.

- Azioni IAM richieste:
 - `ecs:DescribeServices`
 - `elasticloadbalancing:DescribeTargetGroups`
 - `elasticloadbalancing:DescribeLoadBalancers`
- Parametri obbligatori:
 - `cluster_name(stringa)`: nome del cluster Amazon ECS
 - `service_name(stringa)`: nome del servizio Amazon ECS
- Risposta: lo strumento restituisce lo stato della distribuzione e i dettagli associati alla distribuzione.

`fetch_service_events`

Questo strumento recupera gli eventi del servizio Amazon ECS per la diagnostica con finestre temporali personalizzabili.

- Azioni IAM richieste:
 - `ecs:DescribeServices`
- Parametri obbligatori:
 - `cluster_name(stringa)`: nome del cluster Amazon ECS
 - `service_name(stringa)`: nome del servizio Amazon ECS
- Parametri facoltativi:
 - `time_window(numero intero)`: puoi indicare una finestra temporale in secondi (l'impostazione predefinita è 1 ora o 3600 secondi)
 - `start_time(stringa)`: ora di inizio personalizzata in formato ISO
 - `end_time(string)`: Ora di fine personalizzata in formato ISO
- Risposta: Lo strumento restituisce l'elenco degli eventi per il servizio con un riepilogo.

`fetch_task_failures`

Questo strumento ti aiuta a recuperare e analizzare gli errori delle attività di Amazon ECS con riepiloghi. Utilizzando questo strumento, puoi identificare qualsiasi modello di fallimento delle attività.

- Azioni IAM richieste:
 - `ecs:ListTasks`
 - `ecs:DescribeTasks`
- Parametri obbligatori:

- `cluster_name(stringa)`: nome del cluster Amazon ECS in cui risiede l'attività
- Parametri facoltativi:
 - `service_name(stringa)`: puoi filtrare per servizio specifico
 - `time_window(numero intero)`: Puoi indicare una finestra temporale in secondi (l'impostazione predefinita è 1 ora o 3600 secondi)
 - `start_time(stringa)`: ora di inizio personalizzata in formato ISO
 - `end_time(string)`: Ora di fine personalizzata in formato ISO
- Risposta: Lo strumento restituisce `TaskFailureResult` le attività non riuscite e il riepilogo

`fetch_task_logs`

Questo strumento recupera CloudWatch i log per le attività di Amazon ECS con opzioni di intervallo di tempo flessibili. Questo strumento viene utilizzato per risolvere i problemi di runtime.

- Azioni IAM richieste:
 - `ecs:DescribeServices`
 - `ecs:DescribeTaskDefinition`
 - `logs:FilterLogEvents`
- Parametri obbligatori:
 - `cluster_name(stringa)`: nome del cluster Amazon ECS
 - `service_name(stringa)`: nome del servizio Amazon ECS
- Parametri facoltativi:
 - `time_window(numero intero)`: puoi indicare una finestra temporale in secondi (l'impostazione predefinita è 1 ora o 3600 secondi)
 - `start_time(stringa)`: ora di inizio personalizzata in formato ISO
 - `end_time(string)`: Ora di fine personalizzata in formato ISO
 - `log_group_name(stringa)`: Specificare il gruppo di CloudWatch log
- Risposta: lo strumento restituisce `TaskLogsResult` le voci di registro

`get_task_definition_deletion_blockers`

Questo strumento identifica le dipendenze che possono impedire l'eliminazione delle definizioni delle attività. Lo strumento è adatto quando si eseguono operazioni di pulizia e si capisce cosa impedisce l'eliminazione delle risorse di Amazon ECS.

- Azioni IAM richieste:

- `ecs:ListClusters`
- `ecs:ListServices`
- `ecs:DescribeServices`
- `ecs:ListTasks`
- `ecs:DescribeTasks`
- Parametri obbligatori:
 - `task_definition_arn(stringa)`: definizione dell'attività Amazon Resource Name (ARN) da analizzare
- Risposta: lo strumento restituisce i blocchi e lo stato di eliminazione

`detect_image_pull_failures`

Questo strumento consente di rilevare e classificare gli errori di pull delle immagini dei contenitori.

- Azioni IAM richieste:
 - `ecs:ListTasks`
 - `ecs:DescribeTasks`
- Parametri obbligatori:
 - `cluster_name(stringa)`: nome del cluster Amazon ECS
- Parametri facoltativi:
 - `service_name(stringa)`: puoi filtrare per servizio specifico
 - `time_window(numero intero)`: Puoi indicare una finestra temporale in secondi (l'impostazione predefinita è 1 ora o 3600 secondi)
 - `start_time(stringa)`: ora di inizio personalizzata in formato ISO
 - `end_time(string)`: Ora di fine personalizzata in formato ISO
- Risposta: Lo strumento restituisce gli errori e il riepilogo

`fetch_network_configuration`

Questo strumento recupera i dettagli di configurazione della rete del servizio Amazon ECS. Puoi utilizzare questo strumento per comprendere la configurazione di Amazon VPC, sottorete e gruppo di sicurezza della tua configurazione Amazon ECS.

- Azioni IAM richieste:
 - `ecs:DescribeServices`
 - `ec2:DescribeSecurityGroups`

- `ec2:DescribeSubnets`
- Parametri obbligatori:
 - `cluster_name(stringa)`: nome del cluster Amazon ECS
 - `service_name(stringa)`: nome del servizio Amazon ECS
- Risposta: lo strumento restituisce un elenco di dettagli `NetworkConfigurationResult` di rete

Risoluzione dei problemi legati ad Amazon ECS

Potresti aver bisogno di risolvere problemi relativi ai bilanciatori del carico, alle attività, ai servizi o alle istanze di 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 ulteriori informazioni sulle attività interrotte, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Risolvi gli errori relativi alle attività interrotte.	Visualizza gli errori relativi alle attività interrotte in Amazon ECS	
Visualizza gli errori delle attività interrotte.	Risolvi gli errori relativi alle attività interrotte in Amazon ECS	
Rivedi i codici di errore delle attività interrotte.	Amazon ECS ha interrotto i messaggi di errore relativi alle attività	
Esamina gli errori delle attività. CannotPullContainer	CannotPullContainer errori di attività in Amazon ECS	
Visualizza le richieste relative ai ruoli IAM delle attività.	Visualizzare le richieste di ruolo IAM per le attività Amazon ECS	
Risolvi i problemi relativi all'utilizzo degli eventi delle attività.	Acquisizione di eventi Amazon ECS nella console	

Per ulteriori informazioni sugli errori di servizio, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Visualizza i messaggi relativi agli eventi di servizio.	Visualizzare i messaggi relativi agli eventi del servizio Amazon ECS	
Rivedi i messaggi relativi agli eventi di servizio.	Messaggi relativi agli eventi del servizio Amazon ECS	
Esamina i problemi relativi al bilanciatore del carico.	Risoluzione dei problemi dei bilanciatori del carico per il servizio in Amazon ECS	
Rivedi i problemi relativi al dimensionamento automatico del servizio.	Risoluzione dei problemi di dimensionamento automatico del servizio in Amazon ECS	

Per ulteriori informazioni sugli errori relativi alla definizione delle attività, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Risolvi l'errore di memoria della definizione dell'attività.	Risoluzione degli errori relativi a CPU o memoria non valide nella definizione delle attività Amazon ECS	

Per informazioni sugli errori dell'agente Amazon ECS, consulta una delle seguenti risorse.

Azione	Ulteriori informazioni	
Visualizza i log dell'agente container Amazon ECS.	Visualizzare i log dell'agente container Amazon ECS	

Azione	Ulteriori informazioni	
Scopri come raccogliere i log Amazon ECS.	Raccolta log dei container con il raccoglitore di log di Amazon ECS	
Recupera i dettagli diagnostici con l'agente Amazon ECS.	Recupera i dettagli diagnostici di Amazon ECS con l'ispezione dell'agente	

Per ulteriori informazioni sugli errori di Docker, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Usa lo strumento di diagnostica di Docker	Diagnostica Docker in Amazon ECS	
Attiva la modalità debug di Docker.	Configurazione dell'output dettagliato dal daemon Docker in Amazon ECS	
Risolvi l'errore 500 dell'API Docker.	Risolvi i problemi relativi a API error (500): devmapper di Docker in Amazon ECS	

Per informazioni sugli errori di ECS Exec e Amazon ECS Anywhere, consulta una delle seguenti risorse.

Azione	Ulteriori informazioni	
Risoluzione dei problemi relativi a ECS Exec.	Risoluzione dei problemi di Amazon ECS Exec	
Risoluzione dei problemi relativi ad Amazon ECS Anywhere.	Risoluzione dei problemi relativi ad Amazon ECS Anywhere	

Per informazioni sui problemi relativi all'associazione dei volumi Amazon EBS alle attività Amazon ECS, consulta quanto segue:

Azione	Ulteriori informazioni	
Risoluzione dei problemi relativi ai volumi di Amazon EBS collegati alle attività di Amazon ECS.	Risoluzione dei problemi relativi ai volumi di Amazon EBS collegati alle attività di Amazon ECS	
Motivi dello stato degli allegati del volume Amazon EBS alle attività di Amazon ECS.	Motivi dello stato dell'allegato del volume Amazon EBS alle attività di Amazon ECS	

Per informazioni sui problemi relativi all'utilizzo di AWS Cloud Map namespace condivisi con Amazon ECS Service Connect, consulta una delle seguenti pagine:

Azione	Ulteriori informazioni	
Risoluzione dei problemi di Amazon ECS Service Connect con AWS Cloud Map namespace condivisi.	Risoluzione dei problemi di Amazon ECS Service Connect con namespace condivisi AWS Cloud Map	

Per ulteriori informazioni sui problemi di limitazione della larghezza di banda della rete, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Scopri le 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 ulteriori informazioni sugli errori relativi alle API, consulta uno dei seguenti argomenti.

Azione	Ulteriori informazioni	
Risolvi gli errori delle API.	Motivi degli errori dell'API Amazon ECS	

Per informazioni sulla risoluzione dei problemi basata sull'intelligenza artificiale, consulta quanto segue:

Azione	Ulteriori informazioni	
Risolvi i problemi con Amazon Q Developer nella console.	Risoluzione dei problemi con Amazon Q Developer	
Risolvi i problemi con gli assistenti AI utilizzando il server MCP Amazon ECS.	Server MCP Amazon ECS	

Risolvi gli errori relativi alle attività interrotte in Amazon ECS

Quando l'attività non riesce ad avviarsi, viene visualizzato un messaggio di errore nella console e nei parametri di output `describe-tasks` (`stoppedReason` e `stopCode`).

Puoi visualizzare le attività interrotte nella console per un'ora. Per visualizzare le attività interrotte, devi modificare l'opzione di filtro. Per ulteriori informazioni, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

Le pagine seguenti forniscono informazioni sulle attività interrotte.

- Scopri le modifiche apportate ai messaggi di errore relativi alle attività interrotte.

[Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore relativi alle attività](#)

- Visualizza le attività interrotte per ottenere informazioni sulle cause.

[Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#)

- Scopri i messaggi di errore relativi alle attività interrotte e le possibili cause degli errori.

[Amazon ECS ha interrotto i messaggi di errore relativi alle attività](#)

- Scopri come verificare la connettività delle attività interrotte e correggere gli errori.

Verifica della connettività delle attività interrotte di Amazon ECS

Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore relativi alle attività

A partire dal 14 giugno 2024, il team Amazon ECS modificherà i messaggi di errore relativi alle attività interrotte come descritto nelle tabelle seguenti. Il stopCode non cambierà. Se le tue applicazioni dipendono da stringhe di messaggi di errore precise, devi aggiornare le applicazioni con le nuove stringhe. Per assistenza in caso di domande o problemi, contatta. Supporto AWS

 Note

Consigliamo di non fare affidamento sui messaggi di errore per l'automazione, in quanto tali messaggi di errore sono soggetti a modifiche.

CannotPullContainerError

Vecchio messaggio di errore	Nuovo messaggio di errore	
CannotPullContainerError: Risposta di errore dal demone: accesso pull negatorepository , il repository non esiste o potrebbe richiedere 'accesso docker': negato: Utente: roleARN	• CannotPullContainerError: L'operazione non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre le immagini dal registro. Risposta di errore dal demone: pull access negato perrepository y , il repository non esiste o potrebbe richiedere 'docker login': denied: User: roleARN is not authorized to perform: ecr: BatchGetI mage on resource: image	

Vecchio messaggio di errore	Nuovo messaggio di errore	
	perché nessuna policy basata sull'identità consente l'azione ecr:. BatchGetImage CannotPullContainerError: L'attività non può estrarre l'immagine. Verifica che l'immagine esista. 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 di 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: timeout <i>registry-uri</i> i/o	CannotPullContainerError: L'operazione non può essere estratta <i>image-uri</i> dal registro. <i>registry-uri</i> C'è un problema di connessione tra l'attività e il registro. <port>Controlla la configurazione della rete dell'operazione.: failed to copy: httpReadSeeker: failed open: failed to do request: Get <i>registry-uri</i> : dial tcp<ip>: timeout i/o	

ResourceNotFoundException

Vecchio messaggio di errore	Nuovo messaggio di errore	
Il recupero di dati segreti dalla Gestione dei segreti AWS <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. Gestione dei segreti AWS Verifica se il segreto esiste nella regione specificata. ResourceNotFoundException: Recupero di dati segreti dalla Gestione dei segreti AWS 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:: timeout. i/o Verifica la configurazione di rete dell'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: invio della richiesta non riuscito a causa di: Post "https://api.ecr.us-east-1.amazonaws.com»: dial tcp<ip>:<port>: timeout i/o	
ResourceInitializationError : impossibile recuperare segreti o autenticazione del registro: recupero delle risorse di esecuzione non riuscito: impossibile recuperare e segreti da ssm: la chiamata al 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 e segreti da AWS Systems Manager Si è verificato 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. Si è verificato 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 di 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	

Visualizza gli errori relativi alle attività interrotte in 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' Console di gestione AWS,, 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 [Visualizzare i log dell'agente container Amazon ECS](#).

Procedura

Console

Console di gestione AWS

La seguente procedura può essere utilizzata per verificare la presenza di errori nelle attività arrestate utilizzando la nuova console. Per visualizzare le attività interrotte, devi modificare l'opzione di filtro.

Le attività interrotte rimangono visibili nella console solo per 1 ora.

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 Cluster.
3. Nella pagina Cluster, scegliere il cluster.
4. Nella **name** pagina Cluster:, scegli la scheda Attività.
5. Configura il filtro per visualizzare le attività interrotte. In Filtra lo stato desiderato, seleziona 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 arrestata per recuperare le informazioni. Per ulteriori informazioni, consulta [describe-tasks](#) nella Guida di 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
```

Usa i seguenti parametri di output.

- stopCode- Il codice di interruzione indica il motivo per cui un'attività è stata interrotta, ad esempio ResourceInitializationError
- StoppedReason - Il motivo per cui l'attività è stata interrotta.

- `reason`(nella struttura `containers`) - Il motivo fornisce ulteriori dettagli sul container interrotto.

Fasi successive

Visualizza le attività interrotte per ottenere informazioni sulle cause. Per ulteriori informazioni, consulta [Amazon ECS ha interrotto i messaggi di errore relativi alle attività](#).

Amazon ECS ha interrotto i messaggi di errore relativi alle attività

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 relativo alle attività interrotte utilizzando il Console di gestione AWS, vedere [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

Tip

Puoi utilizzarlo [Server MCP Amazon ECS](#) con gli assistenti AI per analizzare gli errori delle attività e i log dei container utilizzando il linguaggio naturale.

Ai codici di errore delle attività interrotte è associata una categoria, ad esempio "». `ResourceInitializationError` Per ulteriori informazioni su ogni categoria, consulta:

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	

Categoria	Ulteriori informazioni	
SpotInterruptionError	Risoluzione degli errori di Amazon ECS SpotInterruption	
InternalError	Risoluzione degli errori di Amazon ECS InternalError	
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 di errore TaskFailedToStart e le azioni che puoi intraprendere per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il Console di gestione AWS, consulta. [Visualizza gli errori relativi alle attività interrotte in 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'attività Fargate usa la modalità di rete awsvpc, viene 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 [Visualizzazione della sottorete](#) nella Guida per l'utente di Amazon VPC.

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 scegli una definizione dell'attività con un tipo di avvio che non corrisponde al tipo di capacità del cluster. Devi selezionare una definizione dell'attività che corrisponda al provider di capacità assegnato al cluster.

Impossibile collegare l'interfaccia di rete all'indice dei dispositivi inutilizzati

Questo errore si verifica quando si utilizza un tipo awsvpc di rete e non ce n'è abbastanza per CPU/memory 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 questo 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 di errore ResourceInitialization e le azioni che puoi intraprendere per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

Errori

- [L'attività non può richiamare l'autenticazione del registro da Amazon ECR. Si è verificato un problema di connessione tra l'attività e Amazon ECR. Verifica la configurazione di rete dell'attività.](#)
- [L'attività non è in grado di scaricare i file delle variabili di ambiente da Amazon S3. Si è verificato un problema di connessione tra l'attività e Amazon S3. Verifica la configurazione di rete dell'attività.](#)
- [L'attività non è in grado di recuperare i segreti da AWS Systems Manager Parameter Store. Controlla la connessione di rete tra l'attività e AWS Systems Manager.](#)
- [L'attività non può estrarre segreti da Gestione dei segreti AWS. Si è verificato un problema di connessione tra l'attività e Secrets Manager. Verifica la configurazione di rete dell'attività.](#)
- [L'attività non può recuperare il segreto da Secrets Manager. L'operazione non può recuperare il segreto con ARN secretARN " da Secrets Manager. Verifica se il segreto esiste nella regione specificata.](#)
- [comando pull non riuscito: impossibile recuperare segreti o autenticazione del registro Verifica la configurazione di 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 di rete.](#)

- [inizializzazione del driver di registrazione non riuscita](#)
- [impossibile invocare i comandi EFS utils per configurare i volumi EFS](#)

L'attività non può richiamare l'autenticazione del registro da Amazon ECR. Si è verificato un problema di connessione tra l'attività e Amazon ECR. Verifica la configurazione di rete dell'attività.

Questo errore indica che l'attività non può connettersi ad Amazon ECR.

Verifica la connessione tra l'attività e Amazon ECR. Per informazioni, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

L'attività non è in grado di scaricare i file delle variabili di ambiente da Amazon S3. Si è verificato un problema di connessione tra l'attività e Amazon S3. Verifica la configurazione di rete dell'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, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

L'attività non è in grado di recuperare i 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 riesce a eseguire il pull dell'immagine definita nella definizione dell'attività usando le credenziali in Systems Manager.

Verifica la connessione tra l'attività e l'endpoint VPC di Systems Manager. Per informazioni, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

L'attività non può estrarre segreti da Gestione dei segreti AWS. Si è verificato un problema di connessione tra l'attività e Secrets Manager. Verifica la configurazione di rete dell'attività.

Questo errore si verifica quando l'attività non riesce a eseguire il pull dell'immagine definita nella definizione dell'attività usando le credenziali in Secrets Manager.

L'errore indica che c'è 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, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

L'attività non può recuperare il segreto da Secrets Manager. L'operazione non può recuperare il segreto con ARN ***secretARN*** " da Secrets Manager. Verifica se il segreto esiste nella regione specificata.

Questo errore si verifica quando l'attività non riesce a eseguire il pull dell'immagine definita nella definizione dell'attività usando 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. Si tratta di un problema di rete quando viene visualizzata una delle seguenti stringhe nel messaggio di errore: • dial tcp • dial udp • <ip><port>:: timeout i/o • net/http: TLS handshake timeout • read: connessione scaduta • Client.Timeout superato durante l'attesa delle intestazioni • net/http: richiesta annullata durante l'attesa della connessione • signal: killed • scadenza contesto superata	Verifica la connettività tra l'attività e l'endpoint di Secrets Manager. Per ulteriori informazioni, consulta Verifica della connettività delle attività interrotte di Amazon ECS.	
Il ruolo di esecuzione dell'attività stabilito nella definizione	Aggiungi le autorizzazioni necessarie per Secrets	

Causa dell'errore.	Esegui questa operazione...	
dell'attività non dispone delle autorizzazioni per Secrets Manager.	Manager al ruolo di esecuzione e dell'attività. Per ulteriori informazioni, consulta Autorizzazioni per Secrets Manager o Systems Manager .	
L'ARN segreto non esiste	Controlla che l'ARN esista in Secrets Manager. Per informazioni sulla visualizzazione delle immagini, consulta Trova segreti in Secrets Manager nella Guida per sviluppatori di Secrets Manager.	

comando pull non riuscito: impossibile recuperare segreti o autenticazione del registro Verifica la configurazione di rete dell'attività.

Questo errore si verifica quando l'attività non è in grado di connettersi ad Amazon ECR, Systems Manager o Secrets Manager. Ciò è dovuto a una configurazione errata della rete.

Per risolvere il problema, controlla la connettività tra l'attività e Amazon ECR. Devi inoltre verificare la connettività tra l'attività e il servizio su cui è archiviato il segreto (Systems Manager o Secrets Manager). Per ulteriori informazioni, consulta [Verifica della connettività delle attività interrotte 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 di 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 della connettività delle attività interrotte di Amazon ECS](#).

inizializzazione del driver di registrazione non riuscita

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
```

Osserva l'output di ciascun container e annota il valore `awslogs-group`.

```
"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 oppure che il gruppo di log non esiste.

3. Risolvi il problema.

Il problema è...	Esegui questa operazione...	
Nella definizione dell'attività è specificato un gruppo di log errato.	Aggiorna la definizione del task per includere la configurazione del gruppo di log nella definizione del	

Il problema è...	Esegui questa operazione...	
	container. Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS attraverso la console o RegisterTaskDefinition consulta il riferimento 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.	

impossibile invocare i comandi EFS utils per configurare i volumi EFS

I seguenti problemi potrebbero impedirti di montare i volumi Amazon EFS sulle tue attività:

- Il file system Amazon EFS non è configurato in modo corretto.
- L'attività non dispone delle autorizzazioni necessarie.
- Ci sono 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 di errore `ResourceNotFoundException` e le azioni che puoi intraprendere per correggere gli errori.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

L'operazione non può recuperare il segreto con ARN **secretARN** " da. Gestione dei segreti AWS
Verifica 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.

Nel messaggio di errore è indicata la regione.

Il recupero di dati segreti dalla Gestione dei segreti AWS regione **region**: secret: **secretARN**
ResourceNotFoundException: Secrets Manager non riesce a trovare il segreto specificato.

Per informazioni su come trovare un segreto, consulta [Scopri i segreti in Gestione dei segreti AWS](#) nella Guida per l'utente di Gestione dei segreti AWS .

Usa la seguente tabella per individuare e risolvere l'errore.

Problema	Azioni	
Il segreto si trova in una regione diversa rispetto a quella della definizione dell'attività.	- Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni, consulta Creare un Gestione dei segreti AWS segreto. - Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS attraverso la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	
La definizione dell'attività contiene un ARN segreto	Aggiorna la definizione dell'attività con il segreto corretto.	

Problema	Azioni	
errato. Il segreto corretto esiste in Secrets Manager.	Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS attraverso la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	
Il segreto non esiste più.	a. Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni, consulta Creare un Gestione dei segreti AWS segreto. b. Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta Aggiornamento di una definizione di attività Amazon ECS attraverso la console o RegisterTaskDefinition consulta l'Amazon Elastic Container Service API Reference.	

Risoluzione degli errori di Amazon ECS SpotInterruption

L'`SpotInterruption` errore ha diverse ragioni per Fargate e EC2s

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il Console di gestione AWS, vedere [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

Fargate

L'errore `SpotInterruption` si verifica quando non c'è capacità Fargate Spot o quando Fargate riprende la capacità spot.

Per aumentare la capacità, puoi eseguire le tue attività in più zone di disponibilità.

EC2

Questo errore si verifica quando non ci sono istanze Spot disponibili o EC2 ripristina la capacità dell'istanza Spot.

Per aumentare la capacità, puoi eseguire le tue istanze in più zone di disponibilità.

Risoluzione degli errori di Amazon ECS `InternalError`

Si applica a: Fargate

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il Console di gestione AWS, vedere [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

L'errore `InternalError` si verifica 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 effettuare il debug e risolvere il problema, consulta [Amazon ECS ha interrotto i messaggi di errore relativi alle attività](#).

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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

container interrotto a causa dell'utilizzo della memoria

Questo errore si verifica quando un container si chiude a causa di attività al suo interno che consumano più memoria di quella allocata nella definizione di attività oppure a causa di limitazioni dell'host o del 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

Impossibile passare all'esecuzione; timeout dopo 1 minuto di attesa 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#).

impossibile ottenere lo stato del contenitore: **<reason>**

Questo errore si verifica quando un container non può essere avviato.

Se il container tenta di superare la memoria specificata qui, viene arrestato. Aumentare la memoria presentata al container. Questo è il parametro `memory` 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in 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 la versione della piattaforma 1.3 o una versione precedente, l'agente Amazon ECS restituisce il motivo da Docker.

Quando si utilizza la versione della piattaforma 1.4.0 o successiva (Linux) oppure 1.0.0 o successiva (Windows), l'agente Fargate restituisce il codice errore 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 Console di gestione AWS, consulta [Visualizza gli errori relativi alle attività interrotte in 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 riesce a recuperare l'immagine del container 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 Console di gestione AWS, vedere. [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#)

Tip

Puoi utilizzare gli assistenti [Server MCP Amazon ECS](#) con intelligenza artificiale per analizzare gli errori di acquisizione delle immagini utilizzando il linguaggio naturale.

Errori

- [L'attività non riesce a eseguire il pull dell'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre le immagini dal registro.](#)
- [L'attività non può estrarre image-name " dal repository Amazon ECR ". repository URI Si è verificato un problema di connessione tra l'attività e Amazon ECR. Verifica la configurazione di rete dell'attività.](#)
- [L'attività non riesce a eseguire il pull dell'immagine. Controlla la configurazione di rete](#)
- [CannotPullContainerError: pull image manifest è stato riprovato 5 volte: impossibile risolvere ref](#)
- [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 oppure Hai raggiunto il limite di pull rate.](#)
- [Risposta di errore dal demone: Get: net/httpurl: richiesta annullata in attesa della connessione](#)
- [ref pull è stato riprovato 1 volta/e: copia non riuscita: apertura fallita: codice di stato imprevisto httpReaderSeeker](#)
- [accesso pull negato](#)
- [comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido oppure dereferenziazione del puntatore nulla](#)
- [errore durante l'estrazione dell'immagine durante l' conf/error estrazione della configurazione dell'immagine](#)
- [Contesto annullato](#)

L'attività non riesce a eseguire il pull dell'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre le immagini dal registro.

Questo errore indica che l'attività non è in grado di 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 [Viewing image details in Amazon ECR](#) nella Guida per l'utente di Amazon Elastic Container Registry.

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'utente di AWS Identity and Access Management

L'attività usa uno dei seguenti ruoli:

- Per le attività con Fargate, questo è il ruolo di esecuzione delle attività. Per informazioni sulle autorizzazioni aggiuntive per Amazon ECR, [Autorizzazioni per attività Fargate che eseguono il pull delle immagini Amazon ECR su endpoint di interfaccia](#).
- Per le attività con EC2, 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* Si è verificato un problema di connessione tra l'attività e Amazon ECR. Verifica la configurazione di rete dell'attività.

Questo errore indica che l'attività non può connettersi ad Amazon ECR. Verifica la connessione al repository. *repository URI*

Per informazioni su come effettuare la verifica e risolvere questi problemi, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

L'attività non riesce a eseguire il pull dell'immagine. Controlla la configurazione di rete

Questo errore indica che l'attività non può connettersi ad Amazon ECR.

Per informazioni su come effettuare la verifica e risolvere questi problemi, consulta [Verifica della connettività delle attività interrotte di Amazon ECS](#).

CannotPullContainerError: pull image manifest è stato riprovato 5 volte: impossibile risolvere ref

Questo errore indica che l'attività non è in grado di eseguire il pull dell'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 mostrato questo errore. Il tag dell'immagine fa parte dell'applicazione di questo

comportamento. Modifica l'immagine nella definizione dell'attività utilizzando una versione specifica invece del tag :latest. Per ulteriori informazioni, consulta [Risoluzione dell'immagine del container](#).

Per informazioni su come effettuare la verifica e risolvere questi problemi, consulta [Verifica della connettività delle attività interrotte 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 una route a 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 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 stai usando l'AMI ottimizzata per Amazon ECS, puoi utilizzare il comando seguente 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 container 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 [Driver di registrazione file JSON](#) sulla 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"
  }
}
```

Se i tuoi log di container utilizzano troppo spazio sul disco, l'alternativa è di utilizzare il driver di log `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](#).

Potresti dover aggiornare le dimensioni del disco a cui Docker può accedere.

Per ulteriori informazioni, consulta [CannotPullContainerError: nessuno spazio rimasto](#) sul dispositivo.

ERRORE: toomanyrequests: Troppe richieste oppure Hai raggiunto il limite di pull rate.

Questo errore indica che è in atto 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 il limite di frequenza di Docker Hub è stato aumentato ed è necessario autenticare i pull Docker per le istanze di container, consulta [Autenticazione di registri privati per istanze di container](#).

Risposta di errore dal demone: Get: net/http~~url~~: richiesta annullata in attesa della connessione

Questo errore indica che la connessione è scaduta perché non esiste una route a 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: codice di stato imprevisto
httpReaderSeeker

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](#).

accesso pull negato

Questo errore indica che l'immagine non è accessibile.

Per risolvere questo problema, potrebbe essere necessario autenticare il client Docker con Amazon ECR. Per ulteriori informazioni, consulta [Autenticazione del registro privato](#) nella Guida per l'utente di Amazon ECR.

comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido oppure dereferenziazione del puntatore nulla

Questo errore indica che non è possibile accedere all'immagine per via di un indirizzo di memoria non valido o di una dereferenziazione del puntatore nulla.

Per risolvere il problema:

- Verifica di disporre delle regole del gruppo di sicurezza per accedere ad Amazon S3.
- Quando utilizzi gli endpoint del gateway, devi aggiungere una route nella tabella di routing per accedere all'endpoint.

errore durante l'estrazione dell'immagine durante l' estrazione della configurazione dell'immagine

Questo errore indica che è stato raggiunto un limite di velocità o che si è verificato 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 della connettività delle attività interrotte di Amazon ECS

A volte un'attività si interrompe a causa di un problema di connettività di rete. Potrebbe trattarsi di un problema intermittente, ma molto probabilmente è causato dall'impossibilità dell'attività di connettersi a un endpoint.

Test della connettività delle attività

È possibile utilizzare il runbook `AWSSupport-TroubleshootECSTaskFailedToStart` per testare la connettività delle attività. Quando usi il runbook, ti servono le seguenti informazioni sulle risorse:

- L'ID dell'attività

Usa l'ID dell'ultima attività non riuscita.

- Il cluster in cui si trovava l'attività

Per informazioni su come usare il runbook, consulta [AWSSupport-TroubleshootECSTaskFailedToStart](#) nel Riferimento al runbook del servizio di automazione AWS Systems Manager .

Il runbook analizza l'attività. Puoi visualizzare i risultati nella sezione Output per i seguenti problemi che possono impedire l'avvio di un'attività:

- Connettività di rete al registro dei container configurato
- Connettività dell'endpoint VPC
- Configurazione delle regole del gruppo di sicurezza

Risoluzione dei problemi degli endpoint VPC

Quando il risultato del runbook `AWSSupport-TroubleshootECSTaskFailedToStart` indica un problema con l'endpoint VPC, controlla la seguente configurazione:

- Il VPC in cui si crea l'endpoint e l'endpoint VPC devono 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 uno dei seguenti argomenti:

Servizio	Informazioni sugli endpoint VPC per il servizio	
Amazon ECR	Endpoint VPC con interfaccia Amazon ECR (AWS PrivateLink)	

Servizio	Informazioni sugli endpoint VPC per il servizio	
Systems Manager	Migliora la sicurezza delle EC2 istanze utilizzando gli endpoint VPC per Systems Manager	
Secrets Manager	Utilizzo di un Gestione dei segreti AWS endpoint VPC	
CloudWatch	CloudWatch Endpoint VPC	
Simple Storage Service (Amazon S3)	AWS PrivateLink per Amazon S3	

- Configura una regola in uscita per la sottorete dell'attività che consenta il traffico DNS (TCP) HTTPS sulla porta 443. Per ulteriori informazioni, consulta [Configurare le regole dei gruppi di sicurezza](#) nella Guida utente di Amazon Elastic Compute Cloud..
- Se usi un server di dominio con nomi personalizzati, conferma le impostazioni della query DNS. La query deve avere accesso in uscita sulla porta 53 e usare i protocolli UDP e TCP. Inoltre, deve disporre di un accesso HTTPS sulla porta 443. Per ulteriori informazioni, consulta [Configurare le regole dei gruppi di sicurezza](#) nella Guida utente di Amazon Elastic Compute Cloud..
- Se la sottorete è dotata di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consenta il traffico sulle porte 1024-65535.
 - Una regola in entrata che consenta 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 runbook `AWSSupport-TroubleshootECSTaskFailedToStart` indica un problema di rete, controlla la seguente configurazione:

Attività che usano la modalità di rete `awsvpc` in una sottorete pubblica

Esegui la configurazione seguente 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 dell'attività deve avere una route per il traffico verso il gateway.

Per ulteriori informazioni, consultare la sezione relativa ad [Aggiungere e rimuovere route da una tabella di routing](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Tipo di gateway	Destinazione della tabella di routing	destinazione della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT
Internet Gateway	0.0.0.0/0	ID gateway Internet

- Se la sottorete dell'attività è dotata di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consenta il traffico sulle porte 1024-65535.
 - Una regola in entrata che consenta 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 usano la modalità di rete awsvpc in una sottorete privata

Esegui la configurazione seguente in base al runbook:

- Scegli **DISABLED** per assegnare automaticamente l'IP pubblico all'avvio dell'attività.
- Configura un gateway NAT nel tuo VPC per instradare le richieste a Internet. Per ulteriori informazioni, consulta [Gateway NAT](#) nella Guida per l'utente di Amazon Virtual Private Cloud.
- La tabella di routing per la sottorete dell'attività deve avere una route per il traffico verso il gateway NAT.

Per ulteriori informazioni, consultare la sezione relativa ad [Aggiungere e rimuovere route da una tabella di routing](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Tipo di gateway	Destinazione della tabella di routing	destinazione della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete dell'attività è dotata di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consenta il traffico sulle porte 1024-65535.
 - Una regola in entrata che consenta 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 usano la modalità di rete awsvpc in una sottorete pubblica

Esegui la configurazione seguente 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 una route per il traffico verso il gateway.

Per ulteriori informazioni, consultare la sezione relativa ad [Aggiungere e rimuovere route da una tabella di routing](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Tipo di gateway	Destinazione della tabella di routing	destinazione della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT
Internet Gateway	0.0.0.0/0	ID gateway Internet

- Se l'istanza della sottorete è dotata di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consenta il traffico sulle porte 1024-65535.
 - Una regola in entrata che consenta 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 usano la modalità di rete awsvpc in una sottorete privata

Esegui la configurazione seguente 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 instradare 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 una route per il traffico verso il gateway NAT.

Per ulteriori informazioni, consultare la sezione relativa ad [Aggiungere e rimuovere route da una tabella di routing](#) nella Guida per l'utente di Amazon Virtual Private Cloud.

Tipo di gateway	Destinazione della tabella di routing	destinazione della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete dell'attività è dotata di un ACL di rete, sono necessarie le seguenti regole ACL:
 - Una regola in uscita che consenta il traffico sulle porte 1024-65535.
 - Una regola in entrata che consenta 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.

Visualizzare le richieste di ruolo IAM per le attività Amazon ECS

Quando usi un provider per le credenziali dell'attività in un ruolo IAM, le richieste del provider vengono salvate in un log 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 log 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:

- Time stamp
- 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
```

Visualizzare i messaggi relativi agli eventi del servizio Amazon ECS

Se stai resolvendo 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 del servizio utilizzando l'`DescribeServices` API AWS CLI, o utilizzando il Console di gestione AWS.

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, vengono mostrati solo i 100 eventi più recenti. I messaggi relativi a eventi duplicati vengono omessi fino a che il problema non viene risolto, o se non passano sei ore. Se la causa non viene risolta entro sei ore, riceverai un altro messaggio relativo all'evento di assistenza relativo alla causa.
- Eventi di scalabilità automatica dei servizi Questi eventi hanno il prefisso Messaggio e si verificano solo quando un servizio è configurato con una policy di dimensionamento automatico delle applicazioni.

 Tip

È possibile utilizzarli [Server MCP Amazon ECS](#) con gli assistenti AI per analizzare gli eventi di servizio utilizzando il linguaggio naturale.

Utilizza la procedura seguente per visualizzare i messaggi relativi a eventi del servizio corrente.

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 Cluster.
3. Nella pagina Cluster, scegliere il cluster.
4. Scegli il servizio da ispezionare.
5. Nella scheda Eventi, 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 relativi agli 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 pianificatore del servizio invia un evento di servizio service (*service-name*) has reached a steady state. quando il servizio è integro e al numero desiderato di attività, raggiungendo così uno stato costante.

Il pianificatore del servizio segnala periodicamente lo stato, quindi potresti ricevere questo messaggio più volte.

service (*service-name*) non è stato in grado di effettuare un'attività perché nessuna istanza del contenitore soddisfaceva tutti i requisiti.

Il pianificatore del servizio invia questo messaggio di evento quando non è stato in grado di 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 [Fornitori di capacità Amazon ECS per carichi di lavoro EC2](#).

Se intendi usare un provider di capacità, assicurati di adottare una strategia per provider di capacità o di avere una strategia predefinita associata al cluster e di non adottare il tipo di lancio e la strategia del provider di capacità come input

Nessuna istanza di container trovata nel cluster

Se nessuna istanza di container è registrata nel cluster in cui tenti di eseguire un'attività, ricevi 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 container corrispondente più vicina per il posizionamento delle attività non può superare il limite massimo consentito di 100 porte host riservate per istanza di container. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione di attività di quest'attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza di container che è stata 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 [Allocazione della memoria di un'istanza di container 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 `awsvpc` 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 `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 fornito il consenso esplicito all'impostazione dell'account `awsvpcTrunking` ma non hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo il consenso esplicito, il limite ENI per ogni istanza di container non è diverso dal 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'`awsvpcTrunking`account 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 un potenziamento delle interfacce di rete dei container Amazon ECS](#).

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsvpcTrunking`, consulta [Aumento delle interfacce di rete delle istanze di container Linux di Amazon ECS](#).

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 di attività contiene un parametro che richiede un determinato attributo dell'istanza di container, ma non disponi di istanze di container in grado di soddisfare questo requisito, l'attività non potrà essere posizionata.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di awsvpc rete e. EC2 Il cluster specificato non ha un'istanza container registrata nella stessa sottorete specificata in `awsvpcConfiguration` al 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 se ci sono 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 di attività Amazon ECS per 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 di container corrispondente più vicina per il posizionamento delle attività non contiene unità CPU sufficienti per soddisfare i requisiti nella definizione di 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 stato in grado di effettuare 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 log 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**) (task) (instance **task-idinstance-id**) non è integro in (elb**lb-name**) a causa di (motivo per cui l'istanza non ha superato almeno il UnhealthyThreshold numero di controlli sanitari consecutivi).

Questo servizio è registrato con un bilanciatore del carico i cui controlli dello stato hanno esito negativo. Il messaggio include l'ID dell'attività per facilitare l'identificazione dell'attività specifica che non supera i controlli dell'integrità. Per ulteriori informazioni, consulta [Risoluzione dei problemi dei bilanciatori del carico per il servizio 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 limitazione 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 dell'implementazione è costituita dai valori `minimumHealthyPercent` e `maximumPercent` che sono definiti al momento della creazione del servizio. Tali valori possono essere aggiornati anche su un servizio esistente.

La `minimumHealthyPercent` rappresenta il limite inferiore del numero di attività che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento. 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, il numero di attività desiderato è quattro, il pianificatore può interrompere due attività esistenti prima di avviare due nuove attività. 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.

La `maximumPercent` rappresenta il limite superiore del numero di attività che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento. 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, il pianificatore può avviare quattro nuove attività prima di avviare arrestare le 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 stato in grado di effettuare un'operazione. 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 [Service Quotas](#). 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 descritta 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 inserire un'attività. 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 [Service Quotas](#). 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: 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 [Service Quotas](#). 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: 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 [Service Quotas](#). 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 maggiore del numero di attività desiderato

Il servizio ha più attività protette 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 usa l'API UpdateTaskProtection con l'opzione protectionEnabled impostata su false per annullare l'impostazione della 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 pianificatore del servizio invia questo messaggio di evento quando non è stato in grado di trovare le risorse disponibili per aggiungere un'altra attività. Le possibili cause sono:

Non esiste alcun provider di capacità associato al cluster

Usa `describe-services` per verificare che sia presente un provider di capacità associato al cluster. Puoi aggiornare la strategia del provider di capacità per il servizio.

Verifica che ci sia capacità disponibile nel provider di capacità. Nel caso di EC2, assicurati che le istanze del contenitore soddisfino i requisiti di definizione delle attività.

Nessuna istanza di container trovata nel cluster

Se nessuna istanza di container è registrata nel cluster in cui tenti di eseguire un'attività, ricevi 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. Solo un container 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 corrispondente più vicina per il posizionamento delle attività non può superare il limite massimo consentito di 100 porte host riservate per istanza di container. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione di attività di quest'attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza di container che è stata 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 [Allocazione della memoria di un'istanza di container Amazon ECS Linux](#).

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 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 fornito il consenso esplicito all'impostazione dell'account `awsVpcTrunking` ma non hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo il consenso esplicito, il limite ENI per ogni istanza di container non è diverso dal 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'`awsVpcTrunking` account 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 un potenziamento delle interfacce di rete dei container Amazon ECS](#).

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsVpcTrunking`, consulta [Aumento delle interfacce di rete delle istanze di container Linux di Amazon ECS](#).

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 di attività contiene un parametro che richiede un determinato attributo dell'istanza di container, ma non disponi di istanze di container in grado di soddisfare questo requisito, l'attività non potrà essere posizionata.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di `awsVpc` rete EC2 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 se ci sono 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 di attività Amazon ECS per Fargate](#) e [Configurazione dell'agente del container Amazon ECS](#).

service (**service-name**) non è stato in grado di effettuare un'operazione. 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 nel servizio.

Per informazioni su come eseguire il debug delle attività interrotte, consulta [Amazon ECS ha interrotto i messaggi di errore relativi alle attività](#).

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 di container tramite SSH, puoi esaminare i log dell'agente. Per ulteriori informazioni, consulta [Parametri di configurazione del log 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**) (task**task-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 dell'integrità perché il gruppo di destinazione non viene trovato. Il messaggio include l'ID dell'attività per facilitare l'identificazione dell'attività specifica che non supera i controlli dell'integrità. Devi eliminare il servizio e ricrearlo. Non eliminare alcun gruppo di destinazione Elastic Load Balancing a meno che il servizio Amazon ECS corrispondente non sia già stato eliminato.

service (**service-name**) task set (**taskSet-ID**) (task**task-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 dell'integrità perché la destinazione non viene trovata. Il messaggio include l'ID dell'attività per facilitare l'identificazione dell'attività specifica che non supera i controlli dell'integrità.

Le policy di autorizzazione IAM sono state configurate o modificate in modo errato ed ECS non è più in grado di gestire il servizio

Il servizio non è in grado di gestire le attività a causa di policy di autorizzazione IAM non configurate o modificate in modo errato. Il ruolo IAM associato al servizio o alle attività ECS potrebbe non disporre delle autorizzazioni richieste.

Per risolvere il problema, aggiungi le autorizzazioni necessarie al ruolo IAM. Per ulteriori informazioni sulla gestione delle autorizzazioni IAM, consulta [Aggiunta e rimozione di autorizzazioni per identità IAM](#) nella Guida per l'utente IAM.

La relazione di attendibilità IAM è stata configurata o modificata in modo errato ed ECS non è più in grado di gestire il servizio

Il servizio non è in grado di gestire le attività a causa di relazioni di attendibilità IAM non configurate o modificate in modo errato. Il ruolo IAM associato al servizio o alle attività ECS potrebbe avere una policy di attendibilità errata.

Per risolvere il problema, configura una policy di attendibilità per il ruolo utilizzato nella definizione dell'attività. Per ulteriori informazioni sulla creazione di policy di attendibilità per ruoli personalizzati, consulta [Creazione di un ruolo per un caso d'uso personalizzato](#) nella Guida per l'utente di IAM.

service (***service-name***) non è riuscito ad avviare ***number*** attività per la distribuzione. ***deployment-id***

Il service scheduler invia questo messaggio di evento quando un flusso di lavoro di distribuzione avvia correttamente alcune attività ma non riesce ad avviare tutte le attività richieste a causa di errori di capacità insufficiente. Ciò si verifica in genere quando Circuit Breaker è abilitato e fornisce visibilità sul motivo per cui le distribuzioni potrebbero fallire o ripristinare.

Il messaggio include il motivo specifico dell'errore, ad esempio CPU, memoria o altri vincoli di risorse insufficienti. Questo aiuta a capire quali risorse devono essere utilizzate per risolvere il problema di distribuzione.

Per ulteriori informazioni, consulta [service \(***service-name***\) non è stato in grado di effettuare un'attività perché nessuna istanza del contenitore soddisfaceva tutti i requisiti..](#)

service (***service-name***) non è riuscito a inserire attività nel cluster perché il limite di capacità di provisioning delle attività è stato superato.

Il service scheduler invia questo messaggio di evento quando il cluster ha raggiunto il limite di 500 attività che possono essere eseguite contemporaneamente nello PROVISIONING stato. Si tratta di un limite a livello di cluster, non di un problema specifico del servizio.

Ciò si verifica in genere quando si avvia un servizio con un numero elevato di attività desiderate con una capacità preimpostata limitata o quando più servizi vengono avviati contemporaneamente con un elevato tasso di abbandono delle attività.

Per risolvere il problema:

- Attendi il completamento del provisioning delle attività esistenti e passa allo stato. RUNNING
- Prendi in considerazione la possibilità di scalare i tuoi servizi in modo più graduale per evitare di raggiungere il limite di approvvigionamento.
- Controlla la configurazione del provider di capacità del cluster per assicurarti che siano disponibili risorse adeguate.

Per ulteriori informazioni sulle quote dei servizi Amazon ECS, consulta gli [endpoint e le quote di Amazon Elastic Container Service](#) nell'Amazon Web Services General Reference.

Messaggi degli eventi relativi a malfunzionamenti del servizio Amazon ECS

Di seguito sono riportati alcuni esempi di messaggi relativi a malfunzionamenti negli eventi di servizio.

EC2: (service **service-name**) (task) (instance **task-id**) (port **instance-idport-number**) non è integro in (target-group) a causa di (motivo **target-group-name**) **failure-reason**

Questo messaggio indica che un'attività in esecuzione su un' EC2 istanza non supera i controlli di integrità. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [In che modo posso far superare il controllo di integrità di Application Load Balancer EC2 alle mie attività Amazon ECS che utilizzo?](#)

EC2 con TaskSet: (service **service-name**, TaskSet **taskSet-id**) (task) (instance) (port **task-id**) non è **instance-id** integro in (**port-number**target-group) a causa di (motivo) **target-group-namefailure-reason**

Questo messaggio indica che un'attività in un set di attività in esecuzione su un'istanza non supera i controlli di integrità. EC2 Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Come faccio a far superare il controllo di integrità di Application Load Balancer EC2 alle mie attività Amazon ECS che utilizzo?](#)

Fargate: (service **service-name**) (task) (port **task-idport-number**) non è integro in (target-group) a causa di (motivo **target-group-name**) **failure-reason**

Questo messaggio indica che un'attività di Fargate non supera i controlli dell'integrità.

Per ulteriori informazioni sulla risoluzione degli errori dei controlli dell'integrità nelle attività di Fargate, consulta [Come posso risolvere i problemi relativi ai controlli dell'integrità delle attività di Amazon ECS su Fargate?](#).

Fargate con TaskSet: (service ***service-name***, TaskSet ***taskSet-id***) (task) (port) non è ***task-id*** integro in (***port-number***target-group) a causa di (motivo) ***target-group-namefailure-reason***

Questo messaggio indica che un'attività in un set di attività in esecuzione su Fargate non supera i controlli dell'integrità.

Per ulteriori informazioni sulla risoluzione degli errori dei controlli dell'integrità nelle attività di Fargate, consulta [Come posso risolvere i problemi relativi ai controlli dell'integrità delle attività di Amazon ECS su Fargate?](#).

Messaggi relativi agli eventi del servizio di ribilanciamento del servizio della zona di disponibilità di Amazon ECS

Di seguito sono elencati alcuni esempi di messaggi di evento relativi ai servizi che potresti visualizzare:

service (***service-name***) non è bilanciato in AZ con ***number-tasks*** le attività in, in e in. ***Availability Zone 1number-tasksAvailability Zone 2number-tasksAvailability Zone 3*** Ribilanciamento AZ in corso.

Il pianificatore dei servizi invia un evento di servizio service (***service-name***) is not AZ balanced quando il numero di attività non è distribuito uniformemente tra le zone di disponibilità. Non ci sono azioni da intraprendere. È un evento informativo.

service (***service-name***) è bilanciato in AZ con ***number-tasksnumber-tasks*** task in ***Availability Zone 1Availability Zone 2***, tasks in e ***number-tasks*** tasks in***Availability Zone 3***.

Il pianificatore dei servizi invia un evento di servizio service (***service-name***) is AZ balanced al termine del ribilanciamento del servizio nella Zona di disponibilità. Non ci sono azioni da intraprendere. È 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 ci sono azioni da intraprendere. È 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 stop ***number*** tasks in ***Availability Zone*** service quando interrompe le attività in una zona di disponibilità a causa del ribilanciamento del servizio. Non ci sono azioni da intraprendere. È 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 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 stato in grado di 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 usa l'API `UpdateTaskProtection` con l'opzione `protectionEnabled` impostata su `false` per annullare l'impostazione della 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*** 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. È un evento informativo. Amazon ECS invia eventi aggiuntivi che forniscono informazioni aggiuntive.

Risoluzione dei problemi dei bilanciatori del carico per il servizio in Amazon ECS

I servizi Amazon ECS possono registrare i processi mediante un bilanciatore del carico Elastic Load Balancing. Gli errori di configurazione del bilanciatore del carico sono una causa comune dell'interruzione delle attività. Se le attività interrotte erano state avviate da servizi che utilizzano un bilanciatore del carico, 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 bilanciatore del carico.

Il bilanciatore del carico Elastic Load Balancing non è configurato per tutte le zone di disponibilità

Il tuo bilanciatore del carico 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 bilanciatore del carico e avvia un'attività su

un'istanza di container che si trova in una zona di disponibilità non configurata per l'utilizzo da parte del bilanciatore del carico, l'attività non supera il controllo dell'integrità. Di conseguenza, l'attività viene interrotta.

Il controllo dell'integrità del bilanciatore del carico Elastic Load Balancing non è configurato correttamente

I parametri di controllo dell'integrità del bilanciatore del carico possono essere troppo restrittivi o puntare a risorse inesistenti. Se un'istanza di container risulta non integra, viene rimossa dal bilanciatore del carico. Assicurati di verificare che i parametri seguenti siano configurati correttamente per il bilanciatore del carico del tuo servizio.

Ping Port (Porta ping)

Il valore Ping Port per il controllo dello stato di un bilanciatore del carico corrisponde alla porta delle istanze di container verificate dal bilanciatore del carico per determinare se sono integre. Se questa porta non è configurata correttamente, il bilanciatore del carico 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 del controllo dell'integrità del bilanciatore del carico. È un endpoint dell'applicazione in grado di restituire un codice di stato corretto (ad esempio, 200) quando l'applicazione funziona correttamente. 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 di mancata integrità pari a 2 e un intervallo tra i controlli dell'integrità di 30 secondi, la tua attività ha 60 secondi di tempo per rispondere al ping del controllo dell'integrità 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 bilanciatore del carico devono rimanere nella definizione di attività.

Hai raggiunto il numero limite di attività che puoi eseguire in un dato momento.

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 di dimensionamento automatico del servizio in Amazon ECS

Application Auto Scaling disattiva i processi di riduzione orizzontale mentre sono in corso le implementazioni di Amazon ECS e vengono riattivati una volta completata l'implementazione. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consultare [Suspending and resuming scaling for Application Auto Scaling](#).

Acquisizione di eventi Amazon ECS nella console

La console Amazon ECS fornisce una funzionalità di acquisizione degli eventi che archivia gli eventi generati da Amazon ECS, come azioni di servizio e modifiche dello stato delle attività, in Amazon

CloudWatch Logs through. EventBridge Questa funzionalità include un'interfaccia di query con funzionalità di filtraggio per il monitoraggio e la risoluzione dei problemi.

Gli eventi forniscono informazioni dettagliate sul funzionamento delle implementazioni, dei servizi, delle attività e delle istanze. È possibile utilizzare queste informazioni per risolvere i problemi relativi alle implementazioni non riuscite di attività o servizi.

Quando attivi l'acquisizione degli eventi, hai accesso a tutti gli eventi generati da Amazon ECS per un periodo di conservazione a tua scelta, che va oltre i limiti nativi degli ultimi 100 eventi non filtrati o attività interrotte visibili solo per 1 ora.

Come funziona

L'acquisizione degli eventi consente EventBridge di archiviare gli eventi in un gruppo di log Amazon CloudWatch Logs predefinito. La console Amazon ECS fornisce query e opzioni di filtro predefinite e correla gli eventi per fornire cicli di vita delle attività in un formato intuitivo.

È possibile eseguire una query e recuperare i seguenti tipi di eventi:

- Eventi di azione del servizio: aiutano a identificare problemi di provisioning o allocazione delle risorse
- Eventi del ciclo di vita delle attività: aiutano a identificare il motivo per cui le attività o i container non vengono avviati o interrompono l'esecuzione

La console Amazon ECS ti consente di configurare l'acquisizione degli eventi con un clic e fornisce query e filtri di uso comune senza dover imparare i linguaggi di query o navigare tra più console.

Event types (Tipi di evento)

Event Capture archivia tutti gli eventi generati da Amazon ECS nelle seguenti categorie:

Eventi di modifica dello stato dei processi

Arresti dei container e altri eventi di terminazione, che puoi utilizzare per la risoluzione dei problemi o per monitorare le tempistiche del ciclo di vita delle attività.

Operazioni di servizio

Eventi come il raggiungimento dello stato stazionario, l'immissione fallita delle attività o la limitazione delle risorse.

Modifiche agli stati di implementazione del servizio

Eventi come implementazioni in corso, completate o non riuscite, attivati dalle impostazioni di circuit breaker e rollback, per monitorare lo stato dell'implementazione di un servizio.

Modifiche dello stato dell'istanza di container

Per i carichi di lavoro su EC2 Amazon ECS Managed Instances, gli eventi mostrano lo stato di connessione e disconnessione.

Configurazione dei gruppi di log

Quando attivi l'acquisizione degli eventi, Amazon ECS crea automaticamente le seguenti risorse:

- Un gruppo di CloudWatch log di Amazon Logs denominato `/aws/events/ecs/containerinsights/${clusterName}/performance`
- Una EventBridge regola che acquisisce tutti gli eventi dall'`aws.ecs` origine e li inoltra al gruppo di log

È possibile specificare un periodo di conservazione per il gruppo di log da 1 giorno a 10 anni. Il periodo di conservazione predefinito è di 7 giorni.

Considerazioni

Quando utilizzi l'acquisizione di eventi, considera quanto segue:

- Event Capture archivia tutti gli eventi per semplicità. Non è possibile configurare regole nella console Amazon ECS per acquisire solo eventi specifici.
- La console Amazon ECS fornisce criteri di interrogazione predefiniti. Per le query avanzate, usa Amazon CloudWatch Logs Logs Insights per interrogare direttamente il gruppo di log.
- La funzionalità Live Tail non è disponibile nella console Amazon ECS. Usa Amazon CloudWatch Logs direttamente per live tail.
- Quando disabiliti l'acquisizione degli eventi, la EventBridge regola viene eliminata.
- L'acquisizione degli eventi comporta costi aggiuntivi per l'inserimento EventBridge dei dati, lo storage di Amazon CloudWatch Logs e l'esecuzione delle query.

[Per informazioni sui EventBridge prezzi, consulta la pagina dei prezzi. EventBridge](#)

Per informazioni sui CloudWatch prezzi, consulta la pagina [CloudWatch dei prezzi](#).

Risoluzione dei problemi basata sugli eventi

Utilizza gli eventi generati da Amazon ECS per rispondere a domande comuni sulla risoluzione dei problemi.

Analisi degli errori delle attività

È possibile esaminare gli eventi di modifica dello stato dell'`STOPPED` attività, i codici di arresto e i codici di uscita del contenitore per determinare il motivo per cui un'attività non è stata avviata o non è riuscita durante l'esecuzione.

È possibile esaminare gli eventi relativi alle azioni di servizio per verificare gli errori di posizionamento e le informazioni sui vincoli di risorse per determinare il motivo per cui un'attività non è riuscita a essere eseguita a causa di vincoli di risorse

Scenari comuni di errore delle attività

Gli errori anomali delle attività più comuni sono correlati ai seguenti problemi:

- Errori di implementazione del servizio CI/CD
- Errori di scalabilità automatica
- Errori di ribilanciamento delle attività
- Uscite anomale dei contenitori, ad esempio errori (OOM) out-of-memory

Gli errori anomali delle attività producono eventi di modifica dello stato dell'`STOPPED` attività con un `EssentialContainerExited` codice di arresto `OR.TaskFailedToStart`. È possibile filtrare in base a questi codici di arresto per esaminare i comportamenti di esecuzione e arresto dei container.

Attiva l'acquisizione degli eventi per un cluster Amazon ECS esistente

Puoi abilitare l'acquisizione di eventi su un cluster Amazon ECS esistente per archiviare gli eventi generati da Amazon ECS in Amazon CloudWatch Logs tramite EventBridge. Questa funzionalità consente di monitorare e risolvere i problemi relativi agli errori delle attività, alle implementazioni di servizi e ad altre attività del cluster.

Dopo aver abilitato l'acquisizione degli eventi, Amazon ECS crea le seguenti risorse:

- Un gruppo di CloudWatch log di Amazon Logs denominato `/aws/events/ecs/containerinsights/${clusterName}/performance`
- Una EventBridge regola che cattura tutti gli eventi dalla fonte `aws.ecs`

Nella visualizzazione cluster viene visualizzata una scheda Cronologia, che consente di interrogare gli eventi del ciclo di vita delle attività e le azioni di servizio. L'acquisizione degli eventi inizia immediatamente e memorizza tutti gli eventi generati da Amazon ECS in base al periodo di conservazione specificato.

Prerequisiti

- Un cluster Amazon ECS esistente
- Autorizzazioni IAM appropriate per modificare le impostazioni del cluster e creare risorse Amazon CloudWatch Logs

Attiva l'acquisizione degli eventi tramite la console

1. Apri la console alla <https://console.aws.amazon.com/ecs/v2>.
2. Nel pannello di navigazione scegliere Cluster.
3. Seleziona il cluster in cui desideri abilitare l'acquisizione degli eventi.

Si apre la pagina dei dettagli del cluster.

4. Scegliere Configuration (Configurazione).
5. Nella sezione Eventi ECS, scegli Attiva l'acquisizione degli eventi.

Viene visualizzata la finestra di dialogo Attiva acquisizione eventi.

6. Per l'evento Expire, scegli il periodo di conservazione per il gruppo di log Amazon CloudWatch Logs. L'impostazione di default è 7 giorni.
7. Scegliere Turn On (Attiva).

Visualizzazione degli eventi di modifica dello stato dei servizi e delle attività di Amazon ECS

La console Amazon ECS fornisce una funzionalità di acquisizione degli eventi che archivia gli eventi generati da Amazon ECS, come azioni di servizio e modifiche dello stato delle attività, in Amazon

CloudWatch Logs through. EventBridge Questa funzionalità include un'interfaccia di interrogazione con funzionalità di filtraggio per migliorare il monitoraggio e la risoluzione dei problemi.

Gli eventi forniscono informazioni dettagliate sul funzionamento delle implementazioni, dei servizi, delle attività e delle istanze. È possibile utilizzare queste informazioni per risolvere i problemi relativi alle implementazioni non riuscite di attività o servizi.

È possibile utilizzare uno dei seguenti criteri per filtrare gli eventi:

- ID di distribuzione (disponibile solo nella pagina dei dettagli del servizio)
- Ora di inizio
- Ora di fine
- Nome del servizio (applicabile solo nella pagina dei dettagli del cluster, nella pagina dei dettagli del servizio, sarà l'impostazione predefinita per il servizio corrente)
- ID attività
- Ultimo stato dell'attività
- Famiglia di definizioni delle attività
- Revisione della definizione delle attività

Visualizzazione degli eventi a livello di cluster

1. [Apri la console nella versione 2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)

2. Scegli Cluster.

Viene visualizzata la pagina con l'elenco dei cluster.

3. Scegli il cluster .

Si apre la pagina dei dettagli del cluster.

4. In Cronologia, determina gli eventi da visualizzare.

- a. Per visualizzare gli eventi relativi alle azioni di servizio, scegli Eventi di azione del servizio.
- b. Per visualizzare gli eventi di modifica dello stato dell'attività, scegli Eventi di modifica dello stato dell'attività.
- c. (Facoltativo) In Criteri di interrogazione, inserite i filtri per gli eventi che desiderate visualizzare.

5. Scegli Esegui query.

Gli eventi vengono visualizzati in un elenco.

6. Per visualizzare i dettagli completi dell'evento, scegli l'evento.

Visualizzazione a livello di servizio

1. [Apri la console alla v2. https://console.aws.amazon.com/ecs/](https://console.aws.amazon.com/ecs/)
2. Nella pagina Cluster, scegliere il cluster.
3. Nella pagina dei dettagli del cluster, nella sezione Servizi, selezionare il servizio.

Si apre la pagina dei dettagli del servizio.

4. In Cronologia, determina gli eventi da visualizzare.
 - a. Per visualizzare gli eventi relativi alle azioni di servizio, scegli Eventi di azione del servizio.
 - b. Per visualizzare gli eventi di modifica dello stato dell'attività, scegli Eventi di modifica dello stato dell'attività.
 - c. (Facoltativo) In Criteri di interrogazione, inserite i filtri per gli eventi che desiderate visualizzare.
5. Scegli Esegui query.

Gli eventi vengono visualizzati in un elenco.

6. Per visualizzare i dettagli completi dell'evento, scegli l'evento.

Risoluzione degli errori relativi a CPU o memoria non valide nella definizione delle attività 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 utilizzi 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. Quando la definizione di attività è registrata, viene convertito in un numero intero che indica le unità CPU. Il valore `memory` può essere espresso in MiB o GB in una definizione dell'attività. Quando la definizione di attività è registrata, viene convertito in un numero intero che indica i MiB.

Per le definizioni di attività che specificano FARGATE per il parametro `requiresCompatibilities` (anche se è specificato EC2), è necessario utilizzare uno dei valori nella tabella seguente. Questi valori determinano l'intervallo di valori supportati per i parametri 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
8192 (8 vCPU)	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
 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.		

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

Il meccanismo di controllo della CPU è diverso da quello di EC2 Fargate:

- Per le attività ospitate su Amazon EC2: Amazon ECS utilizza il periodo della CPU e la quota della CPU per controllare le dimensioni delle attività, i limiti rigidi della CPU. Quando si specifica la vCPU nella definizione dell'attività, Amazon ECS traduce il valore nelle impostazioni del periodo CPU e della quota CPU applicabili al cgroup.
- Per le attività ospitate su Fargate: Amazon ECS usa le quote CPU per controllare l'allocazione della CPU. I valori relativi alla quota CPU e al periodo non vengono usati per limitare la CPU nelle attività Fargate.

Per EC2 le attività di Amazon, 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 microsecondi. Quando la quota della CPU è uguale al periodo della CPU significa che a 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 CPU ha un minimo di 1ms. È possibile usare questi valori per impostare i limiti per il numero di CPU. Quando si modifica il periodo della CPU senza modificare la quota CPU, si ottengono limiti effettivi 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.

Visualizzare i log dell'agente 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 log dei container con il raccoglitore di log di Amazon ECS](#).

Sistema operativo Linux

Il processo `ecs-init` archivia i log in `/var/log/ecs/ecs-init.log`.

Il file `ecs-init.log` 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 il raccoglitore di log di Amazon ECS per Windows. Per ulteriori informazioni, consulta [Raccoglitore di log di Amazon ECS 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
```

È possibile attivare la registrazione di debug per l'agente Amazon ECS e il daemon Docker. Questa opzione permette 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 container in esecuzione sull'istanza. Prima di eseguire il comando seguente, svuota l'istanza di container e sposta tutte le attività importanti su altre istanze di container.

Esegui il seguente comando per attivare la registrazione.

```
.\ecs-logs-collector.ps1 -RunMode debug
```

Raccolta log dei container con il raccoglitore di log di Amazon ECS

Note

Non puoi usare il raccoglitore di log di Amazon ECS sulle istanze gestite da Amazon ECS.

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 GitHub per [Linux](#) che 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.

Note

Su Amazon Linux ottimizzato per Amazon ECS AMIs versione 20250909 e successive, il raccoglitore di log di Amazon ECS è preinstallato e pronto all'uso senza dover essere scaricato da `/opt/amazon/ecs/ecs-logs-collector.sh` GitHub Per ulteriori informazioni, consulta [ECS Logs Collector](#) nella documentazione dell'AMI ottimizzata per ECS.

Al momento, il programma di raccolta log di Amazon ECS supporta i seguenti sistemi operativi:

- Amazon Linux
- Red Hat Enterprise Linux
- Ubuntu
- Debian
- Windows Server

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/awslabs/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 daemon Docker e l'agente del container di Amazon ECS, aggiungi l'opzione `--mode=enable-debug` al seguente comando. Questo potrebbe portare al 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 [Drenare le istanze di container di Amazon ECS](#).

```
[ec2-user ~]$ sudo bash ./ecs-logs-collector.sh
```

Important

Consigliamo di modificare i log e rimuovere tutti i dati sensibili dai file. cercare dati noti e anche variabili di ambiente quali `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 log, che puoi condividere per ricevere assistenza diagnostica. [Supporto AWS](#)

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.

Note

Per abilitare la modalità di debug per il daemon Docker e l'agente del container di Amazon ECS, aggiungi l'opzione `-RunMode debug` al seguente comando. 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 [Drenare le istanze di container di Amazon ECS](#).

```
.\ecs-logs-collector.ps1
```

Important

Consigliamo di modificare i log e rimuovere tutti i dati sensibili dai file. cercare dati noti e anche variabili di ambiente quali `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 dell'agente

L'API di introspezione dell'agente Amazon ECS fornisce informazioni sullo stato generale dell'agente Amazon ECS e delle istanze di container.

Puoi utilizzare l'API di introspezione dell'agente per ottenere l'ID Docker per un container nella 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 interrotta e l'altra attualmente in esecuzione.

Note

Il comando seguente viene reindirizzato tramite `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 del container di 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 utility a 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 container 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 del 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 `head` comando per brevità. 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 container 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 dettagliato dal daemon Docker in Amazon ECS

In caso di problemi con i container o le immagini Docker, puoi abilitare la modalità di debug sul tuo daemon Docker. L'utilizzo del debug fornisce un output più dettagliato dal daemon. È possibile usarlo 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 le sezioni [Abilitazione del debug](#) e [Controllo e configurazione di Docker tramite systemd](#) nella documentazione Docker.

Come usare la modalità di debug del daemon Docker nell'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
\\\"}]}\n\" 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 a **API error (500): devmapper** di Docker in Amazon 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 del container di Amazon ECS 1.8.0 e versioni successive) Puoi ridurre la quantità di tempo in cui i container arrestati o terminati rimangono nelle 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 il valore è troppo basso, potresti non riuscire a esaminare i container interrotti o visualizzare i log prima che vengano rimossi. Per ulteriori informazioni, consulta [Configurazione dell'agente del container Amazon ECS](#).

- Puoi rimuovere i container non in esecuzione e le immagini inutilizzate dalle istanze di container. 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 inutilizzati all'interno dei container. 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/"
```

Risoluzione dei 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 tramite Exec Checker

Lo script ECS Exec Checker consente di verificare e convalidare che il cluster Amazon ECS e l'attività abbiano soddisfatto i prerequisiti per l'utilizzo della funzionalità ECS Exec. Lo script ECS Exec Checker verifica che sia AWS CLI l'ambiente che il cluster e le attività siano pronte per ECS Exec, chiamandone diverse per conto dell'utente. 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 ACS Exec](#).
- L'agente SSM non è installato o non è in esecuzione.
- È presente un endpoint Amazon VPC di interfaccia per Amazon ECS, ma non ne esiste uno per Session Manager di Systems Manager.

Risoluzione dei problemi relativi ad Amazon ECS Anywhere

Amazon ECS Anywhere fornisce supporto per la registrazione di una istanza esterna, ad esempio un server on-premises o una macchina virtuale (VM) nel 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 un'attivazione di Systems Manager, specifica 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 istanza. 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 log dei container con il raccoglitore di log di Amazon ECS](#).

Risoluzione dei problemi relativi al caricamento delle classi Java su Fargate

Le applicazioni Java in esecuzione su Fargate potrebbero riscontrare problemi di caricamento delle classi dopo gli aggiornamenti della piattaforma, in particolare quando l'applicazione si basa su un comportamento di caricamento delle classi non deterministico. Ciò può manifestarsi sotto forma di errori di inserimento delle dipendenze, errori Spring Boot o altre eccezioni di runtime che non erano presenti nelle implementazioni precedenti.

Caratteristiche

Potresti riscontrare i seguenti sintomi:

- Errori di inserimento delle dipendenze di Spring Boot
- `ClassNotFoundException` o eccezioni `NoClassDefFoundError`
- Le applicazioni che prima funzionavano su Fargate ora presentano problemi in maniera intermittente
- La stessa immagine del contenitore funziona su Amazon EC2 ma non funziona su Fargate
- Comportamento incoerente tra le implementazioni con immagini di container identiche

Cause

In genere questi problemi si verificano a causa di:

- Caricamento non deterministico delle classi: le applicazioni Java che dipendono dall'ordine in cui le classi vengono caricate dai file JAR potrebbero non funzionare correttamente quando la piattaforma sottostante modifica le modalità di accesso o memorizzazione nella cache dei file.

- Aggiornamenti della piattaforma: gli aggiornamenti della versione della piattaforma Fargate possono modificare il comportamento del file system sottostante, influenzando l'ordine in cui le classi vengono individuate e caricate.
- Dipendenze per l'ordinamento dei file JAR: applicazioni che si basano implicitamente su sequenze di caricamento JAR specifiche senza una gestione delle dipendenze esplicita.

Risoluzione

Per risolvere i problemi di caricamento delle classi Java su Fargate, implementa pratiche di caricamento delle classi deterministiche:

Correzione immediata

Se hai bisogno di una soluzione immediata:

1. Applica l'ordine di caricamento JAR: specifica in modo esplicito l'ordine in cui i file JAR devono essere caricati nella configurazione del classpath dell'applicazione.
2. Usa una gestione esplicita delle dipendenze: assicurati che tutte le dipendenze siano dichiarate in modo esplicito nella configurazione della build (Maven, Gradle, ecc.) invece di fare affidamento su dipendenze transitive.

Best practice a lungo termine

Implementa queste pratiche per prevenire futuri problemi di caricamento delle classi:

1. Rendi il caricamento delle classi deterministico:
 - Utilizza dichiarazioni di dipendenza esplicite nei tuoi file di build
 - Non basarti sull'ordine di scansione del classpath
 - Usa gli strumenti di gestione delle dipendenze per risolvere i conflitti di versione
 - Usa le opzioni Java Virtual Machine (JVM), come `-verbose:classpath`, per ottenere informazioni sulle classi caricate da JVM.
2. Applicazioni Spring Boot:
 - Usa `@ComponentScan` con pacchetti base espliciti
 - Evita i conflitti di configurazione automatica configurando i bean esplicitamente
 - Usa le annotazioni `@DependsOn` per controllare l'ordine di inizializzazione dei bean

3. Configurazione build:

- Usa le sezioni di gestione delle dipendenze in Maven o in Gradle
- Escludi le dipendenze transitive che causano conflitti
- Usa strumenti come Maven Enforcer Plugin per rilevare i problemi di dipendenza

4. Test:

- Testa l'applicazione con diverse implementazioni JVM
- Esegui test di integrazione che simulano diversi ambienti di implementazione
- Usa gli strumenti per analizzare i conflitti tra classpath durante lo sviluppo

Prevenzione

Per prevenire problemi di caricamento delle classi Java nelle implementazioni future:

- Segui le pratiche deterministiche di caricamento delle classi: progetta la tua applicazione in modo che non dipenda dall'ordine in cui le classi vengono caricate dal classpath.
- Usa la gestione esplicita delle dipendenze: dichiara sempre esplicitamente tutte le dipendenze necessarie e le relative versioni nella configurazione della build.
- Esegui test su più ambienti: testa regolarmente le tue applicazioni su diversi ambienti e versioni della piattaforma per individuare in modo tempestivo potenziali problemi.
- Monitora gli aggiornamenti della piattaforma: continua a informarti sugli aggiornamenti della piattaforma Fargate e testa le tue applicazioni con nuove versioni della piattaforma prima che abbiano un impatto sui carichi di lavoro di produzione.

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'API **RunTask** 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 informazioni sulla limitazione dell'API nell'API **RunTask** di Amazon ECS, consulta la sezione [Limitazione delle richieste API](#) nella Documentazione di riferimento sull'API di Amazon ECS.

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 bilanciatore del carico. I clienti notano delle variazioni nelle velocità di avvio delle attività e dei pod rispetto alle quote rappresentate in precedenza in base alle funzionalità abilitate per i propri servizi.

Adeguamento delle quote di velocità in 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 Quote di servizio .

Risoluzione dei problemi relativi alle istanze gestite da Amazon ECS

Utilizza le seguenti procedure per risolvere i problemi delle istanze gestite di Amazon ECS, inclusi problemi comuni, tecniche di diagnostica e passaggi di risoluzione.

Prerequisiti

Prima di risolvere i problemi relativi alle istanze gestite di Amazon ECS, assicurati di soddisfare i seguenti requisiti.

- AWS CLI è installato e configurato con le autorizzazioni appropriate

Per ulteriori informazioni, consulta [Installing or updating to the latest version of the AWS Command Line Interface](#) nella Guida per l'utente dell'AWS Command Line Interface .

- Accesso a un cluster con il provider di capacità Amazon ECS Managed Instances. Per ulteriori informazioni, consulta [the section called “Creazione di un cluster per Istanze gestite da Amazon ECS”](#).

Scenari per la risoluzione dei problemi comuni

Visualizzazione dei log degli agenti container di Amazon ECS Managed Instances

Puoi visualizzare questi file di log di Amazon ECS in Amazon ECS Managed Instances connettendoti a un contenitore privilegiato in esecuzione nell'istanza.

Fasi di diagnostica

Implementa un contenitore di debug con privilegi e funzionalità Linux come attività Amazon ECS:

Imposta le seguenti variabili di ambiente.

user-input Sostituiscili con i tuoi valori.

```
export ECS_CLUSTER_NAME="your-cluster-name"
export AWS_REGION="your-region"
export ACCOUNT_ID="your-account-id"
```

Crea una definizione di attività utilizzando un file JSON CLI chiamato `node-debugger.json`

```
cat << EOF > node-debugger.json
{
  "family": "node-debugger",
  "taskRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
  "executionRoleArn": "arn:aws:iam::${ACCOUNT_ID}:role/ecsTaskExecutionRole",
  "cpu": "256",
  "memory": "1024",
  "networkMode": "host",
  "pidMode": "host",
  "requiresCompatibilities": ["MANAGED_INSTANCES", "EC2"],
  "containerDefinitions": [
    {
      "name": "node-debugger",
      "image": "public.ecr.aws/amazonlinux/amazonlinux:2023",
      "essential": true,
      "privileged": true,
      "command": ["sleep", "infinity"],
      "healthCheck": {
        "command": ["CMD-SHELL", "echo debugger || exit 1"],
        "interval": 30,
        "retries": 3,
        "timeout": 5
      },
      "linuxParameters": {
        "initProcessEnabled": true
      },
      "mountPoints": [
        {
          "sourceVolume": "host-root",
```

```

        "containerPath": "/host",
        "readOnly": false
    }
],
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "/aws/ecs/node-debugger",
        "awslogs-create-group": "true",
        "awslogs-region": "${AWS_REGION}",
        "awslogs-stream-prefix": "ecs"
    }
}
}
],
"volumes": [
    {
        "name": "host-root",
        "host": {
            "sourcePath": "/"
        }
    }
]
}
EOF

```

Registrarsi, quindi eseguire l'operazione. Esegui i comandi seguenti.

```

aws ecs register-task-definition --cli-input-json file://node-debugger.json

TASK_ARN=$(aws ecs run-task \
  --cluster $ECS_CLUSTER_NAME \
  --task-definition node-debugger \
  --enable-execute-command \
  --capacity-provider-strategy capacityProvider=managed-instances-default,weight=1 \
  --query 'tasks[0].taskArn' --output text)

# Wait for task to be running
aws ecs wait tasks-running --cluster $ECS_CLUSTER_NAME --tasks $TASK_ARN

```

Connect al container. Eseguire il seguente comando seguente.

```
aws ecs execute-command \
```

```
--cluster $ECS_CLUSTER_NAME \  
--task $TASK_ARN \  
--container node-debugger \  
--interactive \  
--command "/bin/sh"
```

Controlla i log degli agenti Amazon ECS:

Nella sessione interattiva del contenitore, esegui i seguenti comandi:

```
# Install required tools  
yum install -y util-linux-core  
  
# View ECS agent logs  
nsenter -t 1 -m -p cat /var/log/ecs/ecs-agent.log | tail -50  
  
# Check agent registration  
nsenter -t 1 -m -p grep "Registered container instance" /var/log/ecs/ecs-agent.log  
  
Example Output:  
  
{"level":"info","time":"2025-10-16T12:39:37.665","msg":"Registered container instance  
with cluster!"}  
  
# Verify capabilities  
nsenter -t 1 -m -p grep "Response contained expected value for attribute" /var/log/ecs/  
ecs-agent.log
```

Controlla le metriche degli agenti:

Esegui il comando seguente per visualizzare i log.

```
# View metrics logs  
nsenter -t 1 -m -p cat /var/log/ecs/metrics.log | tail -20
```

Problemi di posizionamento delle attività

Di seguito sono riportati i sintomi dei problemi di posizionamento delle attività:

- Attività bloccate nello stato IN SOSPESO
- Impossibile avviare le attività su Amazon ECS Managed Instances
- Risorse insufficienti, errori

Fasi diagnostiche

Esegui i comandi seguenti per diagnosticare i problemi di posizionamento delle attività e raccogliere informazioni sulla capacità del cluster, sulle istanze dei container e sui servizi di sistema:

```
# Check cluster capacity
aws ecs describe-clusters --clusters cluster-name --include STATISTICS

# Check cluster capacity providers
aws ecs describe-clusters --clusters cluster-name --include STATISTICS --query
'clusters[].capacityProviders'

# List container instances
aws ecs list-container-instances --cluster cluster-name

# Check container instance details
aws ecs describe-container-instances --cluster cluster-name --container-
instances container-instance-arn

# Check container instance remaining resources CPU/Mem
aws ecs describe-container-instances --cluster $ECS_CLUSTER_NAME --container-
instances container-instance-arn --query 'containerInstances[].remainingResources'

# Check container instance Security Group
aws ecs describe-container-instances --cluster $ECS_CLUSTER_NAME --container-
instances container-instance-arn --query 'containerInstances[].ec2InstanceId' --output
text
aws ec2 describe-instances --instance-ids instance-id --query
'Reservations[0].Instances[0].SecurityGroups'
aws ec2 describe-security-groups --group-ids security-group-id
```

Monitoraggio dei servizi di sistema:

```
# Check Containerd status
nsenter -t 1 -m -p systemctl status containerd.service

# Check Amazon ECS container agent status
nsenter -t 1 -m -p systemctl status ecs
```

Risoluzione

Per risolvere i problemi di posizionamento delle attività, segui questi passaggi per garantire una configurazione e una capacità corrette:

- Verifica i requisiti delle risorse delle attività rispetto alla capacità disponibile
- Verifica i vincoli e le strategie di posizionamento
- Assicurati che il provider di capacità di Amazon ECS Managed Instances sia configurato
- Assicurati che il gruppo di sicurezza dell'attività e dell'istanza del contenitore disponga di una regola in uscita che consenta il traffico per gli endpoint di gestione degli agenti Amazon ECS

Problemi di rete

Di seguito sono riportati i sintomi di problemi di rete:

- Le attività non sono in grado di raggiungere servizi esterni
- Problemi di risoluzione DNS

Fasi diagnostiche

Test di connettività di rete:

Dal contenitore di debug, esegui i seguenti comandi:

Note

Verifica che il gruppo di sicurezza collegato al tuo provider di capacità o all'attività Amazon ECS consenta il traffico.

```
# Install DNS Utility
yum install bind-utils -y

# Test DNS resolution
nslookup amazon.com

# Test external connectivity
curl -I https://amazon.com
```

Vincoli delle risorse

Di seguito sono riportati i sintomi di problemi di rete:

- Attività interrotte a causa di limiti di memoria
- Limitazione della CPU
- Problemi di spazio su disco

Fasi di diagnostica

Esegui comandi per monitorare le risorse e i limiti dei container.

Monitoraggio delle risorse:

```
# Check memory usage
nsenter -t 1 -m -p free -h

# Check disk usage
nsenter -t 1 -m -p lsblk

# Check disk usage
nsenter -t 1 -m -p df -h
```

Limiti del contenitore:

```
# Check OOM kills
nsenter -t 1 -m -p dmesg | grep -i "killed process"
```

Problema di disconnessione dell'agente dell'istanza del contenitore

Di seguito sono riportati i sintomi dei problemi di disconnessione dell'agente dell'istanza del contenitore:

- Le istanze di container vengono visualizzate come disconnesse nella console Amazon ECS
- Attività che non vengono assegnate a istanze specifiche
- Errori di registrazione degli agenti nei registri

Fasi diagnostiche

Se esiste un'attività di privilegio esistente in esecuzione sull'host a cui ECS Exec può accedere, esegui i seguenti comandi per diagnosticare i problemi di connettività dell'agente:

```
# check service status
```



```
nsenter -t 1 -m -p systemctl restart ecs
nsenter -t 1 -m -p systemctl restart containerd

# restart stopped services
nsenter -t 1 -m -p systemctl restart ecs
nsenter -t 1 -m -p systemctl restart containerd
```

Altrimenti, forzate l'annullamento della registrazione delle istanze gestite di Amazon ECS. Esegui il comando seguente:

```
# list ECS Managed Instance container
aws ecs list-container-instances --cluster managed-instances-cluster --query
'containerInstanceArns' --output text

# deregister the specific container instance
aws ecs deregister-container-instance \
    --cluster $ECS_CLUSTER_NAME \
    --container-instance container-instance-arn \
    --force
```

Risoluzione

Per risolvere i problemi di disconnessione degli agenti, segui questi passaggi:

- Verifica le autorizzazioni del ruolo IAM per l'istanza del contenitore
- Verifica che le regole del gruppo di sicurezza consentano il traffico HTTPS in uscita verso gli endpoint ECS
- Garantisci la connettività di rete ai servizi AWS
- Riavvia il servizio dell'agente ECS se necessario: `nsenter -t 1 -m -p systemctl restart ecs`
- Verifica che la configurazione `ECS_CLUSTER` in `/etc/ecs/ecs.config` corrisponda al nome del cluster

Analisi dei log nelle istanze gestite di Amazon ECS

Log di sistema

Utilizza i seguenti comandi per esaminare i log di sistema e identificare potenziali problemi con l'istanza gestita:

```
# Check system messages
nsenter -t 1 -m -p journalctl --no-pager -n 50

# Check kernel logs
nsenter -t 1 -m -p dmesg | tail -20

# Check for disk space errors
nsenter -t 1 -m -p journalctl --no-pager | grep -i "no space\|disk full\|enospc"
```

Usa il EC2 AWS CLI per ottenere l'output della console da un'istanza gestita di Amazon ECS

Usa l'ID dell' EC2 istanza Amazon per recuperare l'output della console.

Sostituiscili *user-input* con i tuoi valori.

```
aws ec2 get-console-output --instance-id instance-id --latest --output text
```

Pulizia

Esegui quanto segue per interrompere l'attività di debug e annullare la registrazione della definizione dell'attività.

```
# Stop debug task
aws ecs stop-task --cluster $ECS_CLUSTER_NAME --task $TASK_ARN

# Deregister task definition (optional)
aws ecs deregister-task-definition --task-definition node-debugger
```

Risorse aggiuntive

Per ulteriori informazioni sulla risoluzione dei problemi delle istanze gestite di Amazon ECS, consulta le seguenti risorse:

- [the section called “Errori delle istanze gestite da Amazon ECS”](#)
- [Risoluzione dei problemi](#)
- [the section called “Configurazione dell'agente del container”](#)

- [the section called “Monitora i container Amazon ECS con ECS Exec”](#)

Risoluzione dei problemi relativi alle istanze gestite da Amazon ECS

Quando si avviano attività con le istanze gestite da Amazon ECS, Amazon ECS tenta innanzitutto di collocare le attività sulla capacità esistente e richiede capacità aggiuntiva per le attività che non possono essere collocate. Se il provisioning dell'istanza fallisce, l'ID della EC2 richiesta Amazon viene incluso nel messaggio di errore dell'attività. Puoi utilizzare questo ID di richiesta per cercare i dettagli della richiesta non riuscita CloudTrail per un'ulteriore risoluzione dei problemi.

Note

Se scegli di applicare le autorizzazioni con privilegi minimi e di specificare le tue autorizzazioni per il profilo dell'istanza anziché utilizzare la policy `AmazonECSInstanceRolePolicyForManagedInstances` gestita, puoi aggiungere le seguenti autorizzazioni per aiutarti a risolvere i problemi relativi alle attività con Amazon ECS Managed Instances:

- `ecs:StartTelemetrySession`
- `ecs:PutSystemLogEvents`

La definizione delle attività non è compatibile con le istanze gestite da Amazon ECS.

Cause comuni

Questo errore si verifica quando la definizione dell'attività contiene parametri o configurazioni non supportati dalle istanze gestite da Amazon ECS. Le incompatibilità più comuni includono modalità di rete, ruoli delle attività o requisiti delle risorse non supportati.

Risoluzione

1. Verifica che la definizione dell'attività usi `requiresCompatibilities` impostato su `MANAGED_INSTANCES`.
2. Assicurati che la definizione dell'attività usi la modalità di rete `awsvpc`.
3. Controlla che i valori di CPU e memoria rientrino negli intervalli supportati per le istanze gestite da Amazon ECS.
4. Controlla il messaggio di errore dettagliato per i dettagli specifici sull'incompatibilità.

Provider di capacità non associato al cluster

Cause comuni

Questo errore si verifica quando il provider di capacità specificato nella strategia del provider di capacità non è associato al cluster oppure non esiste.

Risoluzione

1. Verifica che il provider di capacità esista nel tuo account e nella tua regione.
2. Associa il provider di capacità al cluster usando la CLI o la console Amazon ECS.
3. Assicurati che il provider di capacità sia nello stato ACTIVE prima di utilizzarlo.

Errori relativi alle autorizzazioni dei ruoli dell'infrastruttura

Cause comuni

Questo errore si verifica quando il ruolo dell'infrastruttura Amazon ECS non dispone delle autorizzazioni necessarie per EC2 eseguire operazioni Amazon per tuo conto o quando il ruolo non può essere assunto a causa di problemi di relazione di fiducia.

Risoluzione

1. Verifica che il tuo ruolo dell'infrastruttura abbia un rapporto di fiducia adeguato con Amazon ECS.
2. Assicurati che il ruolo disponga delle EC2 autorizzazioni Amazon richiesteeec2:RunInstances, tra cuiec2:DescribeInstances, eiam:PassRole.
3. Controlla il messaggio codificato di errore di autorizzazione CloudTrail per dettagli specifici sulle autorizzazioni.
4. Aggiorna la policy del ruolo per includere le autorizzazioni mancanti identificate nel messaggio di errore.

VcpuLimitExceeded errore

Cause comuni

Questo errore si verifica quando viene raggiunta la quota del servizio vCPU per la famiglia di tipi di istanza nella regione corrente. Le istanze gestite da Amazon ECS non possono avviare istanze aggiuntive finché non è disponibile la capacità necessaria.

Risoluzione

1. Richiedi un aumento della quota di servizio per la famiglia di tipi di istanze interessata tramite il AWS Support Center.
2. Valuta l'utilizzo di tipi di istanza diversi che rientrano in una categoria di quota vCPU diversa.
3. Termina le EC2 istanze Amazon non utilizzate per liberare capacità vCPU.
4. Rivedi la configurazione del tuo provider di capacità per usare tipi di istanza con requisiti vCPU inferiori.

InsufficientCapacity e i relativi errori di capacità

Cause comuni

Questi errori si verificano quando AWS non dispone di una capacità sufficiente per soddisfare la richiesta dell'istanza. Ciò può includere una capacità insufficiente di istanze, indirizzi o volumi nella zona di disponibilità richiesta.

Risoluzione

1. Prova ad avviare istanze in diverse zone di disponibilità configurando più sottoreti nel tuo provider di capacità.
2. Valuta l'utilizzo di tipi di istanza diversi che potrebbero avere una maggiore capacità disponibile.
3. Attendi e riprova l'operazione in quanto la disponibilità della capacità cambia frequentemente.
4. Per esigenze di capacità persistenti, valuta l'utilizzo di istanze riservate o Savings Plans.

UnauthorizedOperation errore

Cause comuni

Questo errore si verifica quando il servizio Amazon ECS non dispone delle autorizzazioni necessarie per EC2 eseguire operazioni Amazon o passare ruoli IAM. Gli scenari più comuni includono la mancanza di autorizzazioni `ec2:RunInstances` o `iam:PassRole` per il profilo dell'istanza.

Risoluzione

1. Verifica che il ruolo dell'infrastruttura Amazon ECS disponga delle autorizzazioni necessarie per avviare le istanze Amazon EC2 .

2. Assicurati che il ruolo dell'infrastruttura disponga delle autorizzazioni `iam:PassRole` per il profilo dell'istanza usato dalle tue istanze gestite da Amazon ECS.
3. Controlla il messaggio codificato di errore di autorizzazione CloudTrail per dettagli specifici sulle autorizzazioni.
4. Aggiorna la policy del ruolo per includere le autorizzazioni mancanti identificate nel messaggio di errore.

Attività scaduta in attesa di capacità

Cause comuni

Questo errore si verifica quando le istanze impiegano più del previsto per l'avvio e la registrazione nel cluster. Ciò può accadere a causa di limiti di EC2 capacità di Amazon, errori di avvio delle istanze o problemi di connettività di rete.

Risoluzione

1. Verifica lo stato EC2 del servizio Amazon nella tua regione per eventuali problemi in corso.
2. Verifica che le sottoreti dispongano di un numero sufficiente di indirizzi IP disponibili.
3. Assicurati che i tuoi gruppi di sicurezza consentano il traffico necessario per la comunicazione con l'agente Amazon ECS.
4. Valuta l'utilizzo di più zone di disponibilità per migliorare la disponibilità della capacità.
5. Riprova a eseguire l'attività di avvio dell'attività, poiché i limiti di capacità sono spesso temporanei.

Errori di configurazione della rete

Cause comuni

Questi errori avvengono quando ci sono discrepanze tra i requisiti di rete della tua attività e la configurazione di rete del provider di capacità, come discrepanze VPC o configurazione di rete mancante.

Risoluzione

1. Verifica che il provider di capacità sia configurato con il VPC e le sottoreti corretti.
2. Verifica che i gruppi di sicurezza e le sottoreti appartengano allo stesso VPC.

3. Controlla che la configurazione di rete della definizione dell'attività sia compatibile con il provider di capacità.
4. Aggiorna la configurazione del provider di capacità con le impostazioni di rete corrette.

Il Capacity Provider non può essere eliminato a causa di istanze bloccate

Cause comuni

Questi errori si verificano quando le istanze gestite di Amazon ECS sono bloccate in uno DRAINING stato ACTIVE o ma non ci sono attività in esecuzione sulle istanze.

Risoluzione

Per consentire l'eliminazione del provider di capacità, puoi forzare l'annullamento della registrazione delle istanze bloccate utilizzando il seguente comando.

```
aws ecs deregister-container-instance \
  --cluster arn:aws:ecs:us-east-1:111122223333:cluster/MyCluster \
  --container-instance arn:aws:ecs:us-east-1:111122223333:container-instance/
a1b2c3d4-5678-90ab-cdef-1111EXAMPLE \
  --force
```

Risoluzione degli errori delle istanze gestite da Amazon ECS

Di seguito sono riportati alcuni messaggi di errore delle istanze gestite da Amazon ECS e le azioni che puoi intraprendere per correggerli.

Il provider di istanze gestite da Amazon ECS non è supportato senza un nome cluster

Questo errore si verifica quando si tenta di creare un provider di capacità con un parametro del provider di istanze gestite da Amazon ECS senza un campo cluster.

Per risolvere il problema, specifica un nome del cluster valido durante la creazione del provider di capacità.

I dettagli del provider di istanze gestite da Amazon ECS sono necessari durante la creazione di un provider di capacità delle istanze gestite da Amazon ECS.

Questo errore viene visualizzato quando si tenta di creare un provider di capacità con il provider di istanze gestite da Amazon ECS senza fornire l'oggetto effettivo.

Per risolvere il problema, specifica i dettagli del provider di istanze gestite da Amazon ECS valido e riprova.

Il provider di capacità delle istanze gestite da Amazon ECS deve specificare un profilo dell'istanza

Visualizzerai questo errore quando tenti di creare un provider di capacità con il provider Amazon ECS Managed Instances ma non fornisci `Ec2.InstanceProfile`

Per risolvere questo problema, specifica un profilo di EC2 istanza valido per il tuo provider di capacità Amazon ECS Managed Instances. Per ulteriori informazioni, consulta [the section called “Profilo delle istanze gestite da Amazon ECS”](#).

Il provider di capacità di Amazon ECS Managed Instances deve specificare un `InfrastructureRole`

Questo messaggio viene visualizzato quando tenti di creare un provider di capacità con il provider Amazon ECS Managed Instances ma non fornisci `il. InfrastructureRole`

Per correggere questo problema, specifica un ruolo dell'infrastruttura valido per il tuo provider di capacità delle istanze gestite da Amazon ECS. Per ulteriori informazioni, consulta [the section called “ruolo IAM dell'infrastruttura”](#).

Il provider di capacità delle istanze gestite da Amazon ECS deve specificare una configurazione di rete

Questo errore viene visualizzato quando si tenta di creare un provider di capacità con il provider di istanze gestite da Amazon ECS senza fornire la configurazione di rete.

Per risolvere questo problema, specifica una configurazione di rete valida con sottoreti non vuote per il tuo provider di capacità delle istanze gestite da Amazon ECS. Per ulteriori informazioni, consulta [the section called “Rete di attività per le istanze gestite Amazon ECS”](#).

Nessun tipo di istanza soddisfa i requisiti di istanza specificati nel provider di capacità delle istanze gestite da Amazon ECS

Ciò accade quando tenti di creare un provider di capacità con il provider Amazon ECS Managed Instances ma nessun tipo di EC2 istanza soddisfa i requisiti dell'istanza.

Per risolvere questo problema, rivedi e modifica i requisiti dell'istanza in modo che corrispondano ai tipi di istanze disponibili. EC2

Il ruolo arn dell'infrastruttura specificato non è valido

Questo errore si verifica quando InfrastructureRole non segue il formato ARN specifico.

Formato previsto: `arn:partition:iam::account-id:role/role-name`. Specifica un ruolo ARN valido e riprova. Per ulteriori informazioni, consulta [the section called “ruolo IAM dell’infrastruttura”](#).

Il ruolo arn del profilo dell'istanza specificato non è valido

Questo errore si verifica quando il profilo dell'istanza non segue il formato ARN specifico.

Formato previsto: `arn:partition:iam::account-id:instance-profile/profile-name`. Specifica un ruolo ARN valido e riprova. Per ulteriori informazioni, consulta [the section called “Profilo delle istanze gestite da Amazon ECS”](#).

L'ID del gruppo di sicurezza specificato non è valido

Questo errore si verifica quando l'ID del gruppo di sicurezza non segue il formato specifico.

Formato previsto: `sg-xxxxxxx` oppure `sg-xxxxxxxxxxxxxxxxxxxx` (8 o 17 caratteri comprese le lettere (minuscole) e i numeri dopo 'sg-'). Specifica un ID del gruppo di sicurezza e riprova.

L'ID della sottorete specificato non è valido

Questo errore si verifica quando l'ID della sottorete non segue il formato specifico.

Formato previsto: `subnet-xxxxxxx` oppure `subnet-xxxxxxxxxxxxxxxxxxxx` (8 o 17 caratteri comprese le lettere (minuscole) e i numeri dopo 'sottorete-'). Specifica un ID della sottorete e riprova.

Il provider di capacità delle istanze gestite da Amazon ECS deve specificare una configurazione di rete con sottoreti non vuote

Questo errore si verifica quando si tenta di creare un provider di capacità con il provider di istanze gestite da Amazon ECS e una configurazione di rete con sottoreti vuote.

Per risolvere questo problema, specifica una configurazione di rete valida con sottoreti non vuote per il tuo provider di capacità delle istanze gestite da Amazon ECS.

Il provider di capacità di istanze gestite da Amazon ECS non può avere un numero di tag superiore al valore massimo consentito

Questo errore si verifica quando si tenta di creare un provider di capacità con un numero di tag superiore al massimo consentito (45).

Per risolvere questo problema, riduci il numero di tag a 45 o meno e riprova.

Valore non valido per PropagateTags

Questo errore si verifica quando si tenta di creare un provider di capacità con un valore `propagateTags` non valido.

Il valore deve essere uno dei valori `propagateTags` validi. Specifica un valore valido e riprova.

Il provider di capacità specificato esiste già con l'ambito cluster

Questo errore si verifica quando si tenta di creare, aggiornare o eliminare un provider di capacità di istanze gestite da Amazon ECS senza un nome del cluster, ma esiste già un provider di capacità nell'ambito del cluster.

Per modificare la configurazione o eliminare il provider di capacità, aggiorna o elimina il provider di capacità con il parametro `cluster`.

Il provider di capacità specificato esiste già con l'account o con un cluster diverso

Questo errore si verifica quando si tenta di creare, aggiornare o eliminare un provider di capacità di istanze gestite da Amazon ECS con un campo `cluster`, mentre esiste un provider di capacità con ambito `account` esistente o un provider di capacità con ambito `cluster` esistente per un cluster diverso.

Per risolvere il problema, usa un nome del provider di capacità diverso o lavora con il provider di capacità esistente.

Il nome del cluster attivo del provider di capacità non corrisponde al nome del cluster nella richiesta

Questo errore si verifica quando il nome del cluster attivo del provider di capacità non corrisponde al nome del cluster specificato nella richiesta.

Assicurati che il nome del cluster nella tua richiesta corrisponda al cluster associato al provider di capacità.

Il provider di capacità non è il provider di capacità delle istanze gestite da Amazon ECS

Questo errore si verifica quando si tenta di usare un provider di capacità che non è un provider di capacità delle istanze gestite da Amazon ECS in un contesto che ne richiede uno.

Assicurati di usare un provider di capacità delle istanze gestite da Amazon ECS valido per la tua richiesta.

CapacityProvider il nome non deve essere vuoto

Questo errore si verifica quando si tenta di creare un provider di capacità senza specificare il nome del provider di capacità.

Specifica un nome del provider di capacità valido e riprova.

È necessario inserire un nome quando si aggiorna un fornitore di capacità.

Questo errore si verifica quando si tenta di aggiornare un provider di capacità senza specificare il nome del provider di capacità.

Specifica un nome valido e riprova.

I tag non possono essere nulli o contenere elementi nulli

Questo errore si verifica quando si tenta di applicare tag a un provider di capacità con valori nulli.

Assicurati che i valori di tutti i tag siano specificati correttamente e non siano nulli.

Motivi degli errori dell'API Amazon ECS

Quando un'azione API che hai attivato tramite l'API Amazon ECS, la console o AWS CLI esce con un messaggio di errore `failures`, quanto segue potrebbe essere utile nella risoluzione dei problemi della causa. L'errore restituisce un motivo e il nome della risorsa Amazon (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 utilizzate il AWS CLI, assicuratevi che AWS CLI i 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 esempi di messaggi di errore che potresti ricevere durante l'esecuzione dei 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.
<code>DescribeServices</code>	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.
<code>DescribeTasks</code>	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.
<code>DescribeTasks</code>	<code>TaskFailedToStart:</code> <code>RESOURCE: *</code>	In caso di <code>RESOURCE:CPU</code> errori, il numero di CPU richieste dall'attività non è disponibile nelle istanze del contenitore. Ciò si verifica in genere quando il requisito

Azione API	Motivo dell'errore o dell'interruzione	Causa
		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 è presente alcuna istanza di container che soddisfi i vincoli di posizionamento definiti nella definizione di 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 di attività Amazon ECS per 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 a esse 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 awsvpc, 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 di attività Amazon ECS per Fargate e Configurazione dell'agente del container Amazon ECS .
StartTask	MISSING	L'istanza di container su cui hai provato ad avviare l'attività non è stata trovata. Verifica che non siano stati specificati il cluster o la regione corretti oppure che l'ID o l'ARN dell'istanza di container non sia 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 StopServiceDeployment era gestita da un account fraudolento.
TagResource	InvalidParameterException	L'ARN per il servizio su cui stai applicando il tag ha il formato breve. È necessario a la migrazione al formato lungo. Per informazioni sulla migrazione dell'ARN, consulta Eseguire la migrazione di un ARN di servizio breve di Amazon ECS in un ARN lungo .
UpdateTaskProtection	DEPLOYMENT_BLOCKED	Non è possibile impostare la protezione delle attività poiché una o più attività protette impediscono all'implementazione del servizio di raggiungere uno stato stazionario. Annulla l'impostazione della protezione 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 qui descritti, le operazioni API possono non riuscire anche a causa di eccezioni, con conseguenti risposte di errore. Per un elenco di tali eccezioni, consulta [Errori comuni](#).

Risoluzione dei problemi con Amazon Q Developer

Puoi utilizzare [Amazon Q Developer](#) nella console Amazon ECS per diagnosticare e risolvere problemi con le tue risorse Amazon ECS. Per determinati errori e messaggi di stato su contenitori, attività, servizi, distribuzioni e definizioni di attività, la console mostra un'opzione Inspect with Amazon Q Developer. Quando scegli questa opzione, Amazon Q Developer analizza il problema contestualmente e suggerisce possibili cause e passaggi di risoluzione.

Autorizzazioni richieste

- Autorizzazioni per visualizzare le risorse Amazon ECS di cui desideri risolvere i problemi, come cluster, servizi, attività e definizioni delle attività.
- [Autorizzazioni per utilizzare Amazon Q Developer](#) nella console.

- (Consigliato) Autorizzazioni per visualizzare log e metriche correlati, come:
 - CloudWatch Registri
 - CloudWatch

Procedura

1. Apri la console all'indirizzo. <https://console.aws.amazon.com/ecs/v2>
2. Determina la risorsa per la quale desideri risolvere i problemi.

Risorsa	Fasi
Processi	1. Nella pagina Cluster, scegliere il cluster. 2. Selezionare la scheda Tasks (Attività). 3. Scegli l'attività su cui vuoi indagare.
Container	1. Nella pagina Cluster, scegliere il cluster. 2. Scegli la scheda Attività, quindi scegli l'attività. 3. Nella sezione Contenitori, scegli il contenitore che desideri esaminare.
Servizi e implementazioni	1. Nella pagina Cluster, scegliere il cluster. 2. Scegli la scheda Servizi. 3. Scegli il servizio, quindi consulta la sezione Distribuzioni.
Definizioni di processi	1. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo). 2. Scegli la famiglia di definizioni delle attività, quindi scegli la revisione che desideri esaminare.

3. Nella pagina dei dettagli della risorsa, individua lo stato o il motivo di salute che descrive il problema.
4. Fai clic sul motivo dello stato per aprire il popover.

5. Se disponibile, scegli **Inspect with Amazon Q Developer**.
6. Consulta la spiegazione e i passaggi correttivi suggeriti forniti da Amazon Q Developer. Applica le modifiche operative o di configurazione appropriate al tuo ambiente.

Considerazioni

Considera quanto segue quando utilizzi Amazon Q Developer con Amazon ECS:

- Disponibilità dei pulsanti: il pulsante «Ispeziona con Amazon Q Developer» viene visualizzato solo per le risorse che presentano potenziali problemi. Questa opzione non è disponibile per risorse affidabili.
- Operazioni di sola lettura: l'integrazione con Amazon Q Developer esegue solo operazioni di lettura. Non effettua azioni di mutazione o scrittura.
- Elaborazione tra regioni: Amazon Q Developer può elaborare dati tra AWS regioni diverse per fornire analisi basate sull'intelligenza artificiale. Per ulteriori informazioni sull'elaborazione tra regioni, consulta [Elaborazione tra regioni in Amazon Q Developer](#).
- Solo console: questa integrazione è disponibile solo tramite la console. Non è disponibile tramite AWS CLI AWS APIs, o infrastructure as code tools.

Ulteriori informazioni

Per ulteriori informazioni sull'uso di Amazon Q Developer, consulta [Chatting with Amazon Q Developer about AWS](#).

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. L'utente è anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della propria 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](#)
- [Modello di responsabilità condivisa per le istanze gestite da Amazon ECS](#)
- [Best practice sulla 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 servizio 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 tramite 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 richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green](#)
- [Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi 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 in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta [Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service](#))

- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta [Come funziona Amazon Elastic Container Service con IAM](#))
- Amministratore IAM: scrivi policy per gestire l'accesso (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 autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/Facebook Per ulteriori informazioni sull'accesso, consulta [Come accedere all' Account AWS](#) nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta [AWS Signature Version 4 per le richieste API](#) nella Guida per l'utente di IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Consigliamo vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta [Che cos'è il Centro identità IAM?](#) nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un [utente IAM](#) è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ti consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta [Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta [Casi d'uso per utenti IAM](#) nella Guida per l'utente di IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo [passando da un ruolo utente a un ruolo IAM \(console\)](#) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta [Metodi per assumere un ruolo](#) nella Guida per l'utente di IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per maggiori informazioni, consultare [Accesso a risorse multi-account in IAM](#) nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare [Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scegliere tra policy gestite e policy in linea](#) nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#).

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in AWS Organizations. Per ulteriori informazioni, consultare [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations.
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta [Politiche di controllo delle risorse \(RCPs\)](#) nella Guida per l'AWS Organizations utente.

- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, 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 [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 sull'identità	Sì
Policy basate su risorse	No
Operazioni 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 sull'identità: sì

Le policy basate sull'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 di IAM.

Con le policy basate sull'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. 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 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 di 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. In una policy basata sulle risorse è obbligatorio [specificare un'entità principale](#). I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, è possibile specificare un intero account o entità IAM in un altro account come entità principale in una policy basata sulle risorse. 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 entità 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 in una policy. Includere 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

Supporto risorse policy: Parziale

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si 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 entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento Condition specifica quando le istruzioni vengono eseguite in base a criteri definiti. 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. 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	Description	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

Chiave di condizione	Description	Tipi di valutazione
aws: ResourceTag /\$ {} TagKey	La chiave di contesto è formattata "aws:ResourceTag/ <i>tag-key</i>": "<i>tag-value</i> " dove <i>tag-key</i> 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
ecs: ResourceTag /\$ {} TagKey	La chiave di contesto è formattata "ecs:ResourceTag/ <i>tag-key</i>": "<i>tag-value</i> " dove <i>tag-key</i> 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:capacity-provider	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

Chiave di condizione	Description	Tipi di valutazione
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:compute-compatability	La chiave di contesto è formattata "ecs:compute-compatability": " <i>compatability-type</i> " dove <i>compatability-type</i> è "FARGATE«," "oEC2"». 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
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

Chiave di condizione	Description	Tipi di valutazione
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: name space": " <i>namespace-arn</i> " dove si <i>namespace-arn</i> trova l'ARN per AWS Cloud Map lo spazio dei nomi.	ARN, Null
ecs: propagate-tags	La chiave di contesto è formattata "ecs: propagate-tags": " <i>value</i> " dove <i>value</i> è, o. "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

Chiave di condizione	Description	Tipi di valutazione
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, usa la colonna Condition keys per vedere i tasti per la combinazione. action/resource

Supporta ABAC (tag nelle policy): sì

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire 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 Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta [Definizione delle autorizzazioni con autorizzazione ABAC](#) nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi 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 [Aggiunta di tag alle 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

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta [Credenziali di sicurezza temporanee in IAM](#) e [Servizi AWS compatibili con IAM](#) nella Guida per l'utente IAM.

Sessioni di accesso in avanti per Amazon ECS

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama e, in combinazione con la richiesta Servizio AWS, Servizio AWS per effettuare richieste ai servizi downstream. 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 [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 i 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'operazione 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 al servizio, 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. Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

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 di 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 Amazon ECS](#)
- [Consenti agli utenti Amazon ECS di visualizzare le proprie autorizzazioni](#)
- [Esempi di cluster Amazon ECS](#)
- [Esempi di istanze di container Amazon ECS](#)
- [Esempi di definizioni di attività Amazon ECS](#)
- [Esegui esempio di attività Amazon ECS](#)
- [Avvia esempio di attività Amazon ECS](#)
- [Elenca e descrivi alcuni esempi di attività Amazon ECS.](#)
- [Crea un esempio di servizio Amazon ECS](#)
- [Descrizione dei servizi Amazon ECS basati su tag](#)
- [Esempio di negazione della sostituzione del namespace di Amazon ECS Service Connect](#)

Best practice relative alle policy 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 si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le politiche 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 AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere 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 privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle 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 CloudFormation. Per maggiori informazioni, consultare la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente di IAM.

- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare [Convalida delle policy per il Sistema di analisi degli accessi IAM](#) nella Guida per l'utente di 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, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare [Protezione dell'accesso API con MFA](#) nella Guida per l'utente di 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 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 policy 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.

JSON

```

{
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:ListClusters"
            ],
            "Resource": "*"
        }
    ]
}

```

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DescribeClusters",
        "ecs>DeleteCluster"
      ],
      "Resource": ["arn:aws:ecs:us-east-1:123456789012:cluster/cluster-  
name"]
    }
  ]
}
```

La seguente policy IAM concede le autorizzazioni per descrivere ed eliminare un determinato cluster. Le DeleteCluster azioni DescribeClusters e accettano i cluster ARNs come risorse.

JSON

```
{
  "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-1:123456789012: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-1:123456789012:cluster/default"}
        }
    }
]
}

```

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.

JSON

```

{
    "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-1:111122223333: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-1:111122223333: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:

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeregisterContainerInstance",
        "ecs:ListContainerInstances"
      ],
      "Resource": "arn:aws:ecs:us-east-1:123456789012:cluster/cluster_name"
    }
  ]
}
```

Esempi di definizioni di attività 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.

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:RegisterTaskDefinition",
        "ecs:ListTaskDefinitions",
        "ecs:DescribeTaskDefinition"
      ],
      "Resource": "*"
    }
  ]
}
```

Esegui esempio di attività 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:

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:RunTask"],
      "Condition": {
        "ArnEquals": {"ecs:cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/cluster_name"}
      },
      "Resource": ["arn:aws:ecs:us-east-1:123456789012:task-definition/task_family:*"]
    }
  ]
}
```

Avvia esempio di attività 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.

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": ["ecs:StartTask"],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/cluster_name",
          "ecs:container-instances": ["arn:aws:ecs:us-east-1:123456789012:container-instance/cluster_name/container_instance_UUID"]
        }
      },
      "Resource": ["arn:aws:ecs:us-east-1:123456789012:task-definition/task_family:*"]
    }
  ]
}
```

Elenca e descrivi alcuni esempi di attività Amazon ECS.

La seguente policy IAM permette a un utente di descrivere uno specifico processo in un determinato cluster:

JSON

```
{
  "Statement": [
    {
      "Effect": "Allow",
```



```

        "Action": ["ecs:DescribeTasks"],
        "Condition": {
            "ArnEquals": {"ecs:cluster": "arn:aws:ecs:us-
east-1:123456789012:cluster/cluster_name"}
        },
        "Resource": ["arn:aws:ecs:us-
east-1:123456789012: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 Console di gestione AWS:

JSON

```

{
    "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": "*"
    }
]
}

```

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.

JSON

```

{
  "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 negazione della sostituzione del namespace di Amazon ECS Service Connect

La seguente policy IAM impedisce a un utente di sovrascrivere il namespace predefinito di Service Connect in una configurazione di servizio. Il namespace 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 namespace. Utilizza le seguenti chiavi di contesto per richiedere ai servizi di utilizzare un namespace specifico. Nel seguente esempio, sostituisci <region>, <aws_account_id>, <cluster_name> e <namespace_id> con i tuoi valori.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateService",
        "ecs:UpdateService"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/cluster_name",
          "ecs:namespace": "arn:aws:servicediscovery:us-east-1:123456789012: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 che scrivere le politiche da soli. La [creazione di policy gestite dai clienti IAM](#) che forniscono al 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, consultare 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 Console di gestione AWS.

Per visualizzare le autorizzazioni per questa politica, consulta [FullAccessAmazonECS](#) nel Managed Policy Reference.AWS

Amazon ECSInfrastructure RolePolicyForVolumes

Puoi collegare la policy gestita AmazonECSInfrastructureRolePolicyForVolumes anche alle tue entità IAM.

La policy concede le autorizzazioni necessarie ad Amazon ECS per effettuare chiamate AWS API per tuo conto. È possibile associare questa policy al ruolo IAM fornito con la configurazione del volume quando si avviano attività e servizi Amazon ECS. Questo ruolo consente ad Amazon ECS di gestire i volumi associati alle attività. Per ulteriori informazioni, consulta [Ruolo IAM dell'infrastruttura per Amazon ECS](#).

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructure 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 tuo ruolo collegato al servizio [AWSServiceRoleForECS](#).

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 Autorità di certificazione privata 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 .

AmazonECSInfrastructureRolePolicyForLoadBalancers

Puoi collegare la policy `AmazonECSInfrastructureRolePolicyForLoadBalancers` anche alle tue entità IAM. Questa policy concede le autorizzazioni che permettono ad Amazon ECS di gestire le risorse Elastic Load Balancing per tuo conto. La policy comprende:

- Autorizzazioni di sola lettura per descrivere listener, regole, gruppi di destinazione e stato della destinazione
- Autorizzazioni per registrare e annullare la registrazione delle destinazioni presso i gruppi di destinazione
- Autorizzazioni per modificare i listener per Application Load Balancer e Network Load Balancer
- Autorizzazioni per modificare le regole per Application Load Balancer

Queste autorizzazioni permettono ad Amazon ECS di gestire automaticamente le configurazioni del bilanciatore del carico quando i servizi vengono creati o aggiornati, garantendo il corretto instradamento del traffico verso i container.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructureRolePolicyForLoadBalancers](#) nel AWS Managed Policy Reference.

AmazonECSInfrastructureRolePolicyForManagedInstances

Puoi collegare la policy AmazonECSInfrastructureRolePolicyForManagedInstances anche alle tue entità IAM. Questa politica concede le autorizzazioni richieste da Amazon ECS per creare e aggiornare EC2 le risorse Amazon per ECS Managed Instances per tuo conto. La policy comprende:

- Autorizzazioni per creare e gestire modelli di EC2 lancio Amazon per istanze gestite
- Autorizzazioni per il provisioning di EC2 istanze Amazon utilizzando `CreateFleet` `RunInstances`
- Autorizzazioni per creare e gestire tag su EC2 risorse Amazon create da ECS
- Autorizzazioni per trasferire ruoli IAM alle istanze Amazon per EC2 istanze gestite
- Autorizzazioni per creare ruoli collegati ai servizi per le istanze Amazon Spot EC2
- Autorizzazioni di sola lettura per descrivere le EC2 risorse Amazon tra cui istanze, tipi di istanze, modelli di avvio, interfacce di rete, zone di disponibilità, gruppi di sicurezza, sottoreti e immagini VPCs EC2

Queste autorizzazioni consentono ad Amazon ECS di fornire e gestire automaticamente le istanze Amazon per EC2 le tue istanze gestite ECS, garantendo la corretta configurazione e la gestione del ciclo di vita delle risorse di elaborazione sottostanti.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructureRolePolicyForManagedInstances](#) nel AWS Managed Policy Reference.

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.

AmazonECSInfrastructureRoleforExpressGatewayServices

Puoi collegare la policy `AmazonECSInfrastructureRoleforExpressGatewayServices` anche alle tue entità IAM. Questa politica concede le autorizzazioni richieste da Amazon ECS per creare e aggiornare applicazioni Web utilizzando Express Services per tuo conto. La policy comprende:

- Autorizzazioni per creare ruoli collegati ai servizi per Amazon ECS Application Auto Scaling
- Autorizzazioni per creare, modificare ed eliminare Application Load Balancer, listener, regole e gruppi target
- Autorizzazioni per creare, modificare ed eliminare gruppi di sicurezza VPC per risorse gestite da ECS
- Autorizzazioni per richiedere, gestire ed eliminare certificati tramite ACM SSL/TLS
- Autorizzazioni per configurare le politiche e gli obiettivi di Application Auto Scaling per i servizi Amazon ECS
- Autorizzazioni per creare e gestire CloudWatch allarmi per i trigger di scalabilità automatica
- Autorizzazioni di sola lettura per descrivere sistemi di bilanciamento del carico, risorse VPC, certificati, configurazioni di scalabilità automatica e allarmi CloudWatch

Queste autorizzazioni consentono ad Amazon ECS di fornire e gestire automaticamente i componenti dell'infrastruttura necessari per le applicazioni Web Express Services, tra cui bilanciamento del carico, gruppi di sicurezza, certificati SSL e configurazioni di scalabilità automatica.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInfrastructure RoleforExpressGatewayServices](#) nel AWS Managed Policy Reference.

AmazonECSComputeServiceRolePolicy

La `AmazonECSComputeServiceRolePolicy` policy è allegata al ruolo `ECSCompute ServiceRole` collegato al servizio Amazon. Per ulteriori informazioni, consulta [Uso dei ruoli per gestire le istanze gestite da Amazon ECS](#).

Questa policy include le autorizzazioni che consentono ad Amazon ECS di completare le attività seguenti:

- Amazon ECS può descrivere ed eliminare i modelli di avvio.

- Amazon ECS può descrivere ed eliminare le versioni dei modelli di avvio.
- Amazon ECS può terminare le istanze.
- Amazon ECS può descrivere i seguenti parametri dei dati delle istanze:
 - Istanza
 - Interfacce di rete delle istanze: Amazon ECS può descrivere come gestire il ciclo di vita dell' EC2 istanza.
 - Finestra di eventi dell'istanza: Amazon ECS può descrivere le informazioni sulla finestra di eventi per determinare se il flusso di lavoro può essere interrotto per l'applicazione di patch all'istanza.
 - Stato dell'istanza: Amazon ECS può descrivere lo stato dell'istanza per monitorarne l'integrità.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSComputeServiceRolePolicy](#) nel AWS Managed Policy Reference.

AmazonECSInstanceRolePolicyForManagedInstances

La policy AmazonECSInstanceRolePolicyForManagedInstances fornisce le autorizzazioni necessarie alle istanze gestite da Amazon ECS per registrarsi nei cluster Amazon ECS e comunicare con il servizio Amazon ECS.

Questa policy include le autorizzazioni che consentono alle istanze gestite da Amazon ECS di completare le attività seguenti:

- Effettuare e annullare la registrazione con i cluster Amazon ECS.
- Inviare le modifiche dello stato delle istanze di container.
- Inviare le modifiche allo stato dell'attività.
- Scoprire gli endpoint di polling per l'agente Amazon ECS.

Per visualizzare le autorizzazioni per questa politica, consulta [Amazon ECSInstanceRolePolicyForManagedInstances](#) nel AWS Managed Policy Reference.

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 autorizzazioni ad Amazon ECSServiceRolePolicy .	La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata per includere l'autorizzazione <code>ssmmessages:OpenDataChannel</code> . Questa autorizzazione consente ad Amazon ECS di aprire canali di dati per le sessioni ECS Exec.	20 gennaio 2026
Aggiorna la politica AmazonECSInfrastructureRolePolicyForManagedInstances	La AmazonECSInfrastructureRolePolicyForManagedInstances politica è stata aggiornata per modificare le CreateFleet autorizzazioni. Le condizioni basate sulle risorse per sottoreti, gruppi di sicurezza e EC2 immagini sono state rimosse perché: • Le sottoreti e i gruppi di sicurezza non vengono creati o gestiti da Amazon ECS, quindi non contengono i tag richiesti. AmazonECS Managed • Sebbene Amazon ECS possieda e gestisca EC2 le immagini, i tag non sono visibili ai clienti, impedendo l'applicazione di politiche basate sulle risorse.	15 dicembre 2025

Modifica	Descrizione	Data
	Questa modifica garantisce il corretto funzionamento della policy con risorse prive dei tag di gestione ECS previsti.	
Aggiungi autorizzazioni ad Amazon ECSServiceRolePolicy .	La policy IAM AmazonECS ServiceRolePolicy gestita è stata aggiornata con nuove EC2 autorizzazioni Amazon che consentono ad Amazon ECS di recuperare EC2 Amazon Event Windows per i servizi e i cluster associati a Event Windows.	20 novembre 2025
Aggiungi autorizzazioni ad Amazon ECSServiceRolePolicy .	La policy IAM AmazonECS ServiceRolePolicy gestita è stata aggiornata con nuove EC2 autorizzazioni Amazon che consentono ad Amazon ECS di effettuare il provisioning e il de-provisioning di Task ENI.	14 novembre 2025

Modifica	Descrizione	Data
Aggiorna la ECSInfrastructure RolePolicyForServiceConnect TransportLayerSecurity politica di Amazon	È stata aggiornata la policy IAM ECSInfrastructure RolePolicyForServiceConnect TransportLayerSecurity gestita da Amazon per separare secretsmanager: DescribeSecret permission in una propria dichiarazione di policy. L'autorizzazione continua a limitare l'accesso ad Amazon ECS esclusivamente ai segreti creati da Amazon ECS e utilizza il pattern matching ARN anziché i tag delle risorse per l'ambito. Ciò consente ad Amazon ECS di monitorare e lo stato segreto durante tutto il suo ciclo di vita, incluso quando un segreto è stato eliminato.	13 novembre 2025
Aggiunto nuovo the section called “AmazonECS InfrastructureRole forExpressGatewayServices ”	Aggiunta una nuova ECSInfrastructure RoleforExpressGatewayServices policy Amazon che fornisce l'accesso ad Amazon ECS per creare e gestire applicazioni Web utilizzando Express Services.	21 novembre 2025

Modifica	Descrizione	Data
Aggiunto nuovo the section called “AmazonECS InstanceRolePolicy ForManagedInstances”	È stata aggiunta una nuova ECSInstance RolePolicy yForManagedInstances policy di Amazon che fornisce le autorizzazioni per la registrazione delle istanze gestite di Amazon ECS con i cluster Amazon ECS.	30 settembre 2025
Aggiunto nuovo the section called “AmazonECS InfrastructureRole PolicyForManagedInstances”	Aggiunta una nuova ECSInfrastructure RolePolicy yForManagedInstances policy Amazon che fornisce l'accesso ad Amazon ECS per creare e gestire risorse EC2 gestite da Amazon.	30 settembre 2025
Aggiungi un nuovo Amazon ECSCompute ServiceRolePolicy	Permette ad Amazon ECS di gestire le istanze gestite da Amazon ECS e le relative risorse.	31 agosto 2025

Modifica	Descrizione	Data
Aggiunta di autorizzazioni a the section called “EC2ContainerServiceforEC2Role di Amazon”	La policy IAM gestita AmazonEC2ContainerServiceforEC2Role è stata aggiornata per includere l'autorizzazione <code>ecs:ListTagsForResource</code>. Questa autorizzazione consente all'agente Amazon ECS di recuperare i tag delle attività e delle istanze di container tramite l'endpoint dei metadati delle attività (<code>\${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags</code>).	4 agosto 2025
Aggiungi autorizzazioni ad Amazon ECSInfrastructureRolePolicyForLoadBalancers.	La policy IAM gestita AmazonECSInfrastructureRolePolicyForLoadBalancers è stata aggiornata con nuove autorizzazioni per la descrizione, l'annullamento della registrazione e la registrazione dei gruppi di destinazione.	25 luglio 2025
Aggiunta nuova policy the section called “AmazonECSInfrastructureRolePolicyForLoadBalancers”	Aggiunta una nuova ECSInfrastructureRolePolicyForLoadBalancers policy di Amazon che fornisce l'accesso ad altre risorse di AWS servizio necessarie per gestire i sistemi di bilanciamento del carico associati ai carichi di lavoro Amazon ECS.	15 luglio 2025

Modifica	Descrizione	Data
Aggiungi autorizzazioni ad Amazon ECSServiceRolePolicy.	La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata con nuove autorizzazioni AWS Cloud Map con cui Amazon ECS può aggiornare e gli attributi di servizio AWS Cloud Map per i servizi gestiti da Amazon ECS.	15 luglio 2025
Aggiungi autorizzazioni ad Amazon ECSServiceRolePolicy	La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata con nuove autorizzazioni AWS Cloud Map con cui Amazon ECS può aggiornare e gli attributi di servizio AWS Cloud Map per i servizi gestiti da Amazon ECS.	24 giugno 2025
Aggiungi autorizzazioni ad Amazon ECSInfrastructureRolePolicyForVolumes	La policy AmazonECSInfrastructureRolePolicyForVolumes è stata aggiornata per aggiungere l'autorizzazione <code>ec2:DescribeInstances</code>. L'autorizzazione aiuta a prevenire la collisione dei nomi dei dispositivi per i volumi Amazon EBS collegati alle attività di Amazon ECS eseguite sulla stessa istanza di container.	2 giugno 2025

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 policy AmazonECSInfrastructureRolePolicyForVolumes 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 policy AmazonECS_FullAccess è stata aggiornata per aggiungere le autorizzazioni iam:PassRole per i ruoli IAM per un ruolo denominato ecsInfrastructureRole. Questo è il ruolo IAM predefinito creato da Console di gestione AWS 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

Modifica	Descrizione	Data
Aggiungi una nuova ECSInfrastructure RolePolicyForServiceConnectTransport LayerSecurity politica Amazon	È stata aggiunta una nuova ECSInfrastructure RolePolicyForServiceConnectTransport LayerSecurity policy di 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 Autorità di certificazione privata	22 gennaio 2024
Aggiungi una nuova politica Amazon ECSInfrastructure RolePolicyForVolumes	La policy AmazonECSInfrastructureRolePolicyForVolumes 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 ECSServiceRolePolicy	La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata con nuove autorizzazioni events e autorizzazioni autoscaling e autoscaling-plans aggiuntive.	04 dicembre 2023

Modifica	Descrizione	Data
Aggiungi autorizzazioni ad Amazon EC2 Container ServiceEventsRole	La policy IAM AmazonECS ServiceRolePolicy gestita è stata aggiornata per consentire l'accesso al funzionamento dell' AWS Cloud Map DiscoverInstancesRevision API.	4 ottobre 2023
Aggiungi autorizzazioni ad Amazon EC2 Container Servicefor EC2 Role	La policy AmazonEC2 ContainerServiceforEC2Role è stata modificata per aggiungere l'autorizzazione <code>ecs:TagResource</code> , che include una condizione che limita l'autorizzazione solo ai cluster appena creati e alle istanze di container appena registrate.	6 marzo 2023
Aggiunta di autorizzazioni a the section called “Amazon ECS_FullAccess”	La policy AmazonECS_FullAccess è stata modificata per aggiungere l'autorizzazione <code>elasticloadbalancing:AddTags</code> che include una condizione che limita l'autorizzazione solo ai bilanciatori del carico appena creati, ai gruppi di destinazione, alle regole e agli ascoltatori creati. Questa autorizzazione non consente l'aggiunta di tag a risorse Elastic Load Balancing già create.	4 gennaio 2023

Modifica	Descrizione	Data
Amazon ECS ha cominciato a tenere traccia delle modifiche	Amazon ECS ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	08 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 sostituiti 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 si utilizzano attualmente le autorizzazioni concesse dalla policy `AmazonEC2ContainerServiceFullAccess` che non sono nella policy `AmazonECS_FullAccess`, è possibile aggiungerle a un'istruzione della policy 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 politica di `AmazonECS_ () FullAccess` Console di gestione AWS

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 sta correntemente utilizzando la policy `AmazonEC2ContainerServiceFullAccess`, seleziona il ruolo e completa la seguente procedura per scollegare la policy eliminata gradualmente e collegare la policy `AmazonECS_FullAccess`.
 - a. Nella scheda Autorizzazioni, scegli la X accanto alla EC2 ContainerServiceFullAccess politica di Amazon.
 - b. Scegli Add Permissions (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. Usa il comando [generate-service-last-accessed-details](#) per generare un report che includa i dettagli relativi all'ultimo utilizzo della policy eliminata gradualmente.

```
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 processo dell'output precedente con il comando [get-service-last-accessed-details](#) per recuperare l'ultimo report accessibile del servizio. Questo report visualizza il nome della risorsa Amazon (ARN) delle entità IAM che hanno utilizzato per l'ultima volta la policy eliminata gradualmente.

```
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 predefiniti da Amazon ECS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi per tuo conto.

Argomenti

- [Uso dei ruoli per consentire ad Amazon ECS di gestire i cluster](#)
- [Uso dei ruoli per gestire le istanze gestite da Amazon ECS](#)

Uso dei ruoli per consentire ad Amazon ECS di gestire i cluster

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 predefiniti da Amazon ECS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi 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 allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon ECS perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni su altri servizi che supportano i 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 `AWSServiceRoleForECS` — Role per consentire ad Amazon ECS di gestire il cluster.

Il ruolo collegato al servizio `AWSService RoleFor 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 bilanciatore del carico con il servizio Amazon ECS, Amazon ECS gestisce la registrazione e l'annullamento della registrazione delle risorse con il bilanciatore del carico.
- Operazione: quando si utilizza Amazon ECS Service Discovery, Amazon ECS gestisce le risorse `Route 53` e `AWS Cloud Map` richieste per far funzionare il rilevamento servizi.
- 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
- **Azione:** Amazon ECS può aggiornare gli attributi AWS Cloud Map del servizio per i servizi gestiti da Amazon ECS.
- **Azione:** Amazon ECS può richiamare Amazon EC2 provisioning e de-provisioning ENI all'avvio e all'interruzione delle attività.
- **Azione:** Amazon ECS può recuperare Amazon EC2 Event Windows per i servizi e i cluster associati a Event Windows.

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente IAM.

Creazione di un ruolo collegato ai servizi per Amazon ECS

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un cluster o crei o aggiorni un servizio nella Console di gestione AWS, o nell' AWS API AWS CLI, Amazon ECS crea il ruolo collegato al servizio per te.

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 utilizzavi il servizio Amazon ECS prima del 1° gennaio 2017, quando ha iniziato a supportare ruoli collegati al servizio, Amazon ECS ha creato il ruolo ECS nel tuo AWSService RoleFor account. Per ulteriori informazioni, consulta A [new role appeared in my](#). Account AWS

Puoi anche utilizzare la console IAM per creare un ruolo collegato al servizio con lo use case AWSServiceRoleForECS. Nell'API AWS CLI o nell' AWS API, usa IAM per creare un ruolo collegato al servizio con il nome del servizio. `ecs.amazonaws.com` Per ulteriori informazioni, consulta [Creazione di un ruolo collegato ai servizi](#) nella Guida per l'utente IAM. Se elimini il ruolo collegato ai servizi, è possibile utilizzare lo stesso processo per crearlo nuovamente.

Se elimini questo ruolo collegato al servizio, è possibile 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 avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon ECS

Non è necessario eliminare manualmente il ruolo ECS. AWSService RoleFor Quando elimini i cluster in tutte le regioni dell'API Console di gestione AWS, dell'API AWS CLI, Amazon ECS pulisce le risorse ed elimina il ruolo collegato al servizio per te.

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 ai servizi prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di utilizzare IAM; per eliminare un ruolo collegato al servizio, è necessario prima rimuovere qualsiasi risorsa utilizzata dal ruolo.

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 eliminare le risorse Amazon ECS utilizzate da AWSService RoleFor ECS (console)

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](#) e [Eliminazione di un servizio Amazon ECS utilizzando 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 [Eliminare un cluster Amazon ECS](#).

Per eliminare le risorse Amazon ECS utilizzate da AWSService RoleFor ECS (AWS CLI)

1. Dimensione a 0 il conteggio di tutti i servizi Amazon ECS in tutte le regioni, quindi elimina i servizi. [Per ulteriori informazioni, consulta update-service e delete-service nel Reference.](#) AWS Command Line Interface
2. Forza l'annullamento della registrazione di tutte le istanze di container da tutti i cluster in tutte le regioni. Per ulteriori informazioni, consulta [deregister-container-instance](#).
3. Elimina tutti i cluster Amazon ECS in tutte le regioni. Per ulteriori informazioni, consulta [delete-cluster](#).

Per eliminare le risorse Amazon ECS utilizzate dall' AWSServiceRoleForECS (API)

1. Dimensione a 0 il conteggio di tutti i servizi Amazon ECS in tutte le regioni, quindi elimina i servizi. Per ulteriori informazioni, consulta [UpdateService](#) e [DeleteService](#) consulta Amazon ECS API Reference.
2. Forza l'annullamento della registrazione di tutte le istanze di container da tutti i cluster in tutte le regioni. Per ulteriori informazioni, consulta [DeregisterContainerInstance](#).
3. Elimina tutti i cluster Amazon ECS in tutte le regioni. Per ulteriori informazioni, consulta [DeleteCluster](#).

Eliminazione manuale del ruolo collegato ai servizi

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 maggiori informazioni, consulta [Regioni ed endpoint di AWS](#).

Uso dei ruoli per gestire le istanze gestite da 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 predefiniti da Amazon ECS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri AWS servizi 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 allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. Questa procedura protegge le risorse di Amazon ECS perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni su altri servizi che supportano i 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 `AWSServiceRoleForECSCompute—Role` per consentire ad Amazon ECS di gestire le istanze gestite di Amazon, fornito dal provider di capacità Amazon ECS EC2 Managed Instances.

Il ruolo `AWSServiceRoleForECSCompute` collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `ecs-compute.amazonaws.com`

La policy sulle autorizzazioni dei ruoli denominata [AmazonECSComputeServiceRolePolicy](#) consente ad Amazon ECS di completare le seguenti attività:

- Amazon ECS può descrivere ed eliminare i modelli di avvio.
- Amazon ECS può descrivere ed eliminare le versioni dei modelli di avvio.
- Amazon ECS può terminare le istanze.
- Amazon ECS può descrivere i seguenti parametri dei dati delle istanze:
 - Istanza
 - Interfacce di rete delle istanze: Amazon ECS può descrivere come gestire il ciclo di vita dell' EC2istanza.
 - Finestra di eventi dell'istanza: Amazon ECS può descrivere le informazioni sulla finestra di eventi per determinare se il flusso di lavoro può essere interrotto per l'applicazione di patch all'istanza.
 - Stato dell'istanza: Amazon ECS può descrivere lo stato dell'istanza per monitorarne l'integrità.

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta [Autorizzazioni del ruolo collegato ai servizi](#) nella Guida per l'utente IAM.

Creazione di un ruolo collegato ai servizi per Amazon ECS

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un provider di capacità per Amazon ECS Managed Instances nella Console di gestione AWS, nella o nell' AWS API AWS CLI, Amazon ECS crea il ruolo collegato al servizio per te.

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 utilizzavi il servizio Amazon ECS prima del 1° gennaio 2017, quando ha iniziato a supportare ruoli collegati ai servizi, Amazon ECS ha creato il ECSCCompute ServiceRolePolicy ruolo Amazon nel tuo account. Per ulteriori informazioni, consulta [A new role appeared in my Account AWS](#)

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un provider di capacità per le istanze gestite da Amazon ECS, Amazon ECS crea di nuovo automaticamente il ruolo collegato al servizio per te.

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 ECSCompute ServiceRolePolicy servizio Amazon. Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta [Modifica di un ruolo collegato al servizio](#) nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon ECS

Non è necessario eliminare manualmente il ECSCompute ServiceRolePolicy ruolo Amazon. Quando elimini tutti i provider di capacità di Amazon ECS Managed Instances in tutte le regioni dell' AWS API AWS CLI, Amazon ECS pulisce le risorse ed elimina il ruolo collegato al servizio per te. Console di gestione AWS

Eliminazione manuale del ruolo collegato ai servizi

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo ECSCompute ServiceRolePolicy collegato al servizio Amazon. 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 maggiori 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 container o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità usate. Usa la seguente tabella per determinare i ruoli IAM necessari per Amazon ECS.

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri	La tua attività è ospitata su AWS	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
	AWS servizi per tuo conto.	Fargate o un'istanza esterna e: • estrae un'immagine del contenitore da un repository privato Amazon ECR. • estrae un'immagine di container da un repository privato Amazon ECR in un account diverso da quello 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: • usa l'autenticazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzan	

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		do 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 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 dell'infrastruttura Amazon ECS per i bilanciatori del carico	Questo ruolo consente ad Amazon ECS di gestire le risorse di bilanciamento del carico nei tuoi cluster per tuo conto per le distribuzioni. blue/green	Vuoi usare le blue/green distribuzioni di Amazon ECS.	Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
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 permette ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• È preferibile usare le istanze gestite da Amazon ECS per la tua capacità. • Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo dell'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 servizi Amazon ECS Service Connect. • Desideri creare gruppi di destinazione di VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS
Profilo dell'istanza	Questo ruolo consente alle istanze gestite da Amazon ECS di assumere il ruolo dell'infrastruttura in modo sicuro.	Usi le istanze gestite da Amazon ECS nei tuoi cluster.	Profilo delle istanze gestite da Amazon ECS

Best practice 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 usate. Ti suggeriamo di creare ruoli separati nella tabella invece di condividerli.

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 Fargate o un'istanza esterna e: - estrae un'immagine del contenitore da un repository privato Amazon ECR. - estrae un'immagine di container da un repository privato Amazon ECR in un account diverso da quello 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: - usa l'autenticazione del registro privato.	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		- 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 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	Definizione	Quando richiesto	Ulteriori informazioni
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 dell'infrastruttura a Amazon ECS	Questo ruolo permette 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 dell'infrastruttura a 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 servizi Amazon ECS Service Connect. • Desideri creare gruppi di destinazione di VPC Lattice.	Ruolo IAM dell'infrastruttura Amazon ECS

Ruolo del processo

Consigliamo di assegnare un ruolo a un'attività. 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 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 `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` ambiente quando AWS effettuano chiamate API. [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 Gestione dei segreti AWS, 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 di Amazon ECS da un cluster. L'agente del container di Amazon ECS non chiama questa operazione API, ma questa autorizzazione rimane per aiutare a garantire la compatibilità con le versioni precedenti.
- `ecs:DiscoverPollEndpoint`: (necessario) questa operazione restituisce gli endpoint utilizzati dall'agente del container di Amazon ECS per eseguire il polling per gli aggiornamenti.
- `ecs:Poll`: (necessario) consente all'agente del container di Amazon ECS di comunicare con il piano di controllo Amazon ECS per segnalare le modifiche dello stato dell'attività.
- `ecs:RegisterContainerInstance`: (necessario) consente a un principale di annullare la registrazione di un'istanza di container con 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 di container Amazon ECS di comunicare con il piano di controllo Amazon ECS per segnalare le informazioni di integrità e i parametri per ogni container e attività.

Sebbene questa autorizzazione non sia necessaria, ti consigliamo di aggiungerla per permettere alle metriche dell'istanza di container di avviare azioni di dimensionamento e ricevere anche report relativi ai comandi di controllo dell'integrità.

- `ecs:TagResource`: (facoltativo) consente all'agente di container Amazon ECS di assegnare tag al cluster al momento della creazione e di assegnare tag alle istanze di container quando vengono 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*`: (necessario) include le operazioni API `SubmitAttachmentStateChanges`, `SubmitContainerStateChange` e `SubmitTaskStateChange`. 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'autorizzazione `SubmitContainerStateChange` non è più utilizzata dall'agente del container di Amazon ECS, ma rimane per aiutare 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 si esegue il push di un'immagine di container in un repository privato Amazon ECR, ogni livello dell'immagine viene controllato per verificare se ne è già stato eseguito il push. Se lo è, il livello dell'immagine viene ignorato.
- `ecr:GetDownloadUrlForLayer`: (facoltativo) quando un'immagine del container viene estratta da un repository privato Amazon ECR, questa API viene richiamata una volta per ogni livello di immagine che non è già memorizzato nella cache.
- `ecr:BatchGetImage`: (facoltativo) quando si esegue il pull di un'immagine di container 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 log in un flusso di log specificato.

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](#).

Suggerimenti 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.

Uso della modalità di rete **awsvpc**

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.

Usa le informazioni sull'ultimo accesso per perfezionare 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 tal fine, rivedi le informazioni sull'ultimo accesso fornite dall'IAM, quindi rimuovi le azioni che non sono mai state usate o che non sono state usate 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 AWS CloudTrail attività 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": "AR0A36C6WWEJ2YEXAMPLE: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 vengono rese disponibili all'agente in esecuzione sull'istanza tramite l'invio periodico da parte di Amazon ECS delle credenziali temporanee del ruolo, ma non sono direttamente accessibili dai container nell'attività. Per le autorizzazioni IAM necessarie all'esecuzione del codice dell'applicazione all'interno del container, consulta [Ruolo IAM dell'attività Amazon ECS](#).

Di seguito sono elencati i casi di utilizzo comune per un ruolo IAM di esecuzione di attività:

- La tua attività è ospitata su AWS FargateAmazon ECS Managed Instances o su un'istanza esterna e:
 - estrae un'immagine del contenitore da un repository privato Amazon ECR.
 - estrae un'immagine di container da un repository privato Amazon ECR in un account diverso da quello 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:
 - usa 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 per 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 `AmazonECSTaskExecutionRolePolicy` che contiene le autorizzazioni richieste dai casi di utilizzo comune descritti in precedenza. Per ulteriori informazioni, consulta [Amazon ECSTask ExecutionRolePolicy](#) nella AWS Managed Policy Reference Guide. Potrebbe essere necessario aggiungere policy in linea al ruolo di esecuzione delle attività per i casi d'uso speciali

La console Amazon ECS crea un ruolo di esecuzione delle attività. Puoi collegare manualmente la policy IAM gestita per le attività in modo da consentire ad Amazon ECS di aggiungere le autorizzazioni per funzionalità e miglioramenti futuri man mano che vengono introdotti. Puoi usare la ricerca della console IAM per cercare `ecsTaskExecutionRole` e controllare se l'account dispone già di un ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta [Ricerca nella console IAM](#) nella Guida per l'utente IAM.

Se estrai immagini in qualità di utente autenticato, è meno probabile che tu subisca l'impatto delle modifiche apportate [Docker Hub all'utilizzo e ai limiti di](#). 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 l'account non dispone già di un ruolo di esecuzione di attività, attenersi alla seguente procedura per creare il ruolo.

Console di gestione AWS

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi Console di gestione AWS 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 Tipo di entità attendibile, seleziona Servizio AWS.
4. Per Servizio o caso d'uso, seleziona Elastic Container Service e poi il caso d'uso Attività Elastic Container Service.
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:

JSON

```
{
  "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 ad esso ulteriori autorizzazioni per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Estrai le immagini dei contenitori da registri privati esterni AWS (come Docker Hub, Quay.io)	Autorizzazioni di autenticazione del registro privato

Funzionalità	Autorizzazioni aggiuntive
o il tuo registro privato) utilizzando le credenziali di Secrets Manager	
Trasmetti dati sensibili con Systems Manager o Secrets Manager	Autorizzazioni per Secrets Manager o Systems Manager
Fai in modo che le attività Fargate eseguano il pull delle immagini Amazon ECR su endpoint di interfaccia	Autorizzazioni per attività Fargate che eseguono il pull delle immagini Amazon ECR su endpoint di interfaccia
File di configurazione di host in un bucket Amazon S3	Autorizzazioni per l'archiviazione dei file Amazon S3
Configura Approfondimenti sui container per visualizzare gli eventi del ciclo di vita di Amazon ECS	Autorizzazioni necessarie per abilitare gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container
Visualizza gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container	Autorizzazioni necessarie per visualizzare gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container

Autorizzazioni di autenticazione del registro privato

L'autenticazione del registro privato consente alle attività di Amazon ECS di estrarre immagini di container da registri privati esterni AWS (come Docker Hub, Quay.io o il tuo registro privato) che richiedono credenziali di autenticazione. Questa funzionalità utilizza Secrets Manager per archiviare in modo sicuro le credenziali del tuo registro, a cui viene poi fatto riferimento nella definizione dell'attività utilizzando il parametro `repositoryCredentials`.

Per ulteriori informazioni sulla configurazione dell'autenticazione del registro privato, consulta [Utilizzo di immagini non AWS container in 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 contengono credenziali del tuo registro privato. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- `secretsmanager:GetSecretValue`: necessario per recuperare le credenziali del registro privato da Secrets Manager.
- `kms:Decrypt`: obbligatorio solo se il segreto 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret_name",
        "arn:aws:kms:us-east-1:111122223333:key/key_id"
      ]
    }
  ]
}
```

Autorizzazioni per Secrets Manager o Systems Manager

L'autorizzazione per consentire all'agente contenitore di estrarre le risorse necessarie AWS Systems Manager o Secrets Manager. Per ulteriori informazioni, consulta [Trasferimento di dati sensibili a un container 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.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:us-east-1:111122223333: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:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:us-east-1:111122223333:parameter/parameter_name",
        "arn:aws:secretsmanager:us-east-1:111122223333:secret:secret_name",
        "arn:aws:kms:us-east-1:111122223333:key/key_id"
      ]
    }
  ]
}
```

Autorizzazioni per attività Fargate che eseguono il pull delle immagini Amazon ECR su endpoint di interfaccia

Quando avvii attività che usano 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. Puoi 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:

JSON

```
{
  "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": "arn:aws:ecr:*:*:repository/*",
    }
  ]
}
```

```

        "Condition": {
            "StringEquals": {
                "aws:sourceVpce": "vpce-0123456789abcdef0"
            }
        }
    }
}

```

Autorizzazioni Amazon ECR

Le seguenti autorizzazioni sono necessarie quando è necessario eseguire il pull delle immagini di container dai repository privati di Amazon ECR. Il ruolo di esecuzione dell'attività deve disporre di queste autorizzazioni per consentire al container Amazon ECS e agli agenti Fargate di eseguire il pull delle immagini del container per tuo conto. Per le implementazioni ECS di base, queste autorizzazioni devono essere aggiunte al ruolo di esecuzione dell'attività invece che al ruolo IAM dell'attività.

La policy gestita del ruolo di esecuzione delle attività di Amazon ECS (AmazonECSTaskExecutionRolePolicy) comprende le autorizzazioni necessarie per eseguire il pull delle immagini da Amazon ECR. Se usi la policy gestita, non è necessario aggiungere queste autorizzazioni separatamente.

Se stai creando una policy personalizzata, includi le seguenti autorizzazioni per consentire il pull delle immagini da Amazon ECR:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
      "Resource": "*"
    }
  ]
}

```

```
}
```

Tieni presente che queste autorizzazioni sono diverse dalle autorizzazioni che potrebbero essere richieste nel ruolo IAM dell'attività se il codice dell'applicazione deve interagire direttamente con Amazon APIs ECR. Per ulteriori informazioni sulle autorizzazioni Amazon ECR per un ruolo IAM dell'attività, consulta [Autorizzazioni Amazon ECR](#).

Autorizzazioni per l'archiviazione dei file Amazon S3

Quando specifichi un file di configurazione ospitato in Amazon S3, il ruolo di esecuzione dell'attività deve includere l'autorizzazione `s3:GetObject` per il file di configurazione e l'autorizzazione `s3:GetBucketLocation` per il bucket Amazon S3 in cui si trova il file. Per maggiori informazioni, consulta [Azioni di policy per Amazon S3](#) nella Guida per l'utente di Amazon Simple Storage Service.

La policy di esempio seguente aggiunge le autorizzazioni necessarie per recuperare un file da Amazon S3. Specifica il nome del bucket Amazon S3 e il nome del file di configurazione.

JSON

```
{
  "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"
      ]
    }
  ]
}
```

```
}
```

Considerazioni importanti sulla sicurezza

Quando utilizzi le funzionalità di Amazon ECS che si integrano con i bucket Amazon S3, implementa la convalida della proprietà dei bucket per prevenire attacchi di acquisizione dei bucket. Senza un'adeguata convalida, se un bucket Amazon S3 viene eliminato e ricreato da un malintenzionato con lo stesso nome, le tue attività potrebbero caricare inconsapevolmente configurazioni dannose o inviare dati sensibili a bucket controllati dagli aggressori.

Condizione di policy IAM consigliata:

```
"Condition": {
  "StringEquals": {
    "aws:ResourceAccount": "TRUSTED-ACCOUNT-ID"
  }
}
```

Sostituiscilo *TRUSTED-ACCOUNT-ID* con l'ID AWS dell'account proprietario del bucket S3.

Questa condizione garantisce che il ruolo di esecuzione dell'attività possa accedere solo ai bucket Amazon S3 di proprietà dell'account affidabile specificato.

Ruolo IAM dell'attività Amazon ECS

Le attività Amazon ECS possono avere un ruolo IAM associato. Le autorizzazioni concesse nel ruolo IAM sono vendute ai container in esecuzione nell'attività. Questo ruolo consente al codice dell'applicazione (in esecuzione nel contenitore) di utilizzare altri AWS servizi. Il ruolo dell'attività è necessario quando l'applicazione accede ad altri AWS servizi, come Amazon S3.

Note

Queste autorizzazioni non sono accessibili dal container Amazon ECS e dagli 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 delle attività:

- Separazione delle preoccupazioni: se utilizzi EC2, i ruoli task IAM ti consentono di specificare le autorizzazioni IAM per i tuoi contenitori senza richiedere che tali autorizzazioni vengano specificate utilizzando i profili di EC2 istanza (per ulteriori informazioni, consulta [Using instance profiles](#) nella AWS Identity and Access Management User Guide). Pertanto, puoi distribuire le tue applicazioni in modo indipendente e uniforme sulle istanze di container ECS senza dover modificare le autorizzazioni IAM associate alle istanze. EC2
- Verificabilità: la registrazione degli accessi e degli eventi sono disponibili per garantire un controllo retrospettivo. CloudTrail Le credenziali dell'attività hanno il contesto 'taskArn' allegato alla sessione, pertanto i CloudTrail registri mostrano per quale attività sono state fornite le credenziali del ruolo.
- Consegna uniforme delle credenziali: ECS fornisce le credenziali dei ruoli IAM ai tuoi container e le rende accessibili tramite un'interfaccia ben definita, indipendentemente dall'opzione di elaborazione associata alle tue attività. In ECS Fargate EC2 , i profili di istanza non sono disponibili per i contenitori delle tue attività. I ruoli Task IAM ti consentono di associare le autorizzazioni IAM ai tuoi contenitori indipendentemente dall'opzione di calcolo quando utilizzi AWS SDK o AWS CLI nei tuoi contenitori. [Per ulteriori informazioni su come l' AWS SDK accede a queste credenziali, consulta Container credential provider.](#)

Important

I container non costituiscono un limite di sicurezza e l'uso dei ruoli IAM delle attività non cambia questa situazione. Ogni attività in esecuzione su 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'altra attività. Per EC2 le istanze di container esterne su ECS, non esiste alcun isolamento delle attività (a differenza di Fargate) e i contenitori possono potenzialmente accedere alle credenziali per altre attività sulla stessa istanza di contenitore. Possono anche accedere alle autorizzazioni assegnate al [ruolo dell'istanza di container ECS](#). Segui i consigli contenuti nelle raccomandazioni [relative ai ruoli](#) per bloccare l'accesso ai metadati di Amazon EC2 Instance per i contenitori (per ulteriori informazioni, consulta [Use the Instance Metadata Service per accedere ai metadati delle istanze](#) nella Amazon EC2 User Guide).

Tieni presente che quando specifichi un ruolo IAM per un'attività, la AWS CLI o altro SDKs nei contenitori per quell'attività utilizzano esclusivamente le AWS credenziali fornite dal ruolo dell'attività e non ereditano alcuna autorizzazione IAM da EC2 Amazon o dall'istanza esterna su cui sono in esecuzione.

Creazione del ruolo IAM dell'attività

Quando crei una policy IAM da utilizzare per le tue attività, essa deve includere le autorizzazioni che vuoi assegnare ai container nelle 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 [Creating IAM policies](#) 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 attendibilità di un ruolo](#) nella Guida per l'utente di 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. Potrai quindi allegare al ruolo la policy IAM specifica per l'assegnazione delle autorizzazioni desiderate ai container nella tua attività. 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 [Endpoint di servizio](#) nella Guida di 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 crei il tuo ruolo IAM per attività, ti consigliamo di utilizzare le chiavi di condizione `aws:SourceAccount` o `aws:SourceArn` nella policy di relazione di fiducia associata al ruolo per limitare ulteriormente le autorizzazioni e prevenire il problema di sicurezza `confused deputy`. 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 deputato confuso e su come proteggere il tuo AWS account, consulta [il problema del vice confuso nella Guida](#) per l'utente IAM.

JSON

```
{
  "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"
        }
      }
    }
  ]
}
```

Usa la seguente procedura per creare una policy per il recupero di oggetti da Amazon S3 con una policy di esempio. Sostituisci tutti i *user input* con i valori in tuo possesso.

Console di gestione AWS

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi Console di gestione AWS 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:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::my-task-secrets-bucket/*"
      ]
    }
  ]
}
```

2. Usa il seguente comando per creare la policy IAM utilizzando il file del 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
```

Per creare il ruolo di servizio, usa la seguente procedura.

Console di gestione AWS

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi Console di gestione AWS 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 Tipo di entità attendibile, seleziona Servizio AWS.
4. Per Servizio o caso d'uso, seleziona Elastic Container Service e poi il caso d'uso Attività Elastic Container Service.
5. Scegli Next (Successivo).
6. Per Aggiungi permessi, cerca e seleziona 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` contenente la policy di attendibilità da utilizzare per il ruolo IAM dell'attività. Il file deve contenere il testo seguente. Sostituisci l'identificatore della regione e specifica il numero di AWS account che usi per avviare le attività.

JSON

```
{
  "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"
      }
    }
  }
]
}

```

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 usando il seguente comando.
taskRolePolicy Sostituiscilo con il nome della politica che hai creato.

```

aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`taskRolePolicy`].Arn'

```

4. Allega la policy IAM che hai creato al ruolo `ecsTaskRole`. 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 ad esso ulteriori autorizzazioni per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Usare l'ECS Exec	Autorizzazioni ACS Exec
Utilizza un'immagine da un repository Amazon ECR privato	Autorizzazioni Amazon ECR

Funzionalità	Autorizzazioni aggiuntive
Usa EC2 istanze (Windows e Linux)	Configurazione aggiuntiva EC2 delle istanze Amazon
Usa istanze esterne	Configurazione dell'istanza esterna aggiuntiva
Usa istanze Windows EC2	Configurazione aggiuntiva dell'istanza Amazon EC2 Windows

Autorizzazioni Amazon ECR

Le autorizzazioni seguenti sono necessarie quando il codice dell'applicazione deve interagire direttamente con i repository Amazon ECR. Nota che per l'implementazione di base in cui è sufficiente estrarre immagini da Amazon ECR, queste autorizzazioni non sono richieste a livello di ruolo IAM dell'attività. Al contrario, il ruolo di esecuzione delle attività di Amazon ECS dovrebbe avere queste autorizzazioni. Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta [Ruolo IAM di esecuzione di attività Amazon ECS](#).

Se il codice dell'applicazione in esecuzione nel contenitore deve interagire APIs direttamente con Amazon ECR, è necessario aggiungere le seguenti autorizzazioni a un ruolo IAM dell'attività e includere il ruolo IAM dell'attività nella definizione dell'attività. Per informazioni, consulta [Aggiunta e rimozione delle policy IAM](#) nella Guida per l'utente di IAM.

Usa la seguente policy per il ruolo IAM dell'attività e aggiungi le autorizzazioni Amazon ECR richieste per le applicazioni container che devono interagire direttamente con Amazon ECR di:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:BatchGetImage",
        "ecr:GetDownloadUrlForLayer",
        "ecr:GetAuthorizationToken"
      ],
    }
  ],
}
```



```

        "Resource": "*"
    }
]
}

```

Autorizzazioni ACS Exec

La funzionalità [ECS Exec](#) richiede un ruolo IAM dell'attività per concedere ai container le autorizzazioni necessarie per la comunicazione tra SSM Agent 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 informazioni, consulta [Aggiunta e rimozione delle policy IAM](#) nella Guida per l'utente di IAM.

Utilizza la seguente policy per il ruolo IAM del processo e aggiungere le autorizzazioni di SSM richieste.

JSON

```

{
  "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 usi 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 istanze di container, aggiungi l'opzione `--net=host` al comando `docker run` 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 AWS credenziali:

```
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 `awsipc` 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_AWSIPC_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 `awsipc`. 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 dell'istanza esterna aggiuntiva

Le istanze esterne richiedono l'agente del container almeno alla versione `1.11.0` per utilizzare i ruoli IAM dell'attività; tuttavia, 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 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 istanze di container, aggiungi l'opzione `--net=host` al comando `docker run` 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 \(per prodotti non ottimizzati per Amazon ECS\) AMLs](#). 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 AWS credenziali:

```
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 del container 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 bilanciatore del carico. 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 del container 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](#).

Crea il ruolo dell'istanza di container

 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. Usa la seguente procedura per collegare la policy IAM gestita, se necessario.

## Console di gestione AWS

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi Console di gestione AWS 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 Tipo di entità attendibile, seleziona 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 Console di gestione AWS di to create 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. Usa il seguente comando per creare il ruolo IAM dell'istanza utilizzando il documento della policy di attendibilità.

```
aws iam create-role \
 --role-name ecsInstanceRole \
```



```
--assume-role-policy-document file://instance-role-trust-policy.json
```

3. Create un profilo di istanza denominato `ecsInstanceRole-profile` utilizzando il [create-instance-profile](#) comando.

```
aws iam create-instance-profile --instance-profile-name ecsInstanceRole-profile
```

Esempio di risposta

```
{
 "InstanceProfile": {
 "InstanceId": "AIPAJTLPJLEGREXAMPLE",
 "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. Collega la policy gestita da AmazonEC2ContainerServiceForEC2Role al ruolo usando 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 ad esso ulteriori autorizzazioni per le seguenti funzionalità.

| Funzionalità                           | Autorizzazioni aggiuntive                 |
|----------------------------------------|-------------------------------------------|
| Amazon ECR ha l'immagine del container | <a href="#">Autorizzazioni Amazon ECR</a> |

| Funzionalità                                                        | Autorizzazioni aggiuntive                                                    |
|---------------------------------------------------------------------|------------------------------------------------------------------------------|
| Fai in modo che CloudWatch Logs monitori le istanze del contenitore | <a href="#">Monitoraggio delle autorizzazioni delle istanze di container</a> |
| File di configurazione di host in un bucket Amazon S3               | <a href="#">Accesso in sola lettura ad Amazon S3</a>                         |

## Autorizzazioni Amazon ECR

Il ruolo dell'istanza di container di Amazon ECS utilizzato con le istanze di container deve possedere le seguenti autorizzazioni di policy IAM per Amazon ECR.

### JSON

```
{
 "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.

## Autorizzazioni necessarie per configurare l'impostazione dell'account `awsvpctrunking`

Amazon ECS supporta il lancio di istanze di container con una maggiore densità ENI utilizzando i tipi di istanze Amazon EC2 supportati. Se usi questa funzionalità, consigliamo di creare 2 ruoli dell'istanza di container. Abilita l'impostazione dell'account `awsvpctrunking` per un ruolo e

usare quel ruolo per attività che richiedono il trunking ENI. Per informazioni sulle impostazioni dell'account `awsvpcTrunking`, consulta [Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account](#).

Il ruolo dell'istanza di container utilizzato con le istanze di container deve possedere le seguenti autorizzazioni di policy IAM per configurare le impostazioni dell'account

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:ListAccountSettings",
 "ecs:ListAttributes",
 "ecs:PutAccountSetting"
],
 "Resource": "*"
 }
]
}
```

Per usare il ruolo dell'istanza di container, aggiungi quanto segue ai dati utente dell'istanza:

```
#!/bin/bash
aws ecs put-account-setting --name awsvpcTrunking --value enabled --region region
ECS_CLUSTER=MyCluster>> /etc/ecs/ecs.config
EOF
```

Per ulteriori informazioni sull'aggiunta di dati utente alle tue EC2 istanze, consulta [Esegui comandi sulla tua istanza Linux all'avvio](#) nella Amazon EC2 User Guide.

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 [Archiviazione della configurazione di un'istanza di container 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 usare la console IAM per aggiungere l'accesso in sola lettura ad Amazon S3 (AmazonS3ReadOnlyAccess) al tuo ruolo. Per ulteriori informazioni, consulta [Aggiornamento delle autorizzazioni per un ruolo](#) nella Guida per l'utente di AWS Identity and Access Management .

## Monitoraggio delle autorizzazioni delle istanze di 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, collegala a `ecsInstanceRole`.

## Console di gestione AWS

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi Console di gestione AWS 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 avere creato la policy, collegala al ruolo dell'istanza di container. Per informazioni su come associare la policy al ruolo, consulta [Aggiornamento delle autorizzazioni per un ruolo](#) nella Guida per l'utente di AWS Identity and Access Management

## 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 seguente comando 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 usando il seguente comando. Sostituiscila *cwlogspolicy* con il nome della policy che hai creato.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cwlogspolicy`].Arn'
```

4. Usa il seguente comando per collegare la policy al ruolo IAM dell'istanza di container 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 Console di gestione AWS. Puoi usare la ricerca della console IAM per cercare `ecsAnywhereRole` e controllare se l'account dispone già di un ruolo. Per ulteriori informazioni, consulta [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, `AmazonEC2ContainerServiceforEC2Role` e `AmazonSSMManagedInstanceCore` e le policy. 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 ruolo `ECSAnywhereRole` associato alla tua istanza esterna include anche le autorizzazioni necessarie per estrarre le immagini da Amazon ECR, allora il ruolo di esecuzione dell'attività non dovrà includerle.

### Creazione del ruolo Amazon ECS Anywhere

Sostituisci tutto *user input* con le tue informazioni.

1. Creare un file locale denominato `ssm-trust-policy.json` con la seguente policy di attendibilità:

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": {
 "Effect": "Allow",
 "Principal": { "Service": [
 "ssm.amazonaws.com"
] },
 "Action": "sts:AssumeRole"
 }
}
```

2. Crea il ruolo e allega la policy di attendibilità utilizzando il seguente comando AWS CLI .

```
aws iam create-role --role-name ecsAnywhereRole --assume-role-policy-document
file://ssm-trust-policy.json
```

3. Allega le politiche AWS gestite utilizzando il seguente comando.

```
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 inoltre usare il flusso di lavoro della policy di fiducia personalizzata IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando criteri di attendibilità personalizzati \(console\)](#) nella Guida per l'utente di 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 te e viene usato quando:

- Vuoi allegare volumi Amazon EBS alle tue attività Fargate EC2 o avviare Amazon ECS. Il ruolo dell'infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività.

Puoi usare le policy gestite `AmazonECSInfrastructureRolePolicyForVolumes`.



- Desideri utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i servizi Amazon ECS Service Connect.

Puoi usare le policy gestite

`AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity`.

- È preferibile creare gruppi di destinazione di Amazon VPC Lattice.

Puoi usare le policy gestite `AmazonECSInfrastructureRolePolicyForVpcLattice`.

- È preferibile le istanze gestite da Amazon ECS nei tuoi cluster Amazon ECS. Il ruolo dell'infrastruttura consente ad Amazon ECS di gestire il ciclo di vita delle istanze gestite.

Puoi usare le policy gestite `AmazonECSInfrastructureRolePolicyForManagedInstances`.

- Vuoi usare la modalità Express. Il ruolo di infrastruttura consente ad Amazon ECS di fornire e gestire i componenti dell'infrastruttura necessari per i servizi in modalità Express, tra cui bilanciamento del carico, gruppi di sicurezza, certificati SSL e configurazioni di scalabilità automatica.

Puoi usare le policy gestite `AmazonECSInfrastructureRoleforExpressGatewayServices`.

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`. Se il ruolo viene utilizzato per gestire Amazon ECS Managed Instances, il CloudTrail log `roleSessionName` sarà `ECSManagedInstancesForCompute`. Puoi usare questo nome per cercare eventi nella CloudTrail console filtrando per nome utente.

Amazon ECS fornisce policy gestite che contengono le autorizzazioni necessarie per il collegamento del volume, TLS, VPC Lattice e le istanze gestite. Per ulteriori informazioni, consulta [Amazon ECSInfrastructure RolePolicyForVolumes](#), [Amazon ECSInfrastructure RolePolicyForServiceConnectTransportLayerSecurity](#), `AmazonECSInfrastructureRolePolicyForVpcLattice`, [Amazon](#) e [Amazon ECSInfrastructure RolePolicyForManagedInstances](#) [ECSInfrastructure RoleforExpressGatewayServices](#) 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:

JSON

```
{
 "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, allega la policy gestita al `ecsInfrastructureRole` ruolo.
  - Per allegare volumi Amazon EBS alle tue attività Fargate EC2 o digita Amazon ECS di avvio, `AmazonECSInfrastructureRolePolicyForVolumes` allega la policy gestita.
  - Per utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect, allega la `AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity` policy gestita.
  - Per creare gruppi target Amazon VPC Lattice, allega la `AmazonECSInfrastructureRolePolicyForVpcLattice` policy gestita.

- Se desideri utilizzare Amazon ECS Managed Instances nei tuoi cluster Amazon ECS, allega la policy gestita. `AmazonECSInfrastructureRolePolicyForManagedInstances`

```
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
```

```
aws iam attach-role-policy \
 --role-name ecsInfrastructureRole \
 --policy-arn arn:aws:iam::aws:policy/
AmazonECSInfrastructureRolePolicyForManagedInstances
```

Puoi inoltre usare il flusso di lavoro della policy di attendibilità della console IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando criteri di attendibilità personalizzati \(console\)](#) nella Guida per l'utente di IAM.

#### Important

Se il ruolo dell'infrastruttura viene usato da Amazon ECS per gestire i volumi Amazon EBS associati alle attività, assicurati di quanto segue prima di interrompere le attività che usano i volumi Amazon EBS.

- Il ruolo non viene eliminato.
- La policy di attendibilità per il ruolo non viene modificata per rimuovere l'accesso ad Amazon ECS (`ecs.amazonaws.com`).
- La policy gestita da `AmazonECSInfrastructureRolePolicyForVolumes` non viene rimossa. Se devi modificare le autorizzazioni del ruolo, mantieni almeno `ec2:DetachVolume`, `ec2>DeleteVolume` e `ec2:DescribeVolumes` per l'eliminazione dei volumi.

L'eliminazione o la modifica del ruolo prima dell'interruzione delle attività con volumi Amazon EBS collegati comporterà il blocco delle attività in DEPROVISIONING e l'impossibilità di eliminare i volumi Amazon EBS associati. Amazon ECS riproverà automaticamente a intervalli regolari a interrompere l'attività e a eliminare il volume fino a quando non saranno ripristinate le autorizzazioni necessarie. Puoi visualizzare lo stato del volume allegato a un'attività e il motivo dello stato associato utilizzando l'API. [DescribeTasks](#)

Dopo aver creato il file, è necessario concedere all'utente l'autorizzazione a trasferire il ruolo ad Amazon ECS.

Autorizzazione a trasferire il ruolo dell'infrastruttura ad Amazon ECS

Per usare un ruolo IAM dell'infrastruttura ECS, devi concedere all'utente l'autorizzazione a trasferire il ruolo ad Amazon ECS. Allega la seguente autorizzazione all'utente `iam:PassRole`. Sostituiscilo *ecsInfrastructureRole* con il nome del ruolo di infrastruttura che hai creato.

JSON

```
{
 "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

### Profilo delle istanze gestite da Amazon ECS

Un profilo dell'istanza è un container IAM che contiene esattamente un ruolo IAM e permette alle istanze gestite da Amazon ECS di assumere tale ruolo in modo sicuro. Il profilo dell'istanza contiene un ruolo dell'istanza che l'agente ECS assume per registrare le istanze nei cluster e comunicare con il servizio ECS.

#### Important

Se utilizzi Amazon ECS Managed Instances con la policy AWS-managed Infrastructure, il profilo dell'istanza deve essere denominato `ecsInstanceRole`. Se usi una policy personalizzata per il ruolo Infrastruttura, il profilo dell'istanza può avere un nome alternativo.

Crea il ruolo con la policy di attendibilità

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `ecsInstanceRole-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": { "Service": "ec2.amazonaws.com" },
 "Action": "sts:AssumeRole"
 }
]
}
```

2. Utilizzate il AWS CLI comando seguente per creare un ruolo denominato `ecsInstanceRole` utilizzando la politica di fiducia creata nel passaggio precedente.

```
aws iam create-role \
 --role-name ecsInstanceRole \
 --assume-role-policy-document file://ecsInstanceRole-trust-policy.json
```

3. Allega la AmazonECSInstanceRolePolicyForManagedInstances politica AWS gestita al ecsInstanceRole ruolo.

```
aws iam attach-role-policy \
 --role-name ecsInstanceRole \
 --policy-arn arn:aws:iam::aws:policy/
AmazonECSInstanceRolePolicyForManagedInstances
```

#### Note

Se scegli di applicare le autorizzazioni con privilegi minimi e di specificare invece le tue autorizzazioni, puoi aggiungere le seguenti autorizzazioni per aiutarti a risolvere i problemi relativi alle attività con Amazon ECS Managed Instances:

- `ecs:StartTelemetrySession`
- `ecs:PutSystemLogEvents`

Puoi inoltre usare il flusso di lavoro della policy di attendibilità della console IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando criteri di attendibilità personalizzati \(console\)](#) nella Guida per l'utente di IAM.

Dopo aver creato il file, è necessario concedere all'utente l'autorizzazione a trasferire il ruolo ad Amazon ECS.

Crea il profilo dell'istanza utilizzando il AWS CLI

Dopo aver creato il ruolo, crea il profilo dell'istanza usando la AWS CLI:

```
aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

Aggiungi il ruolo al profilo dell'istanza:

```
aws iam add-role-to-instance-profile \
 --instance-profile-name ecsInstanceRole \
 --role-name ecsInstanceRole
```

```
--role-name ecsInstanceRole
```

Verificare che il profilo sia stato creato correttamente:

```
aws iam get-instance-profile --instance-profile-name ecsInstanceRole
```

## Ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico

Un ruolo IAM dell'infrastruttura Amazon ECS per i bilanciatori del carico consente ad Amazon ECS di gestire le risorse dei bilanciatori del carico nei cluster per te e viene usato quando:

- Vuoi utilizzare le blue/green distribuzioni con Amazon ECS. Il ruolo infrastruttura consente ad Amazon ECS di gestire le risorse del bilanciatore del carico per le tue implementazioni.
- È necessario Amazon ECS per creare, modificare o eliminare risorse del bilanciatore del carico, quali gruppi di destinazione e listener, durante le operazioni di implementazione.

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 le risorse del bilanciamento del carico per le tue distribuzioni blu/green, il log sarà o. CloudTrail `roleSessionName ECSNetworkingWithELB ecs-service-scheduler` Puoi usare questo nome per cercare eventi nella CloudTrail console filtrando per nome utente.

Amazon ECS fornisce una policy gestita che contiene le autorizzazioni necessarie per la gestione del bilanciatore del carico. Per ulteriori informazioni, consulta [Amazon ECSInfrastructure RolePolicyForLoadBalancers](#) nella AWS Managed Policy Reference Guide.

Creazione del ruolo dell'infrastruttura Amazon ECS per i bilanciatori del carico

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:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
```

```
{
 "Sid": "AllowAccessToECSForInfrastructureManagement",
 "Effect": "Allow",
 "Principal": {
 "Service": "ecs.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
}
```

- Utilizzate il AWS CLI comando seguente per creare un ruolo denominato `ecsInfrastructureRoleForLoadBalancers` utilizzando la politica di fiducia creata nel passaggio precedente.

```
aws iam create-role \
 --role-name ecsInfrastructureRoleForLoadBalancers \
 --assume-role-policy-document file://ecs-infrastructure-trust-policy.json
```

- Allega la `AmazonECSInfrastructureRolePolicyForLoadBalancers` politica AWS gestita al `ecsInfrastructureRoleForLoadBalancers` ruolo.

```
aws iam attach-role-policy \
 --role-name ecsInfrastructureRoleForLoadBalancers \
 --policy-arn arn:aws:iam::aws:policy/
AmazonECSInfrastructureRolePolicyForLoadBalancers
```

Puoi inoltre usare il flusso di lavoro della policy di attendibilità della console IAM per creare il ruolo. Per ulteriori informazioni, consulta [Creazione di un ruolo utilizzando criteri di attendibilità personalizzati \(console\)](#) nella Guida per l'utente di IAM.

#### Important

Se il ruolo di infrastruttura viene utilizzato da Amazon ECS per gestire le risorse di bilanciamento del carico per le tue blue/green distribuzioni, verifica quanto segue prima di eliminare o modificare il ruolo:

- Il ruolo non viene eliminato durante le implementazioni attive.
- La policy di attendibilità per il ruolo non viene modificata per rimuovere l'accesso ad Amazon ECS (`ecs.amazonaws.com`).



- La policy gestita da `AmazonECSInfrastructureRolePolicyForLoadBalancers` non viene rimossa mentre sono in corso implementazioni attive.

L'eliminazione o la modifica del ruolo durante le blue/green distribuzioni attive può causare errori di distribuzione e lasciare i servizi in uno stato incoerente.

Dopo aver creato il file, è necessario concedere all'utente l'autorizzazione a trasferire il ruolo ad Amazon ECS.

Autorizzazione a trasferire il ruolo dell'infrastruttura ad Amazon ECS

Per usare un ruolo IAM per i bilanciatori del carico dell'infrastruttura ECS, devi concedere all'utente l'autorizzazione a trasferire il ruolo ad Amazon ECS. Allega la seguente autorizzazione all'utente `iam:PassRole`. *`ecsInfrastructureRoleForLoadBalancers`* Sostituiscilo con il nome del ruolo di infrastruttura che hai creato.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": "iam:PassRole",
 "Effect": "Allow",
 "Resource": [
 "arn:aws:iam::*:role/ecsInfrastructureRoleForLoadBalancers",
 {
 "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 Guida di riferimento alle policy gestite da AWS :

- [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

### Console di gestione AWS

Per creare il ruolo di servizio per CodeDeploy (console IAM)

1. Accedi Console di gestione AWS 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 Tipo di entità attendibile, seleziona 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.

### JSON

```
{
 "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. Collega la policy gestita da `AWSCodeDeployRoleForECS` o da `AWSCodeDeployRoleForECSLimited` al ruolo `ecsTaskRole`.

```
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 delle 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 policy.

### Autorizzazioni del ruolo di esecuzione dell'attività

Quando le attività del tuo servizio richiedono un ruolo di esecuzione delle 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 policy. 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

### Console di gestione AWS

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi Console di gestione AWS 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>"]
 }
]
}
```

```
}
]
}
```

- 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.

- 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.
- Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al CodeDeploy ruolo. Per informazioni su come associare la policy al ruolo, consulta [Aggiorna autorizzazioni per un ruolo](#) nella Guida per l'utente di AWS Identity and Access Management

## AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

- Crea un file denominato `blue-green-iam-passrole.json`, con il seguente contenuto:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": ["arn:aws:iam::*:role/code-deploy-role"],
 "Condition": {
```

```

 "StringEquals": {"iam:PassedToService":
 "ecs.amazonaws.com"}
 }
 }
]
}

```

2. Usa il seguente comando per creare la policy IAM utilizzando il file del 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 usando il seguente comando.

```

aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cdTaskExecutionPolicy`].Arn'

```

4. Utilizza il seguente comando 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`.

JSON

```

{
 "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.

## JSON

```

{
 "Version": "2012-10-17",

```

```

 "Statement": [
 {
 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": [
 "arn:aws:iam::111122223333:role/ecsTaskExecutionRole_or_TaskRole_name"
]
 }
]
 }
}

```

Puoi scegliere di lasciare che Console di gestione AWS 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:

JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "",
 "Effect": "Allow",
 "Principal": {
 "Service": "events.amazonaws.com"
 },
 "Action": "sts:AssumeRole"
 }
]
}

```

2. Usa il seguente comando per creare un ruolo IAM denominato `ecsEventsRole` usando la policy di attendibilità creata nel passaggio precedente.



```
aws iam create-role \
 --role-name ecsEventsRole \
 --assume-role-policy-document file://eventbridge-trust-policy.json
```

3. Associa il AWS manager AmazonEC2ContainerServiceEventsRole al ecsEventsRole ruolo usando il seguente comando.

```
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 criteri di attendibilità personalizzati \(console\)](#) nella Guida per l'utente di 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.

### Console di gestione AWS

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi Console di gestione AWS 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::111122223333: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 associare la policy al ruolo, consulta [Aggiorna autorizzazioni per un ruolo](#) nella Guida per l'utente di AWS Identity and Access Management

## AWS CLI

Sostituisci tutto *user input* con le tue informazioni.

1. Crea un file denominato `ev-iam-passrole.json`, con il seguente contenuto:

### JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {

```

```

 "Effect": "Allow",
 "Action": "iam:PassRole",
 "Resource": [
 "arn:aws:iam::111122223333: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 usando 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 la registrazione di 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.
  - lo sono: `AttachRolePolicy`
  - lo sono: `CreateRole`
  - lo sono: `CreateInstanceProfile`
  - sono: `AddRoleToInstanceProfile`
  - lo sono: `ListInstanceProfilesForRole`
  - lo sono: `GetRole`
- `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`.

## JSON

```
{
 "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.
  - lo sono: `AttachRolePolicy`
  - lo sono: `CreateRole`
  - lo sono: `GetRole`

 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`.

## JSON

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:GetRole"
],
 "Resource": "arn:aws:iam::*:role/ecsTaskExecutionRole"
 }
]
}
```

Autorizzazioni necessarie per usare Amazon Q Developer per fornire consigli nella console

Affinché Amazon Q Developer fornisca raccomandazioni nella console Amazon ECS, è necessario abilitare le autorizzazioni IAM corrette per il proprio utente o ruolo IAM. Devi aggiungere l'autorizzazione `codewhisperer:GenerateRecommendations`.

## JSON

```
{
 "Statement": [
 {
 "Sid": "AmazonQDeveloperPermissions",
```

```

 "Effect": "Allow",
 "Action": ["codewhisperer:GenerateRecommendations"],
 "Resource": "*"
 }
]
}

```

Per utilizzare la chat in linea nella console Amazon ECS, è necessario abilitare le autorizzazioni IAM corrette per il ruolo o l'utente IAM. Devi aggiungere l'autorizzazione `q:SendMessage`.

## JSON

```

{
 "Statement": [
 {
 "Sid": "AmazonQDeveloperInlineChatPermissions",
 "Effect": "Allow",
 "Action": ["q:SendMessage"],
 "Resource": "*"
 }
]
}

```

## 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`.

JSON

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "iam:AttachRolePolicy",
 "iam:CreateRole",
 "iam:GetRole",
 "iam:PassRole"
],
 "Resource": "arn:aws:iam::*:role/ecsEventsRole"
 }
]
}
```

## Autorizzazioni richieste per visualizzare le implementazioni del servizio

Quando si segue la best practice della concessione dei privilegi minimi, è necessario aggiungere ulteriori autorizzazioni per visualizzare le implementazioni del servizio nella console.

È necessario accedere alle seguenti azioni:

- `ListServiceDeployments`
- `DescribeServiceDeployments`
- `DescribeServiceRevisions`

È necessario accedere alle seguenti risorse:

- Servizio
- Implementazioni del servizio
- Revisione del servizio



La seguente policy di esempio contiene le autorizzazioni necessarie e limita le operazioni a un servizio specifico.

Sostituire `account`, `cluster-name` e `service-name` con i valori.

JSON

```
{
 "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 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. Per ulteriori informazioni, consulta [Aggiunta e rimozione delle policy IAM](#).

- eventi: DescribeRule
- eventi: ListTargetsByRule
- registri: DescribeLogGroups

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "events:DescribeRule",
 "events:ListTargetsByRule",
 "logs:DescribeLogGroups"
],
 "Resource": "*"
 }
]
}
```

## Autorizzazioni necessarie per abilitare gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container

Per configurare gli eventi del ciclo di vita sono necessarie le seguenti autorizzazioni:

- eventi: PutRule
- eventi: PutTargets
- registri: CreateLogGroup

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "events:PutRule",
 "events:PutTargets",
 "logs:CreateLogGroup"
],
 }
]
}
```

```
 "Resource": "*"
 }
]
}
```

## Autorizzazioni richieste per la console Amazon ECS con CloudFormation

Prima di Console di gestione AWS utilizzarlo per creare le tue risorse, devi assicurarti di disporre delle autorizzazioni IAM corrette. Per informazioni su come impostare 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 [Creating IAM policies](#) nella Guida per l'utente di IAM.

### Autorizzazioni necessarie per la creazione di un cluster

Quando crei un cluster nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. CloudFormation

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack CloudFormation . Ciò è necessario quando si creano cluster Amazon ECS tramite la Console di gestione AWS e la successiva gestione di tali cluster.
- `ssm`— Consente di CloudFormation fare riferimento alle più recenti AML ottimizzate per Amazon ECS. Ciò è necessario quando si creano cluster Amazon ECS utilizzando. Console di gestione AWS

La seguente policy contiene le CloudFormation autorizzazioni richieste e limita le azioni alle risorse create nella console Amazon ECS.

## JSON

```
{
 "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 utilizzi gruppi con dimensionamento automatico, sono necessarie autorizzazioni aggiuntive affinché la console possa aggiungere tag ai gruppi con dimensionamento automatico quando si utilizza la funzionalità di dimensionamento automatico 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.

JSON

```
{
 "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 la creazione di un servizio

Quando crei un servizio nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. CloudFormation Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack CloudFormation . Ciò è necessario quando si creano servizi Amazon ECS utilizzando la Console di gestione AWS 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.

JSON

```
{
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "cloudformation:CreateStack",
 "cloudformation>DeleteStack",
 "cloudformation:DescribeStack*",
 "cloudformation:UpdateStack"
],
 "Resource": [
 "arn:*:cloudformation:*:*:stack/ECS-Console-V2-Service-*"
]
 }
]
}
```

## Autorizzazioni richieste per le funzioni Lambda nelle distribuzioni Amazon ECS blue/green

Quando usi le funzioni Lambda come hook del ciclo di vita dell'implementazione nelle implementazioni blu/verdi di Amazon ECS, devi creare un ruolo IAM con autorizzazioni specifiche. Questo ruolo consente ad Amazon ECS di invocare le funzioni Lambda in varie fasi del ciclo di vita dell'implementazione.

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `lambda:InvokeFunction`: consente ad Amazon ECS di invocare le funzioni Lambda configurate come hook del ciclo di vita durante il processo di implementazione.

Per la policy di attendibilità, è necessario consentire al servizio di assumere questo ruolo:

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Principal": {
 "Service": [
 "ecs.amazonaws.com"
]
 },
 "Action": "sts:AssumeRole"
 }
]
}
```

## Autorizzazioni IAM richieste per il dimensionamento automatico dei servizi 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 dimensionamento automatico del servizio.

- 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*`

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 i 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 policy che consente agli utenti di creare cluster e aggiungere tag solo 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`.

## Controllo dell'accesso a tag specifici di Amazon ECS

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 con più valori, consulta [Condizioni con più chiavi o valori di contesto](#) nella Guida per l'utente IAM.

## 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 policy 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 policy 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 o un SDK. AWS CLI AWS

Esempio: consentire 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 una key/value coppia di «Scopose/Testing».

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Action": [
```

```
 "ecs:DeleteCluster"
],
 "Effect": "Allow",
 "Resource": "arn:aws:ecs:us-east-1:111122223333: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](#)
- [Sto riscontrando problemi con il mio profilo di istanza Amazon ECS Managed Instances](#)
- [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 amministratore. AWS 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 trasmettere 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 consentire alle persone di accedere alle tue risorse.

Per maggiori 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 tue risorse attraverso Account AWS le risorse di tua proprietà, consulta [Fornire l'accesso a un utente IAM in 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 di IAM.

## Sto riscontrando problemi con il mio profilo di istanza Amazon ECS Managed Instances

Se utilizzi la policy `AmazonECSInfrastructureRolePolicyForManagedInstances` gestita fornita da Amazon ECS, il profilo dell'istanza deve avere `ecsInstanceRole` come prefisso. Il motivo è che la policy gestita è autorizzata solo a trasferire ruoli con questo prefisso.

## Altre risorse per la risoluzione dei problemi

Nelle pagine seguenti vengono fornite informazioni sui codici di errore:

- [Amazon ECS ha interrotto i messaggi di errore relativi alle attività](#)
- [Visualizzare i 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 [Panoramica delle autorizzazioni e delle policy per la gestione degli accessi in Amazon ECS](#).

## 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. Per ulteriori informazioni, consulta [Policy resources for Amazon ECS](#).

## Lascia 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.

JSON

```
{
 "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 policy seguente, un principale IAM avvia e interrompe tutte le attività con una chiave di tag di Department e un valore di tag di Accounting.

JSON

```

{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ecs:StartTask",
 "ecs:StopTask",
 "ecs:RunTask"
],
 "Resource": "arn:aws:ecs:us-east-1:123456789012:cluster/my-cluster",
 "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 [Linee guida sugli audit di sicurezza AWS](#) nella Guida per l'utente di AWS Identity and Access Management .

## 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 Fargate, è possibile 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 [Scalabilità automatica del servizio Amazon ECS](#).

Per i cluster che utilizzano attività o servizi EC2, 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.. EC2

## 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 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 [Registrazione delle chiamate API Amazon ECS tramite 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. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la [Documentazione AWS sulla sicurezza](#).

## Best practice di conformità e sicurezza per Amazon ECS

La tua responsabilità di conformità durante l'utilizzo di Amazon ECS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e normative vigenti.

### 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 best practice per la sicurezza dell'infrastruttura, consulta [Protezione dell'infrastruttura](#) nel pilastro Sicurezza del framework AWS Well-Architected.

## AWS Security Hub CSPM

Usa AWS Security Hub CSPM. Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno AWS. Security Hub CSPM utilizza i controlli di sicurezza per valutare le AWS risorse e verificare la conformità rispetto agli standard e alle migliori pratiche del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta il riferimento ai controlli [CSPM di Security Hub](#).

## 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 garantire 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)

Il Federal Information Processing Standard (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 140.3 del FIPS-140
- La conformità a FIPS-140 è disattivata per impostazione predefinita. Devi attivarla.
- Amazon CloudWatch non supporta un endpoint FIPS dualstack che può essere utilizzato per monitorare le attività di Amazon ECS in IPv6 una sola configurazione che utilizza la conformità FIPS-140.
- 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 è valido per 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 "Registrazione delle chiamate API Amazon ECS tramite AWS CloudTrail"](#).

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'azione dell'PutAccountSettingDefaultAPI:

```
{
 "eventVersion": "1.08",
 "userIdentity": {
 "type": "IAMUser",
 "principalId": "AIDAIIV5AJI5LXF5EXAMPLE",
 "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.

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

Considerazioni relative agli endpoint nelle regioni 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:

- Devi disporre dei seguenti endpoint VPC specifici della regione:

#### Note

Se non configuri tutti gli endpoint, il traffico passerà attraverso gli endpoint pubblici, non attraverso l'endpoint VPC.

- `com.amazonaws.region.ecs-agent`
- `com.amazonaws.region.ecs-telemetry`
- `com.amazonaws.region.ecs`

Ad esempio, la regione del 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`
- Quando 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 in cui è stata copiata, esegui una delle seguenti operazioni.

Ad esempio, la regione da cui viene copiato è quella degli Stati Uniti orientali (Virginia settentrionale) (us-east-1). La regione di destinazione della copia è il Canada occidentale (Calgary) (ca-west-1).

| Configurazione                                                                               | Azione                                                                                                                                                                                                                                                                                                                               |  |
|----------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| La regione da cui è stata effettuata la copia non dispone di 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 da cui è stata effettuata la copia contiene endpoint VPC specifici della regione. | <ol style="list-style-type: none"> <li>Crea tutti e tre gli endpoint VPC per la nuova regione (ad esempio, <code>com.amazonaws.ca-west-1.ecs-agent</code>).</li> <li>Elimina tutti e tre gli endpoint VPC per la regione da cui è stata effettuata la copia (ad esempio, <code>com.amazonaws.us-east-1.ecs-agent</code>).</li> </ol> |  |

### Considerazioni sugli endpoint VPC di Amazon ECS per Fargate

Quando è presente un endpoint VPC per `ecr.dkr` e `ecr.api` nello stesso VPC in cui viene implementata un'attività Fargate, verrà utilizzato l'endpoint VPC. Se non è presente alcun endpoint VPC, verrà utilizzata 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 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 [Autorizzazioni per attività Fargate che eseguono il pull delle immagini Amazon ECR su endpoint di interfaccia](#).
  - Se le tue attività sono in una configurazione IPv6 solo e utilizzano un URI di immagine dualstack Amazon ECR, tieni presente che Amazon ECR non supporta gli endpoint VPC con interfaccia dualstack. Per ulteriori informazioni, consulta la sezione [Guida introduttiva all'invio di richieste IPv6](#) nella Amazon Elastic Container Registry User Guide.
- 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 Gestione dei segreti AWS .
  - 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 EC2

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- Le attività utilizzate EC2 richiedono che le istanze del contenitore su cui vengono lanciate 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 Gestione dei segreti AWS.
- 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.

## Comprensione dei modelli di denominazione degli endpoint Amazon ECS

È importante comprendere che l'agente Amazon ECS può inviare richieste a endpoint con suffissi numerati, ad esempio:

- `ecs-a-1.region.amazonaws.com`, `ecs-a-2.region.amazonaws.com`, ecc. per gli endpoint degli agenti
- `ecs-t-1.region.amazonaws.com`, `ecs-t-2.region.amazonaws.com`, ecc. per gli endpoint di telemetria

Questo comportamento si verifica perché l'agente Amazon ECS utilizza l'[DiscoverPollEndpoint](#) API per determinare dinamicamente a quale endpoint specifico connettersi. Se gli endpoint VPC non gestiscono correttamente queste varianti numerate degli endpoint, l'agente tornerà a usare gli endpoint pubblici, anche se hai impostato gli endpoint VPC per i nomi di base.

### Il ruolo dell'API DiscoverPollEndpoint

L'[DiscoverPollEndpoint](#) API viene utilizzata dall'agente Amazon ECS per individuare l'endpoint appropriato su cui eseguire il polling delle attività. Quando l'agente richiama questa API, riceve un URL dell'endpoint specifico che può includere un suffisso numerato. Per garantire il corretto funzionamento degli endpoint VPC, la configurazione di rete deve permettere all'agente di:

1. Accedi all'API DiscoverPollEndpoint
2. Connect all'endpoint restituito URLs, compresi quelli con suffissi numerati

Se stai risolvendo i problemi di connettività degli endpoint VPC, verifica che il tuo agente possa raggiungere sia gli endpoint di base che qualsiasi variante numerata che potrebbe essere restituita dall'API. DiscoverPollEndpoint

## 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 attraverso gli endpoint pubblici, non attraverso l'endpoint VPC.

Quando crei un endpoint, Amazon ECS crea anche un nome DNS privato per l'endpoint. Ad esempio, `ecs-a.region.amazonaws.com` per `ecs-agent` e `ecs-t.region.amazonaws.com` per `ecs-telemetry`.

- `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'endpoint `ecs-agent` usa l'API `ecs:poll`, mentre l'endpoint `ecs-telemetry` utilizza le API `ecs:poll` e `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 della gestione del sistema operativo guest (con relativi aggiornamenti e patch di sicurezza), di altri software applicativi associati e della configurazione del firewall del gruppo di sicurezza fornito da AWS. 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.

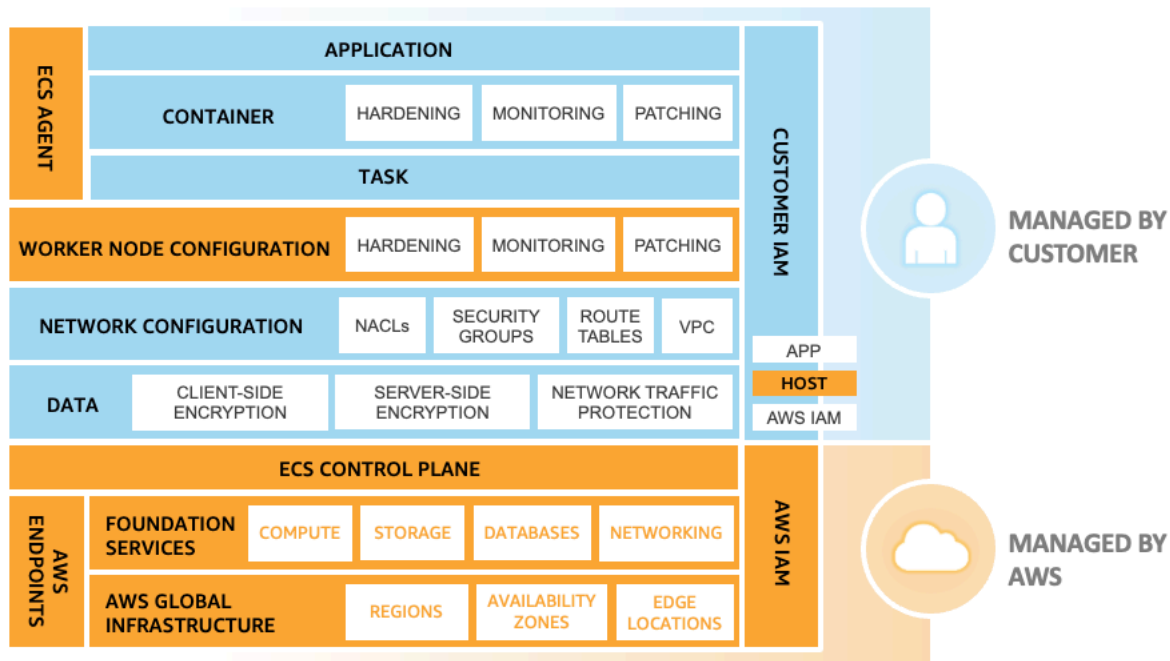
## Fargate

La seguente figura 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 dell'infrastruttura dedicata. I carichi di lavoro containerizzati in esecuzione su Fargate non condividono il sistema operativo, il kernel Linux, l'interfaccia di rete, l'archiviazione temporanea, la CPU o la memoria con altre attività. Quando usano Fargate, i clienti non sono responsabili della sicurezza dell'infrastruttura di elaborazione che esegue i loro container. Fargate provvede 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](#).

Sei responsabile della gestione delle seguenti risorse:

- Configurazione di rete tra cui VPC NACLs, gruppi di sicurezza e tabelle di routing
- Crittografia dei dati dei clienti e dei servizi. Per ulteriori informazioni, consulta [Opzioni di archiviazione per le attività di Amazon ECS](#).
- Immagini di container. Per ulteriori informazioni, consulta [Best practice per la sicurezza di attività e container di Amazon ECS](#).
- Autorizzazioni IAM per le applicazioni usando il task role. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

## AWS Shared Responsibility Model for Amazon ECS with Fargate

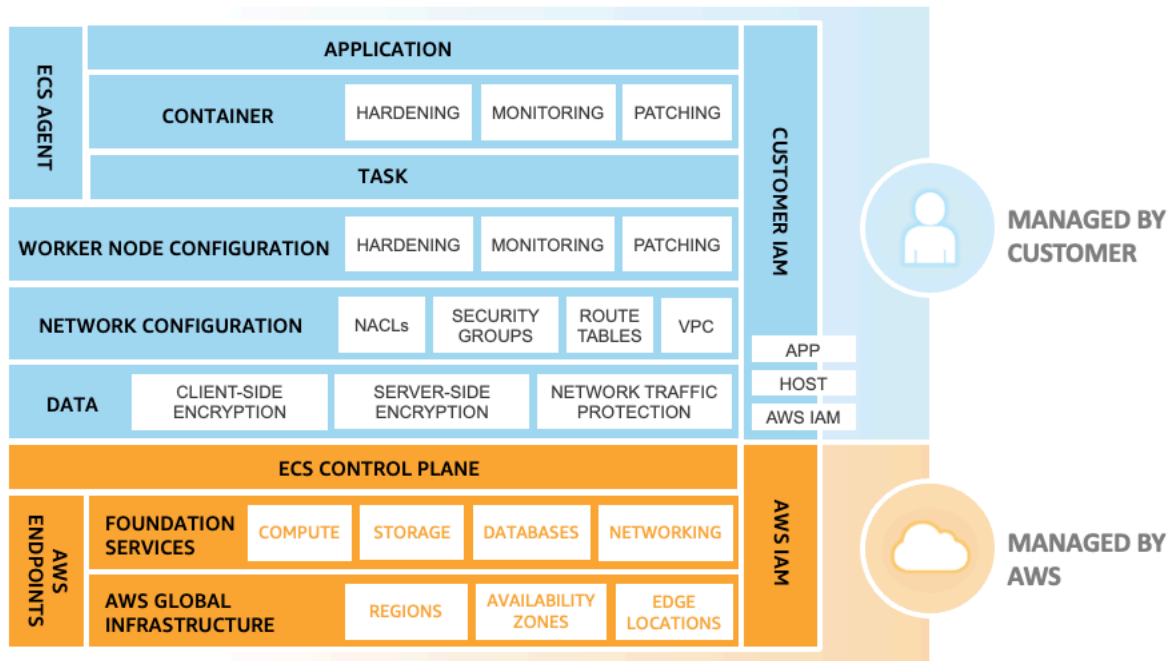


## EC2

La figura seguente mostra la responsabilità condivisa di EC2. 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 dei dati dei clienti e dei servizi. Per ulteriori informazioni, consulta [Opzioni di archiviazione per le attività di Amazon ECS](#).
- Immagini di container. Per ulteriori informazioni, consulta [Best practice per la sicurezza di attività e container di Amazon ECS](#).
- Autorizzazioni IAM per le applicazioni usando il task role. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).

## AWS Shared Responsibility Model for Amazon ECS



## Modello di responsabilità condivisa per le istanze gestite da Amazon ECS.

Amazon ECS Managed Instances offre una soluzione gestita per carichi di lavoro containerizzati che combina la semplicità operativa di Fargate con l'accesso all'intera gamma di tipi e funzionalità di istanze Amazon. EC2 AWS gestisce il provisioning, l'applicazione di patch, la scalabilità e la manutenzione dell'infrastruttura, mentre i clienti mantengono il controllo sulle proprie applicazioni e configurazioni specifiche.

A differenza di Fargate, i carichi di lavoro containerizzati in esecuzione sulle istanze gestite da Amazon ECS condividono il sistema operativo, il kernel Linux, l'interfaccia di rete, l'archiviazione temporanea, la CPU, la memoria e le risorse GPU con altre attività sulla stessa istanza. Amazon ECS ottimizza l'uso dell'infrastruttura assegnando più attività a istanze più grandi per ridurre al minimo la capacità inutilizzata.

## AWS responsabilità

Quando utilizza Amazon ECS Managed Instances, AWS è responsabile di:

- Del provisioning delle istanze e della gestione del ciclo di vita

- Dell'applicazione di patch del sistema operativo e degli aggiornamenti di sicurezza
- Del dimensionamento e dell'ottimizzazione dell'infrastruttura
- Della sostituzione e manutenzione delle istanze (durata massima dell'istanza di 21 giorni)
- Delle restrizioni al controllo degli accessi (nessun accesso SSH, nessun accesso a SSM Session Manager)
- Crittografia Amazon EC2 Instance Storage, ovvero lo storage collegato direttamente all'istanza. Per ulteriori informazioni, consulta la sezione [Protezione dei dati in Amazon EC2](#).
- Amazon ECS gestisce i volumi collegati alle EC2 istanze Amazon al momento della creazione, inclusi i volumi root e di dati.
- Amazon ECS utilizza istanze EC2 under-the-hood gestite da Amazon. Per ulteriori informazioni sulle istanze EC2 gestite da Amazon, consulta [Security in Amazon EC2](#).

## Responsabilità del cliente

Sei responsabile della gestione delle seguenti risorse:

- Configurazione di rete tra cui VPC NACLs, gruppi di sicurezza e tabelle di routing
- Crittografia dei dati dei clienti e dei servizi. Per ulteriori informazioni, consulta [Opzioni di archiviazione per le attività di Amazon ECS](#).
- Immagini di container. Per ulteriori informazioni, consulta [Best practice per la sicurezza di attività e container di Amazon ECS](#).
- Autorizzazioni IAM per le applicazioni usando il task role. Per ulteriori informazioni, consulta [Ruolo IAM dell'attività Amazon ECS](#).
- Configurazione e monitoraggio a livello di applicazione
- Definizioni di attività e servizi
- Considerazioni relative alla sicurezza per i carichi di lavoro che condividono le risorse dell'istanza sottostante
- Configurazioni di container privilegiate e funzionalità Linux avanzate (CAP\_NET\_ADMIN, CAP\_BPF, ecc.) quando abilitate
- Operazioni di gestione tramite l'API Amazon ECS (non è disponibile l'accesso diretto all'istanza tramite SSH o SSM)

# Best practice sulla sicurezza di rete per Amazon ECS

La sicurezza di rete è un argomento ampio che comprende diversi argomenti secondari. Questi includono segmentazione e isolamento della rete encryption-in-transit, firewall, routing del traffico e osservabilità.

## Crittografia dei dati 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 bilanciatore 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 listener. A tale scopo, ciascuna dispone di un suo certificato TLS. Consigliamo di effettuare il provisioning dei certificati per il bilanciatore del carico utilizzando AWS Certificate Manager (ACM) e di aggiungerli all'elenco di certificati del listener. 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 bilanciatore del carico seleziona tale certificato. Se un hostname fornito da un client corrisponde a più certificati nell'elenco di certificati, il bilanciatore 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 container che emetta una richiesta di firma del certificato (CSR) ad ACM e quindi monti il segreto risultante in 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 Autorità di certificazione privata](#)

## 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 usi 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.api.aws`).

## 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 di 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.

## Raccomandazioni sulla sicurezza di rete

Consigliamo di completare la procedura seguente quando configuri Amazon VPC, i sistemi di bilanciamento del carico e la rete.

### Uso della crittografia di rete laddove 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, utilizzalo 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 rete **awsvpc** e i gruppi di sicurezza per controllare il traffico tra le 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 basato su 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](#).

**Configurazione di AWS PrivateLink endpoint ove giustificato per Amazon ECS**

È necessario configurare gli endpoint quando necessario. AWS PrivateLink 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. Gestione dei segreti AWS CloudWatch

**Uso di Amazon VPC Flow Logs per analizzare il traffico da e verso 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 aws-vpc 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 dalla Galleria pubblica di Amazon ECR, 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 documentazione su distroless. GitHub](#)

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 multifase, consulta [Creazione di build multifase](#) nella documentazione Docker.

## 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 un rilevamento delle vulnerabilità più accurato su un'ampia gamma di sistemi operativi diffusi. Ciò permette ai clienti di rafforzare ulteriormente la sicurezza delle immagini di container. Tutti i nuovi registri dei clienti sono inseriti in questa versione migliorata per impostazione predefinita.
- **Scansione di base di Clair:** la versione precedente della scansione di base che usa 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. Dato che il supporto per Clair è obsoleto, Clair non sarà supportato nelle nuove regioni man mano che vengono aggiunte e non sarà più supportato in tutte le regioni a partire dal 1° ottobre 2025.

Amazon ECR utilizza la severity di un CVE dalla fonte di distribuzione upstream, se disponibile. Altrimenti, viene usato il punteggio CVSS (Common Vulnerability Scoring System). 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 CSPM](#) spiega come far emergere le informazioni sulle vulnerabilità da Amazon ECR in AWS Security Hub CSPM 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 cosa c'è nell'immagine e le immagini possono presentare delle vulnerabilità.

## 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 utilizzare Amazon Inspector. Per ulteriori informazioni, consulta la sezione [Scansione delle immagini dei container Amazon ECR con Amazon Inspector](#) nella Guida per l'utente di Amazon Inspector.

## 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.

Come parte della tua CI/CD pipeline, dovresti lint Dockerfiles per cercare la `USER` direttiva e fallire la compilazione se manca. 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, puoi AWS Lambda utilizzarlo 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 `pscap` 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.

È possibile seguire i seguenti tutorial per saperne di più su come iniziare a utilizzare Amazon ECS.

| Panoramica del tutorial                               | Ulteriori informazioni                                                                              |  |
|-------------------------------------------------------|-----------------------------------------------------------------------------------------------------|--|
| Iniziare a utilizzare Amazon ECS su Fargate.          | <a href="#">Ulteriori informazioni su come creare un'attività di Amazon ECS Linux per Fargate</a>   |  |
| Iniziare a utilizzare i container Windows su Fargate. | <a href="#">Ulteriori informazioni su come creare un'attività di Amazon ECS Windows per Fargate</a> |  |
| Inizia a usare i contenitori Windows per EC2.         | <a href="#">Scopri come creare un'attività Amazon ECS Windows per EC2</a>                           |  |

Puoi utilizzare uno qualsiasi dei seguenti tutorial per distribuire attività su Amazon ECS utilizzando AWS CLI

| Panoramica del tutorial                       | Ulteriori informazioni                                                              |  |
|-----------------------------------------------|-------------------------------------------------------------------------------------|--|
| Creazione di un'attività Linux per Fargate.   | <a href="#">Creazione di un'attività Amazon ECS Linux per Fargate con AWS CLI</a>   |  |
| Creazione di un'attività Windows per Fargate. | <a href="#">Creazione di un'attività Amazon ECS Windows per Fargate con AWS CLI</a> |  |
| Crea un'attività Linux per EC2                | <a href="#">Creazione di un'attività Amazon ECS per EC2 con AWS CLI</a>             |  |

È possibile seguire questi tutorial per saperne di più sul monitoraggio e sulla registrazione di log.

| 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 gli eventi di attività Amazon ECS arrestate</a> |  |
| Concatenare i messaggi di log originariamente appartenenti a un contesto ma suddivisi su più record o righe di log.                                                                                          | <a href="#">Concatenazione di messaggi di log Amazon ECS multilinea o di traccia dello stack</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="#">Implementazione di Fluent Bit su container Windows di Amazon ECS</a>                                      |  |

È possibile utilizzare i seguenti tutorial per saperne di più su come utilizzare l'autenticazione Active Directory con un account di servizio gestito del gruppo su Amazon ECS.

| Panoramica del tutorial                                                                                                                                        | Ulteriori informazioni                                                                            |  |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------|--|
| Usa l'account di servizio gestito di gruppo con contenuti Linux attivi. EC2                                                                                    | <a href="#">Utilizzo gMSA per EC2 Linux contenitori su Amazon ECS</a>                             |  |
| Usa un account di servizio gestito di gruppo con contenuti Windows attivi EC2.                                                                                 | <a href="#">Scopri come usare i contenuti g MSAs for EC2 Windows per Amazon ECS</a>               |  |
| Utilizzo dell'account di servizio gestito del gruppo con container Linux su Fargate.                                                                           | <a href="#">Utilizzo del gMSA per i container Linux su Fargate</a>                                |  |
| Creare un'attività che esegue un container Windows con credenziali di accesso ad Active Directory con un account di servizio gestito del gruppo senza dominio. | <a href="#">Utilizzo di contenitori Amazon ECS Windows senza dominio gMSA utilizzando AWS CLI</a> |  |

## Creazione di un'attività Amazon ECS Linux per 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 dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the 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](#).
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS\\_FullAccess](#).
- 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, consultare [Monitor Amazon ECS containers with Amazon ECS Exec](#).

- (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",
 "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 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, consulta [Installing or updating to the latest version of the 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](#).
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS\\_FullAccess](#).
- 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 EC2 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 dell' AWS CLI. Per ulteriori informazioni su come eseguire l'aggiornamento alla versione più recente, consulta [Installing or updating to the latest version of the 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](#)
- [Creazione di un cluster](#)
- [Avvio di un'istanza di container con l'AMI Amazon ECS](#)

- [Elencare le istanze di container](#)
- [Descrivere l'istanza di container](#)
- [Registrare una definizione di attività](#)
- [Elencare le definizioni di attività](#)
- [Creazione di un servizio](#)
- [Elencare i servizi](#)
- [Descrivere il servizio](#)
- [Descrivere l'attività in esecuzione](#)
- [Testare il server web](#)
- [Eseguire la pulizia delle risorse](#)

## 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](#).
- L'utente IAM dispone delle autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS\\_FullAccess](#).
- È stato creato un ruolo IAM dell'istanza di container per l'uso. Per ulteriori informazioni, consulta [Ruolo IAM delle istanze di container Amazon ECS](#).
- È stata creata una VPC 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 una propria istanza. EC2 Per ulteriori informazioni, consulta [Cos'è AWS CloudShell?](#) nella Guida AWS CloudShell per l'utente.

## Creazione di 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"
 }
}
```

## Avvio di un'istanza di container con l'AMI Amazon ECS

Le istanze di container sono EC2 istanze che eseguono l'agente container Amazon ECS e sono state registrate in un cluster. In questa sezione, avvierai un' EC2 istanza utilizzando l'AMI ottimizzata per ECS.

Per avviare un'istanza di contenitore con AWS CLI

1. Recupera l'ID AMI Amazon Linux 2 più recente ottimizzato per ECS utilizzando Regione AWS il seguente comando. Questo comando utilizza AWS Systems Manager Parameter Store per ottenere l'ID AMI ottimizzato per ECS più recente. L'AMI include l'agente del container Amazon ECS e il runtime di Docker preinstallati.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
recommended --query 'Parameters[0].Value' --output text | jq -r '.image_id'
```

Output:

```
ami-abcd1234
```

2. Creare un gruppo di sicurezza che consenta l'accesso SSH per la gestione dell'istanza di container e l'accesso HTTP per il server web.

```
aws ec2 create-security-group --group-name ecs-tutorial-sg --description "ECS
tutorial security group"
```

Output:

```
{
 "GroupId": "sg-abcd1234"
}
```

3. Aggiungere una regola in entrata al gruppo di sicurezza eseguendo il seguente comando.

```
aws ec2 authorize-security-group-ingress --group-id sg-abcd1234 --protocol tcp --
port 80 --cidr 0.0.0.0/0
```

Output:

```
{
 "Return": true,
 "SecurityGroupRules": [
 {
 "SecurityGroupRuleId": "sgr-efgh5678",
 "GroupId": "sg-abcd1234",
 "GroupOwnerId": "123456789012",
 "IsEgress": false,
 "IpProtocol": "tcp",
 "FromPort": 80,
 "ToPort": 80,
 "CidrIpv4": "0.0.0.0/0"
 }
]
}
```

Ora il gruppo di sicurezza consente l'accesso SSH dall'intervallo IP specificato e l'accesso HTTP da qualsiasi posizione. In un ambiente di produzione, è necessario limitare l'accesso SSH al

proprio indirizzo IP specifico e valutare la possibilità di limitare l'accesso HTTP in base alle esigenze.

4. Crea una EC2 key pair per l'accesso SSH all'istanza del tuo contenitore.

```
aws ec2 create-key-pair --key-name ecs-tutorial-key --query 'KeyMaterial' --output
text > ecs-tutorial-key.pem
chmod 400 ecs-tutorial-key.pem
```

La chiave privata viene salvata nel computer locale con autorizzazioni adeguate per l'accesso SSH.

5. Avvia un' EC2 istanza utilizzando l'AMI ottimizzata per ECS e configurala per unirsi al cluster.

```
aws ec2 run-instances --image-id ami-abcd1234 --instance-type t3.micro --key-
name ecs-tutorial-key --security-group-ids sg-abcd1234 --iam-instance-profile
Name=ecsInstanceRole --user-data '#!/bin/bash
echo ECS_CLUSTER=MyCluster >> /etc/ecs/ecs.config'
{
 "Instances": [
 {
 "InstanceId": "i-abcd1234",
 "ImageId": "ami-abcd1234",
 "State": {
 "Code": 0,
 "Name": "pending"
 },
 "PrivateDnsName": "",
 "PublicDnsName": "",
 "StateReason": {
 "Code": "pending",
 "Message": "pending"
 },
 "InstanceType": "t3.micro",
 "KeyName": "ecs-tutorial-key",
 "LaunchTime": "2025-01-13T10:30:00.000Z"
 }
]
}
```

Lo script di dati utente configura l'agente Amazon ECS per registrare l'istanza in MyCluster. L'istanza utilizza il ruolo IAM `ecsInstanceRole`, che fornisce le autorizzazioni necessarie per l'agente.

## Elencare le istanze di container

Entro pochi minuti dal lancio dell'istanza del contenitore, l'agente Amazon ECS registra l'istanza nel cluster. MyCluster Per visualizzare l'elenco delle istanze di container in un cluster, esegui questo comando:

```
aws ecs list-container-instances --cluster MyCluster
```

Output:

```
{
 "containerInstanceArns": [
 "arn:aws:ecs:us-east-1:aws_account_id:container-instance/container_instance_ID"
]
}
```

## Descrivere l'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 MyCluster --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.

## Registrare una definizione di attività

Prima di eseguire un'attività Windows nel cluster Amazon ECS, è necessario 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 `nginx`. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta [Definizioni dei processi di Amazon ECS](#).

```

{
 "family": "nginx-task",
 "containerDefinitions": [
 {
 "name": "nginx",
 "image": "public.ecr.aws/ecs-sample-image/amazon-ecs-sample:latest",
 "cpu": 256,
 "memory": 512,
 "essential": true,
 "portMappings": [
 {
 "containerPort": 80,
 "hostPort": 80,

```

```

 "protocol": "tcp"
 }
]
 },
 "requiresCompatibilities": ["EC2"],
 "networkMode": "bridge"
}

```

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 è possibile inserire il carattere di escape prima delle virgolette nel JSON e trasmettere le definizioni del container JSON nella riga di comando. 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/nginx.json
```

Il comando register-task-definition restituisce una descrizione della definizione di attività una volta completata la registrazione.

```

{
 "taskDefinition": {
 "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/nginx-task:1",
 "family": "nginx-task",
 "revision": 1,
 "status": "ACTIVE",
 "containerDefinitions": [
 {
 "name": "nginx",
 "image": "public.ecr.aws/docker/library/nginx:latest",
 "cpu": 256,
 "memory": 512,
 "essential": true,
 "portMappings": [
 {
 "containerPort": 80,

```



```

 "hostPort": 80,
 "protocol": "tcp"
 }
],
 "environment": [],
 "mountPoints": [],
 "volumesFrom": []
 }
],
"volumes": [],
"networkMode": "bridge",
"compatibilities": [
 "EC2"
],
"requiresCompatibilities": [
 "EC2"
]
}
}

```

## Elencare 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 possono essere utilizzati insieme nelle chiamate `create-service`.

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

Output:

```

{
 "taskDefinitionArns": [
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep360:1",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep360:2",
 "arn:aws:ec2:us-east-1:aws_account_id:task-definition/nginx-task: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"
]
}

```

## Creazione di un servizio

Dopo aver registrato un'attività per l'account e aver lanciato un'istanza di container registrata nel cluster, è possibile creare un servizio Amazon ECS che esegua e gestisca il numero desiderato di attività in contemporanea utilizzando la definizione di attività registrata. Per questo esempio, si inserisce una singola istanza della definizione dell'`nginx:1`attività nel cluster `MyCluster`.

```
aws ecs create-service --cluster MyCluster --service-name nginx-service --task-definition nginx-task:1 --desired-count 1
```

Output:

```
{
 "service": {
 "serviceArn": "arn:aws:ecs:us-east-1:aws_account_id:service/MyCluster/nginx-service",
 "serviceName": "nginx-service",
 "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/MyCluster",
 "taskDefinition": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/nginx-task:1",
 "desiredCount": 1,
 "runningCount": 0,
 "pendingCount": 0,
 "launchType": "EC2",
 "status": "ACTIVE",
 "createdAt": "2025-01-13T10:45:00.000Z"
 }
}
```

## Elencare i servizi

Ottieni un elenco dei servizi per il tuo cluster. Verrà visualizzato il servizio creato nella sezione precedente. In seguito sarà possibile utilizzare l'ID del servizio o l'ARN completo restituito da questo comando per la descrizione del servizio.

```
aws ecs list-services --cluster MyCluster
```

Output:

```
{
```

```
"taskArns": [
 "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID"
]
}
```

## Descrivere il servizio

Descrivere il servizio utilizzando il seguente comando per ottenere maggiori informazioni sul servizio.

```
aws ecs describe-services --cluster MyCluster --services nginx-service
```

Output:

```
{
 "services": [
 {
 "serviceArn": "arn:aws:ecs:us-east-1:aws_account_id:service/MyCluster/
nginx-service",
 "serviceName": "nginx-service",
 "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/MyCluster",
 "taskDefinition": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/
nginx-task:1",
 "desiredCount": 1,
 "runningCount": 1,
 "pendingCount": 0,
 "launchType": "EC2",
 "status": "ACTIVE",
 "createdAt": "2025-01-13T10:45:00.000Z",
 "events": [
 {
 "id": "abcd1234-5678-90ab-cdef-1234567890ab",
 "createdAt": "2025-01-13T10:45:30.000Z",
 "message": "(service nginx-service) has started 1 tasks: (task
abcd1234-5678-90ab-cdef-1234567890ab)."
 }
]
 }
]
}
```

## Descrivere l'attività in esecuzione

Dopo aver descritto il servizio, eseguire il seguente comando per ottenere maggiori informazioni sull'attività in esecuzione nell'ambito del servizio.

```
aws ecs list-tasks --cluster MyCluster --service-name nginx-service
```

Output:

```
{
 "tasks": [
 {
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/MyCluster/abcd1234-5678-90ab-cdef-1234567890ab",
 "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/MyCluster",
 "taskDefinitionArn": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/nginx-task:1",
 "containerInstanceArn": "arn:aws:ecs:us-east-1:aws_account_id:container-instance/MyCluster/abcd1234-5678-90ab-cdef-1234567890ab",
 "lastStatus": "RUNNING",
 "desiredStatus": "RUNNING",
 "containers": [
 {
 "containerArn": "arn:aws:ecs:us-east-1:aws_account_id:container/MyCluster/abcd1234-5678-90ab-cdef-1234567890ab/abcd1234-5678-90ab-cdef-1234567890ab",
 "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/MyCluster/abcd1234-5678-90ab-cdef-1234567890ab",
 "name": "nginx",
 "lastStatus": "RUNNING",
 "networkBindings": [
 {
 "bindIP": "0.0.0.0",
 "containerPort": 80,
 "hostPort": 80,
 "protocol": "tcp"
 }
]
 }
]
 },
 {
 "createdAt": "2025-01-13T10:45:00.000Z",
 "startedAt": "2025-01-13T10:45:30.000Z"
 }
]
}
```

```
]
}
```

## Testare il server web

Per testare il server web

1. Recuperare l'indirizzo IP pubblico dell'istanza di container eseguendo il seguente comando.

```
aws ec2 describe-instances --instance-ids i-abcd1234 --query
'Reservations[0].Instances[0].PublicIpAddress' --output text
```

Output:

```
203.0.113.25
```

2. Dopo aver recuperato l'indirizzo IP, eseguire il seguente comando `curl` con tale indirizzo.

```
curl http://203.0.113.25
```

Output:

```
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
...
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you can see this page, the nginx web server is successfully installed and
 working.</p>
...
</body>
</html>
```

La pagina di benvenuto di nginx conferma che il servizio è in esecuzione correttamente ed è accessibile da Internet.

# Eeguire la pulizia delle risorse

Per evitare gli addebiti, eliminare le risorse create in questo tutorial.

Per eliminare le risorse

1. Aggiornare il servizio in modo che le attività desiderate siano pari a zero, quindi eliminare il servizio.

```
aws ecs update-service --cluster MyCluster --service nginx-service --desired-count 0
{
 "service": {
 "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/nginx-service",
 "serviceName": "nginx-service",
 "desiredCount": 0,
 "runningCount": 1,
 "pendingCount": 0,
 "status": "ACTIVE"
 }
}
```

2. Attendere l'arresto delle attività in esecuzione, poi eliminare il servizio.

```
aws ecs delete-service --cluster MyCluster --service nginx-service
{
 "service": {
 "serviceArn": "arn:aws:ecs:us-east-1:123456789012:service/MyCluster/nginx-service",
 "serviceName": "nginx-service",
 "status": "DRAINING"
 }
}
```

3. Terminare l'istanza di container creata.

```
aws ec2 terminate-instances --instance-ids i-abcd1234
{
 "TerminatingInstances": [
 {
 "InstanceId": "i-abcd1234",
 "CurrentState": {
```

```
 "Code": 32,
 "Name": "shutting-down"
 },
 "PreviousState": {
 "Code": 16,
 "Name": "running"
 }
}
]
}
```

4. Eliminare il gruppo di sicurezza e la coppia di chiave creati.

```
aws ec2 delete-security-group --group-id sg-abcd1234
aws ec2 delete-key-pair --key-name ecs-tutorial-key
rm ecs-tutorial-key.pem
```

5. Elimina il cluster Amazon ECS.

```
aws ecs delete-cluster --cluster MyCluster
{
 "cluster": {
 "clusterArn": "arn:aws:ecs:us-east-1:123456789012:cluster/MyCluster",
 "clusterName": "MyCluster",
 "status": "INACTIVE"
 }
}
```

## 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 i carichi di lavoro 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 Console di gestione AWS 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 gli eventi di attività Amazon ECS arrestate

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 Amazon Simple Notification Service.

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 Amazon Simple Notification Service e utilizza la tabella seguente per determinare quali opzioni selezionare.

| Opzione | Valore   |  |
|---------|----------|--|
| Tipo    | Standard |  |

| Opzione    | Valore                                            |  |
|------------|---------------------------------------------------|--|
| Name       | TaskStoppedAlert                                  |  |
| Protocollo | Email                                             |  |
| 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"   ],   "detail-type": [     "ECS Task State Change"   ],   "detail": {     "lastStatus": [</pre> |  |

| Opzione                            | Valore                                                                                                                 |  |
|------------------------------------|------------------------------------------------------------------------------------------------------------------------|--|
|                                    | <pre>       "STOPPED"     ],     "stoppedReason": [       "Essential container in task       exited"     ]   } }</pre> |  |
| Target type (Tipo di destinazione) | AWS servizio                                                                                                           |  |
| Target                             | Argomento SNS                                                                                                          |  |
| Topic                              | 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 di traccia dello stack

A partire AWS dalla versione 2.22.0 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: Concatenate Container Logs. FireLens Partial/Split](#)

## 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 di esecuzione di attività.

Sono necessarie le seguenti autorizzazioni:

- `logs:CreateLogStream`
- `logs:CreateLogGroup`
- `logs:PutLogEvents`

## Determinazione dei casi in cui utilizzare l'impostazione del log multilinea

Di seguito sono riportati alcuni esempi di frammenti di log visualizzati 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.model.(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.

- b. 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`

Ad 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`

Ad 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`

Ad 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`

Ad 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`

Ad 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. Scegliere Ruoli e trovare il ruolo del processo ecs-task-role-for-firelens che è stato 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 Console di gestione AWS. 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!
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`

Ad 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`

Ad esempio: `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`

Ad 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`

Ad 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`

Ad 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. Scegliere Ruoli e trovare il ruolo del processo `ecs-task-role-for-firelens` che è stato 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 Console di gestione AWS. 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/\nlocal/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.

## Implementazione di Fluent Bit su container Windows di Amazon ECS

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 sito Web di [Fluent Bit](#).

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](#) il sito Web. GitHub

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 del namespace 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 del namespace dell'host.
- Fluent Bit riceve il record di log sulla porta 24224 perché è esposto al namespace 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, consultare [Installing or updating to the latest version of the 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.

JSON

```
{
 "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 [Retrieving Amazon ECS-Optimized AMI metadata](#).
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 On

```

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 verso 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 DAEMON
```

```
--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 Managed Service Account (gMSA) di gruppo.

Le applicazioni di rete basate su Linux, ad esempio le applicazioni .NET Core, 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. Tuttavia, poiché i container Linux non possono essere aggiunti a un dominio, è necessario configurare un container Linux da eseguire con gMSA.

Un Linux contenitore che viene eseguito con gMSA si basa sul `credentials-fetcher` daemon che viene eseguito sull'istanza Amazon host del contenitore. EC2 Quindi, il daemon recupera le credenziali gMSA dal controller di dominio Active Directory e successivamente le trasferisce

all'istanza di container. Per ulteriori informazioni sugli account di servizio, consulta [Crea gMSAs per i container Windows](#) nel sito Web Microsoft Learn.

## Considerazioni

Prima di utilizzare gMSA per i container Linux, valuta quanto segue:

- Se i tuoi container funzionano EC2, puoi utilizzarli gMSA per Windows container e container. Linux Per informazioni su come utilizzare il gMSA per container Linux su Fargate, consultare [Utilizzo del gMSA per i container Linux su Fargate](#).
- Potrebbe essere necessario un computer Windows aggiunto al dominio per completare i prerequisiti. Ad esempio, potresti aver bisogno di un computer Windows aggiunto al dominio per creare gMSA in Active Directory con PowerShell. Gli PowerShell strumenti RSAT Active Director sono disponibili solo per Windows. Per ulteriori informazioni, consulta [Installazione degli strumenti di amministrazione Active Directory](#).
- Hai scelto tra la modalità gMSA senza dominio e l'aggiunta di ogni istanza in un unico dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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 di dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per gMSA senza dominio.

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container. Il file CredSpec viene generato e quindi archiviato in una delle opzioni di archiviazione di CredSpec nella tabella seguente, specifica del sistema operativo delle istanze di container. Per utilizzare il metodo senza dominio, una sezione facoltativa del file CredSpec può specificare le credenziali in una delle opzioni di archiviazione domainless user credentials riportate nella tabella seguente, in base al sistema operativo delle istanze di container.

| Posizione di archiviazione    | Linux                            | Windows                          |
|-------------------------------|----------------------------------|----------------------------------|
| Amazon Simple Storage Service | CredSpec                         | CredSpec                         |
| Gestione dei segreti AWS      | credenziali utente senza dominio | credenziali utente senza dominio |

| Posizione di archiviazione                 | Linux    | Windows                                    |
|--------------------------------------------|----------|--------------------------------------------|
| Amazon EC2 Systems Manager Parameter Store | CredSpec | CredSpec, credenziali utente senza dominio |
| File locale                                | N/D      | CredSpec                                   |

## Prerequisiti

Prima di utilizzare la funzionalità gMSA per container Linux con Amazon ECS, assicurati di completare le seguenti operazioni:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un Directory Service Active Directory. 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-premises. 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 account gMSA nell'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](#) su GitHub

- Configuri le credenziali per l'autenticazione del daemon `credentials-fetcher` con Active Directory. Le credenziali devono essere membri del gruppo di sicurezza di Active Directory che ha accesso all'account gMSA. Sono disponibili diverse opzioni in [Decidi se aggiungere le istanze al dominio o utilizzare gMSA senza dominio](#).

- 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, sono necessarie le autorizzazioni IAM per il ruolo di esecuzione dell'attività. Gestione dei segreti AWS
  - 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 di container Linux compatibili con gMSA in 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 vengono fornite le credenziali iniziali e configura i dati EC2 utente in un modello di EC2 avvio riutilizzabile per installare il daemon. `credentials-fetcher`

#### 1. Decidi se aggiungere le istanze al dominio o utilizzare gMSA senza dominio.

- Unisci le EC2 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 [Aggiunta di un Active Directory o a un dominio FreeIPA](#) in Fedora Docs.

- Creazione di un utente Active Directory per gMSA senza dominio

Il daemon `credentials-fetcher` ha una funzionalità chiamata gMSA senza dominio. Questa funzionalità richiede un dominio, ma l'EC2istanza non deve essere aggiunta al dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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. Al contrario, fornisci il nome di un segreto in Gestione dei segreti AWS nel file CredSpec. 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 funzionalità gMSA support for non-domain-joined container hosts. Per ulteriori informazioni sulla funzionalità di Windows, consulta [Architettura e miglioramenti di gMSA](#) sul sito Web Microsoft Learn.

- a. Crea un utente nel dominio di Active Directory. L'utente di Active Directory deve disporre dell'autorizzazione per accedere agli account del servizio di gMSA utilizzati nelle attività.
- b. Crea un account segreto in Gestione dei segreti AWS, dopo aver creato l'utente in Active Directory. Per ulteriori informazioni, consulta [Creare un Gestione dei segreti AWS 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. Aggiungi la configurazione al file CredSpec 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. La funzionalità gMSA senza dominio richiede autorizzazioni aggiuntive nel ruolo di esecuzione dell'attività. Segui il passaggio [\(Facoltativo\) segreto gMSA senza dominio](#).

## 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 o lo script `bash`. 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"
```

- Script bash

Se hai più dimestichezza con gli script bash e disponi di più variabili da scrivere per `/etc/ecs/ecs.config`, utilizza il seguente formato heredoc. 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) segreto gMSA senza dominio

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`.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": "arn:aws:secretsmanager:us-east-1:123456789012:secret:my-secret-AbCdEf"
 }
]
}
```

#### 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. Decidere 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 utilizzi gMSA senza dominio, 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.

JSON

```
{
 "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.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ssm:GetParameters"
],
 "Resource": "arn:aws:ssm:us-east-1:123456789012:parameter/my-parameter"
 }
]
}
```

## File di specifica delle credenziali

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container Linux. Genera il file CredSpec e fai riferimento a esso nel campo `credentialSpecs` della definizione delle attività. Il file CredSpec non contiene segreti.

Di seguito è riportato un esempio del file CredSpec.

```
{
 "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"
 }
 }
}
```

## Creazione di una CredSpec

Crea un file CredSpec utilizzando il modulo PowerShell CredSpec su un computer Windows che fa parte del dominio. Segui i passaggi descritti in [Creazione di una specifica delle credenziali](#) sul sito Web Microsoft Learn.

## Utilizzo del gMSA per i container Linux su Fargate

Amazon ECS supporta l'autenticazione di Active Directory per i container Linux su Fargate tramite un tipo speciale di account di servizio chiamato account di servizio gestito del gruppo (gMSA).

Le applicazioni di rete basate su Linux, ad esempio le applicazioni .NET Core, 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. Tuttavia, poiché i container Linux non possono essere aggiunti a un dominio, è necessario configurare un container Linux da eseguire con gMSA.

## Considerazioni

Prima di utilizzare il gMSA per i container Linux su Fargate, valutare quanto segue:

- È necessario eseguire la versione della piattaforma 1.4 o successiva.
- Potrebbe essere necessario un computer Windows aggiunto al dominio per completare i prerequisiti. Ad esempio, potresti aver bisogno di un computer Windows aggiunto al dominio per creare gMSA in Active Directory con PowerShell. Gli PowerShell strumenti RSAT Active Director

sono disponibili solo per. Windows Per ulteriori informazioni, consulta [Installazione degli strumenti di amministrazione Active Directory](#).

- È necessario utilizzare un gMSA senza dominio.

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container. È possibile generare il file CredSpec e poi archiviarlo in un bucket Amazon S3.

- Un'attività può supportare solo un Active Directory.

## Prerequisiti

Prima di utilizzare la funzionalità gMSA per container Linux con Amazon ECS, assicurati di completare le seguenti operazioni:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un Directory Service Active Directory. 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-premises. 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 account gMSA esistente in Active Directory e di un utente con l'autorizzazione di accesso all'account di servizio gMSA. Per ulteriori informazioni, consulta [Creazione di un utente Active Directory per gMSA senza dominio](#).
- Disponi di un bucket Amazon S3. Per ulteriori informazioni, consultare [Creating a bucket](#) nella Guida per l'utente di Amazon S3.

## Configurazione di container Linux compatibili con gMSA in Amazon ECS

### Preparazione dell'infrastruttura

I passaggi seguenti sono considerazioni e configurazioni che vengono eseguite una sola volta.

- Creazione di un utente Active Directory per gMSA senza dominio

Quando si utilizza un gMSA senza dominio, il container non viene aggiunto al dominio. Le altre applicazioni eseguite sul container non possono utilizzare le credenziali per accedere al dominio.



Le attività che utilizzano un dominio diverso possono essere eseguite sullo stesso container. Fornisci il nome di un segreto Gestione dei segreti AWS nel CredSpec file. Il segreto deve contenere un nome utente, una password e un dominio a cui accedere.

Questa funzionalità è simile alla funzionalità gMSA support for non-domain-joined container hosts. Per ulteriori informazioni sulla funzionalità di Windows, consulta [Architettura e miglioramenti di gMSA](#) sul sito Web Microsoft Learn.

- a. Configurare un utente nel dominio di Active Directory. L'utente di Active Directory deve disporre dell'autorizzazione di accesso all'account di servizio gMSA utilizzato nelle attività.
- b. Disponi di un VPC e delle sottoreti in grado di risolvere il nome di dominio Active Directory. Configurare il VPC con opzioni DHCP con il nome di dominio che indirizza al nome del servizio Active Directory. Per informazioni su come configurare le opzioni DHCP per un VPC, consultare [Work with DHCP option sets](#) nella Guida per l'utente di Amazon Virtual Private Cloud.
- c. Crea un segreto in Gestione dei segreti AWS.
- d. Creare il file di specifica delle credenziali.

## Configurazione di autorizzazioni e segreti

Eseguire i passaggi descritti di seguito 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. Crea un utente nel dominio di Active Directory. L'utente di Active Directory deve disporre dell'autorizzazione per accedere agli account del servizio di gMSA utilizzati nelle attività.
2. Dopo aver creato l'utente di Active Directory, crea un account segreto in Gestione dei segreti AWS. Per ulteriori informazioni, consulta [Creazione di un segreto Gestione dei segreti AWS](#).
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`.

## JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "secretsmanager:GetSecretValue"
],
 "Resource": [
 "arn:aws:secretsmanager:us-east-1:111122223333: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.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "VisualEditor",
 "Effect": "Allow",
 "Action": [
 "s3:GetObjectVersion",
 "s3:ListBucket"
],
 "Resource": [
 "arn:aws:s3:::{bucket_name}",
 "arn:aws:s3:::{bucket_name}/{object}"
]
 }
]
}
```

## File di specifica delle credenziali

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container Linux. Genera il file CredSpec e fai riferimento a esso nel campo `credentialSpecs` della definizione delle attività. Il file CredSpec non contiene segreti.

Di seguito è riportato un esempio del file CredSpec.

```
{
 "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"
 }
 }
}
}

```

## Creazione di CredSpec e caricamento su Amazon S3

Crea un file CredSpec utilizzando il modulo PowerShell CredSpec su un computer Windows che fa parte del dominio. Segui i passaggi descritti in [Creazione di una specifica delle credenziali](#) sul sito Web Microsoft Learn.

Dopo aver creato il file di specifica delle credenziali, caricarlo su un bucket Amazon S3. Copia il file CredSpec nel computer o nell'ambiente in cui esegui i comandi AWS CLI .

Esegui il AWS CLI comando seguente per CredSpec caricarlo 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 senza dominio gMSA utilizzando 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 gMSA senza dominio, l'istanza di container non viene 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: creazione e configurazione dell'account gMSA su Active Directory Domain Services \(AD DS\)](#)
- [Fase 2: caricamento delle credenziali in Secrets Manager](#)
- [Fase3: modifica del codice JSON CredSpec per includere informazioni relative a gMSA senza dominio](#)
- [Fase 4: caricamento di 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](#)
- [Fase 9: registrazione di una definizione di attività che utilizza gMSA senza dominio](#)
- [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](#)

- [Fase 13: verifica che il container disponga delle credenziali gMSA](#)
- [Fase 14: pulizia](#)
- [Debug di gMSA senza dominio in Amazon ECS per i container 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](#).
- L'utente IAM ha le autorizzazioni necessarie specificate nell'esempio di policy IAM [Amazon ECS FullAccess](#).
- 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. Directory Service 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-premises. 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 la modalità gMSA senza dominio e l'aggiunta di ogni istanza in un unico dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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 di dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per gMSA senza dominio.

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container. Il file CredSpec viene generato e quindi archiviato in una delle opzioni di archiviazione di CredSpec nella tabella seguente, specifica del sistema operativo delle istanze di container. Per utilizzare il metodo senza dominio, una sezione facoltativa del file CredSpec può specificare le credenziali in una delle opzioni di archiviazione domainless user credentials riportate nella tabella seguente, in base al sistema operativo delle istanze di container.

| Posizione di archiviazione                 | Linux                            | Windows                                    |
|--------------------------------------------|----------------------------------|--------------------------------------------|
| Amazon Simple Storage Service              | CredSpec                         | CredSpec                                   |
| Gestione dei segreti AWS                   | 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: creazione e configurazione dell'account gMSA su Active Directory Domain Services (AD DS)

Crea e configura un account gMSA nel dominio Active Directory.

### Note

Questa fase crea due account separati: un account di servizio gestito del gruppo (gMSA) che fornisce l'identità per i container e un account utente normale utilizzato per l'autenticazione del dominio. Questi account hanno scopi diversi e devono avere nomi diversi.

## 1. Genera una chiave principale del servizio di distribuzione delle chiavi

### Note

Se lo stai utilizzando Directory Service, puoi saltare questo passaggio.

La chiave principale KDS e le autorizzazioni gMSA sono configurate con Microsoft AD gestito da AWS .

Se non hai ancora creato un account del servizio gMSA nel tuo dominio, dovrai prima generare una chiave principale del servizio di distribuzione delle chiavi (KDS). Il KDS è responsabile della creazione, della rotazione e del rilascio della password gMSA agli host autorizzati. Quando `ccg.exe` ha bisogno di recuperare le credenziali gMSA, contatta KDS per recuperare la password attuale.

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 l'account gMSA

Per creare l'account gMSA e consentire il `ccg.exe` recupero della gMSA password, esegui i seguenti PowerShell comandi da un server o client Windows con accesso al dominio. Sostituire `ExampleAccount` con il nome desiderato per l'account gMSA e sostituire `example-domain` con il nome del dominio Active Directory (ad esempio, se il dominio è `contoso.com`, utilizzare `contoso`).



- 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 "example-domain" -ServicePrincipalNames "host/ExampleAccount", "host/example-domain" -PrincipalsAllowedToRetrieveManagedPassword "ExampleAccountHosts"
```
- d. Crea un utente con una password permanente senza scadenza. Queste credenziali vengono archiviate Gestione dei segreti AWS e utilizzate da ogni attività per accedere al dominio. Si tratta di un account utente separato dall'account gMSA creato in precedenza. Sostituire ExampleServiceUser con il nome desiderato per questo account utente di servizio.

```
PS C:\> New-ADUser -Name "ExampleServiceUser" -AccountPassword (ConvertTo-SecureString -AsPlainText "Test123" -Force) -Enabled 1 -PasswordNeverExpires 1
```

- e. 

```
PS C:\> Add-ADGroupMember -Identity "ExampleAccountHosts" -Members "ExampleServiceUser"
```
- f. Installa il PowerShell modulo per la creazione di CredSpec oggetti in Active Directory e genera il codice 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. Si tratta del file CredSpec che viene utilizzato nella fase 3, [Fase3: modifica del codice JSON CredSpec per includere informazioni relative a gMSA senza dominio](#).

## 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. Questo è il metodo gMSA senza dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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. Utilizzare il nome dell'account utente di servizio (non il nome dell'account gMSA) come nome utente. Mantieni l'ARN del segreto da utilizzare nella fase successiva, [Fase3: modifica del codice JSON CredSpec per includere informazioni relative a gMSA senza dominio](#)

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\":\"ExampleServiceUser\",\"password\":\"Test123\",\
\"domainName\":\"contoso.com\"}"
```

## Fase3: modifica del codice JSON CredSpec per includere informazioni relative a gMSA senza dominio

Prima di caricare CredSpec su una delle opzioni di archiviazione, aggiungi le informazioni a 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 file CredSpec 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. Il file CredSpec modificato sarà simile all'esempio seguente:

```
{
 "CmsPlugins": [
 "ActiveDirectory"
],
 "DomainJoinConfig": {
 "Sid": "S-1-5-21-4066351383-705263209-1606769140",
 "MachineAccountName": "ExampleAccount",
 "Guid": "ac822f13-583e-49f7-aa7b-284f9a8c97b6",
 "DnsTreeName": "example-domain",
 "DnsName": "example-domain",
 "NetBiosName": "example-domain"
 },
 "ActiveDirectoryConfig": {
 "GroupManagedServiceAccounts": [
 {
 "Name": "ExampleAccount",
 "Scope": "example-domain"
 },
 {
 "Name": "ExampleAccount",
 "Scope": "example-domain"
 }
]
 },
 "HostAccountConfig": {
 "PortableCcgVersion": "1",
 "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
```

```
"PluginInput": "{\\"credentialArn\\": \\"arn:aws:secretsmanager:aws-
region:111122223333:secret:gmsa-plugin-input\\"}"
}
}
}
```

## Fase 4: caricamento di 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 file CredSpec nel computer o nell'ambiente in cui esegui i comandi AWS CLI .
2. Esegui il AWS CLI comando seguente per CredSpec caricarlo 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 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 Console di gestione AWS, AWS CLI, SDKs, o CloudFormation. Le impostazioni e la configurazione nel cluster non influiscono su 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`.

JSON

```
{
 "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:us-east-1:111122223333:secret:gmsa-plugin-AbCdEf"
 }
]
}
```

## 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 ACS 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:

JSON

```
{
 "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:

JSON

```
{
 "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`.

## Fase 9: registrazione di una definizione di attività che utilizza gMSA senza dominio

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 Console di gestione AWS, consulta [Avvio di un'istanza di container Windows di Amazon ECS](#). Fermati alla fase relativa ai dati utente.
2. Per quanto riguarda gMSA, i dati utente devono impostare la variabile di ambiente ECS\_GMSA\_SUPPORTED prima di avviare l'agente del container 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 Console di gestione AWS. Tuttavia, gMSA necessita di funzionalità aggiuntive indicate come attributi. Questi attributi non sono visibili in Console di gestione AWS, 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
```

## Fase 13: verifica che il container disponga delle credenziali gMSA

Verifica che il container dell'attività abbia un token Kerberos. 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 il file CredSpec 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
```

## Debug di gMSA senza dominio in Amazon ECS per i container 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 Console di gestione AWS, 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 [Visualizza gli errori relativi alle attività interrotte in Amazon ECS](#). Console di gestione AWS

## Eventi Windows

Gli eventi di Windows per gMSA nei container vengono registrati nel file di log Microsoft-Windows-Containers-CCG e sono disponibili all'interno del 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.

## Plug-in gMSA per l'agente ECS

La registrazione del plug-in gMSA per l'agente ECS sull'istanza di container 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 di gMSA per container 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 la modalità gMSA senza dominio e l'aggiunta di ogni istanza in un unico dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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 di dati per CredSpec e, facoltativamente, per le credenziali utente di Active Directory per gMSA senza dominio.

Amazon ECS utilizza un file di specifica delle credenziali di Active Directory (CredSpec). Questo file contiene i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container. Il file CredSpec viene generato e quindi archiviato in una delle opzioni di archiviazione di CredSpec nella tabella seguente, specifica del sistema operativo delle istanze di container. Per utilizzare il metodo senza dominio, una sezione facoltativa del file CredSpec può specificare le credenziali in



una delle opzioni di archiviazione domainless user credentials riportate nella tabella seguente, in base al sistema operativo delle istanze di container.

| Posizione di archiviazione                 | Linux                            | Windows                                    |
|--------------------------------------------|----------------------------------|--------------------------------------------|
| Amazon Simple Storage Service              | CredSpec                         | CredSpec                                   |
| Gestione dei segreti AWS                   | 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 la funzionalità gMSA per container Windows con Amazon ECS, assicurati di completare le seguenti operazioni:

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un Directory Service Active Directory. 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-premises. 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 account gMSA nell'Active Directory. Per ulteriori informazioni, consulta [Create g MSAs for Windows containers](#).
- Hai scelto di utilizzare gMSA senza dominio o l'istanza di container Amazon ECS Windows che ospita il processo Amazon ECS deve essere un dominio aggiunto ad Active Directory ed essere membro del gruppo di sicurezza di Active Directory che ha accesso all'account gMSA.

Utilizzando gMSA senza dominio, l'istanza di container non viene 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, Gestione dei segreti AWS sono necessarie le autorizzazioni IAM per il ruolo dell'istanza Amazon. EC2
  - 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 di gMSA per container Windows su Amazon ECS

Per configurare gMSA per container Windows su Amazon ECS, puoi seguire il tutorial completo che include la configurazione dei prerequisiti [Utilizzo di contenitori Amazon ECS Windows senza dominio gMSA utilizzando AWS CLI](#).

Le seguenti sezioni illustrano la configurazione di CredSpec in dettaglio.

### Argomenti

- [Esempio CredSpec](#)
- [Configurazione di gMSA senza dominio](#)
- [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 i metadati gMSA utilizzati per propagare il contesto dell'account gMSA al container 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"
 }
]
 }
}

```

## Configurazione di gMSA senza dominio

Consigliamo gMSA senza dominio anziché aggiungere le istanze di container a un singolo dominio. Utilizzando gMSA senza dominio, l'istanza di container non viene 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 CredSpec su una delle opzioni di archiviazione, aggiungi le informazioni a 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.

### Formato di credenziali gMSA senza dominio

Di seguito è riportato il formato JSON per le credenziali gMSA senza dominio 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 file CredSpec 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. Il file CredSpec modificato sarà simile all'esempio seguente:

```
{
 "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`: o `domainlesscredentialSpec`:, a seconda che tu stia unendo rispettivamente le istanze di container a un singolo dominio o utilizzando gMSA senza dominio.

### 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.

## JSON

```
{
 "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.

JSON

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Effect": "Allow",
 "Action": [
 "ssm:GetParameters"
],
 "Resource": "arn:aws:ssm:*:111122223333:parameter/example*"
 }
]
}
```

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 ottimizzato AMIs perché è preconfigurato con i requisiti e i consigli per eseguire i carichi di lavoro dei container. In alcuni casi, potrebbe essere necessario personalizzare l'AMI per integrare software aggiuntivi. È possibile utilizzare EC2 Image Builder per la creazione, la gestione e la distribuzione delle immagini del server. La proprietà delle immagini personalizzate create nell'account viene mantenuta. 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.

Viene creata una ricetta per l'immagine. La ricetta include un'immagine principale ed eventuali componenti aggiuntivi. Inoltre, viene creata una pipeline che distribuisce l'AMI personalizzata.

Viene creata una ricetta per l'immagine. Una ricetta di immagine di Image Builder è un documento che definisce l'immagine di base e i componenti applicati all'immagine di base per produrre la configurazione desiderata per l'immagine AMI di output. Inoltre, viene creata una pipeline che distribuisce l'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 ottimizzata per Amazon ECS
  - Windows 2022 Core x86 ottimizzata per Amazon ECS



- Windows 2019 Full x86 ottimizzata per Amazon ECS
- Windows 2019 Core x86 ottimizzata per Amazon ECS
- Windows 2016 Full x86 ottimizzata per Amazon ECS

Consigliamo anche di selezionare l'opzione "Usa la versione del sistema operativo più recente". Per l'immagine parent, la pipeline utilizzerà il controllo delle versioni semantico, che aiuta a rilevare gli aggiornamenti delle dipendenze nei processi con pianificazione automatica. 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 si utilizza un ID AMI come immagine principale nella ricetta di immagine, è necessario 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 pubblicamente. Sono necessarie le autorizzazioni corrette per accedere all'immagine. Quando si utilizza l'ARN di un'immagine anziché un'AMI nella ricetta Image Builder, la pipeline utilizza automaticamente la versione più recente dell'immagine parent ogni volta che viene eseguita. Questo approccio elimina la necessità di creare manualmente nuove versioni della ricetta per ogni aggiornamento.

## Pulizia delle AMI ottimizzate per Amazon ECS

Quando si utilizza un'AMI ottimizzata per Amazon ECS come immagine principale, è necessario ripulire l'immagine per evitare problemi transitori. L'AMI ottimizzata per Amazon ECS è preconfigurata per l'avvio automatico dell'agente Amazon ECS e la registrazione come istanza di container con Amazon ECS. Utilizzarla come immagine di base senza una pulizia adeguata può causare problemi nell'AMI personalizzata.

Per ripulire l'immagine per utilizzi futuri, creare un componente che esegua i seguenti comandi per arrestare il pacchetto ecs-init e i processi Docker:

```
sudo systemctl stop ecs
sudo systemctl stop docker
```

Rimuovere tutti i file di log dall'istanza attuale per evitare che vengano preservati con il salvataggio dell'immagine. Utilizzate lo script di esempio nelle [best practice di sicurezza per EC2 Image Builder](#) per pulire i vari file dall'istanza.

Per eliminare i dati specifici di Amazon ECS, eseguire i seguenti comandi:

```
sudo rm -rf /var/log/ecs/*
sudo rm /var/lib/ecs/data/agent.db
```

Per ulteriori informazioni sulla creazione di un'AMI personalizzata ottimizzata per Amazon ECS AMIs, consulta [Come posso creare un'AMI personalizzata da un'AMI ottimizzata per Amazon ECS?](#) nel Knowledge Center. AWS

## Utilizzo dell'ARN dell'immagine con infrastructure as code (IaC)

È possibile configurare la ricetta utilizzando la console EC2 Image Builder o l'infrastruttura come codice (ad esempio 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. Il formato ARN dell'immagine è il seguente:

```
arn:${Partition}:imagebuilder:${Region}:${Account}:image/${ImageName}/${ImageVersion}
```

Il formato di ImageVersion è il seguente. Sostituisci *minor* e *patch* con *major* 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 risolversi automaticamente nella versione più recente dell'immagine. L'utilizzo di un ARN senza versione garantisce che il riferimento punti sempre all'immagine più recente disponibile, semplificando il processo di mantenimento delle immagini aggiornate nell'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'immagine più recente disponibile. Consigliamo di creare uno script automatico per filtrare e visualizzare solo le immagini in linea con i criteri desiderati. Il seguente script Python mostra come recuperare un elenco di contenuti ottimizzati per Amazon ECS. AMLs

Lo script accetta due argomenti facoltativi: `owner` e `platform`, i cui valori predefiniti sono rispettivamente “Amazon” e “Windows”. I valori validi per l'argomento `owner` sono `Self`, `Shared`, `Amazon` e `ThirdParty`. I valori validi per l'argomento `platform` sono `Windows` e `Linux`. Ad esempio, se lo script viene eseguito con l'argomento `owner` impostato su `Amazon` e `platform` impostata su `Linux`, genera un elenco di immagini pubblicate da Amazon, incluse immagini ottimizzate per Amazon ECS.

```
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 CloudFormation

Una ricetta di immagine Image Builder è un blueprint che specifica l'immagine principale e i componenti necessari per ottenere la configurazione prevista dell'immagine di output. Viene utilizzata la risorsa `AWS::ImageBuilder::ImageRecipe`. Impostare il valore `ParentImage` sull'ARN dell'immagine. Utilizzare l'ARN senza versione dell'immagine desiderata per assicurare che la pipeline usi 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 della risorsa `AWS::ImageBuilder::ImageRecipe` utilizza l'ARN di un'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 risorsa [AWS::ImageBuilder::ImageRecipe](#), consultare la Guida per l'utente di AWS CloudFormation .

È possibile automatizzare la creazione di nuove immagini nella pipeline impostando la proprietà `Schedule` della risorsa `AWS::ImageBuilder::ImagePipeline`. La proprietà di pianificazione `schedule` 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'esempio seguente di `AWS::ImageBuilder::ImagePipeline` fa in modo che la pipeline esegua una build ogni giorno alle 10:00 UTC (Coordinated Universal Time). Impostare i seguenti valori di `Schedule`:

- Imposta `PipelineExecutionStartCondition` su `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`. La build viene avviata solo se vengono aggiornate risorse dipendenti come l'immagine principale o i componenti, che utilizzano il carattere jolly “x” nelle proprie versioni semantiche. In questo modo si garantisce che la build incorpori gli aggiornamenti più recenti 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'ARN dell'immagine con Terraform

L'approccio per specificare l'immagine principale e la pianificazione della pipeline in Terraform è in linea con quello in AWS CloudFormation. Viene utilizzata la risorsa `aws_imagebuilder_image_recipe`. Impostare il valore `parent_image` sull'ARN dell'immagine. Utilizzare l'ARN senza versione dell'immagine desiderata per assicurare che la pipeline usi sempre la versione più recente dell'immagine. Per ulteriori informazioni, consultare [aws\\_imagebuilder\\_image\\_recipe](#) nella documentazione di Terraform.

Nel blocco di configurazione della programmazione di `aws_imagebuilder_image_pipeline` resource, impostare il valore dell'argomento `schedule_expression` su un'espressione Cron a scelta per specificare la frequenza di esecuzione della pipeline e impostare `pipeline_execution_start_condition` su `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`. Per ulteriori informazioni, consultare [aws\\_imagebuilder\\_image\\_pipeline](#) nella documentazione di Terraform.

## Utilizzo di AWS Deep Learning Containers su Amazon ECS

AWS 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 senza Elastic Inference su Amazon ECS, consultare [Amazon ECS setup](#) nella Guida per gli sviluppatori di AWS Deep Learning AMIs .

## Tutorial: creazione di un servizio utilizzando una distribuzione blu/verde

Amazon ECS ha integrato le blue/green distribuzioni nella procedura guidata Create Service sulla console Amazon ECS. Per ulteriori informazioni, consulta [Creazione di un'implementazione di aggiornamenti continui di Amazon ECS](#).

Il seguente tutorial mostra come creare un servizio Amazon ECS contenente un'attività Fargate che utilizza blue/green il tipo di distribuzione con. AWS CLI

### Note

È stato aggiunto il supporto per l'esecuzione di una blue/green distribuzione per CloudFormation. Per ulteriori informazioni, consulta [Esegui le distribuzioni blu/green di Amazon ECS tramite l' utilizzo CodeDeploy 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](#) di. AWS Command Line Interface
- Hai completato le fasi descritte in [Configurazione per l'uso di Amazon ECS](#).
- L'utente IAM dispone delle autorizzazioni necessarie 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 Amazon ECS che utilizzano il tipo di blue/green implementazione richiedono l'uso di un Application Load Balancer o 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
```

```
--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
```

## 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" \
 --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 `serviceRoleArn`, 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 distribuzione CodeDeploy

Utilizza 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 le risorse create in precedenza nel tutorial.

```
version: 0.0
Resources:
 - TargetService:
 Type: AWS::ECS::Service
 Properties:
 TaskDefinition: "arn:aws:ecs:region:aws_account_id:task-
definition/first-run-task-definition:7"
 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"
}
```

- c. 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
```

Continuare a recuperare i dettagli di distribuzione fino a quando lo stato è Succeeded, come mostrato nel seguente output.

```
{
 "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": 1543431490.226,
 "lifecycleEvents": [
 {
 "lifecycleEventName": "BeforeInstall",
 "startTime": 1543431361.022,
 "endTime": 1543431361.433,
 "status": "Succeeded"
 },
 {
 "lifecycleEventName": "Install",
 "startTime": 1543431361.678,
```



```

 "endTime": 1543431485.275,
 "status": "Succeeded"
 },
 {
 "lifecycleEventName": "AfterInstall",
 "startTime": 1543431485.52,
 "endTime": 1543431486.033,
 "status": "Succeeded"
 },
 {
 "lifecycleEventName": "BeforeAllowTraffic",
 "startTime": 1543431486.838,
 "endTime": 1543431487.483,
 "status": "Succeeded"
 },
 {
 "lifecycleEventName": "AllowTraffic",
 "startTime": 1543431487.748,
 "endTime": 1543431488.488,
 "status": "Succeeded"
 },
 {
 "lifecycleEventName": "AfterAllowTraffic",
 "startTime": 1543431489.152,
 "endTime": 1543431489.885,
 "status": "Succeeded"
 }
],
"status": "Succeeded",
"taskSetsInfo": [
 {
 "identifer": "ecs-svc/9223370493425779968",
 "desiredCount": 1,
 "pendingCount": 0,
 "runningCount": 1,
 "status": "ACTIVE",
 "trafficWeight": 0.0,
 "targetGroup": {
 "name": "bluegreentarget1"
 }
 },
 {
 "identifer": "ecs-svc/9223370493423413672",
 "desiredCount": 1,

```

```
 "pendingCount": 0,
 "runningCount": 1,
 "status": "PRIMARY",
 "trafficWeight": 100.0,
 "targetGroup": {
 "name": "bluegreentarget2"
 }
 }
]
}
}
```

## Fase 7: eliminare

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/appspect.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: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
```

# Service Quotas di Amazon ECS

Quando crei le tue Account AWS, impostiamo quote predefinite (note anche come limiti) sulle 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 relative alle quote che ti mettiamo a disposizione 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 Amazon ECS e Fargate, consulta [Endpoint e quote Amazon ECS](#) nella 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](#).

Le istanze gestite di Amazon ECS seguono le quote Amazon EC2 . Per ulteriori informazioni, consulta le [quote dei EC2 servizi Amazon](#) nella Amazon EC2 User Guide.

## Gestione delle quote di Amazon ECS e dei AWS Fargate servizi in Console di gestione AWS

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.

### Console di gestione AWS

Come visualizzare le quote di servizio di Amazon ECS e Fargate tramite la Console di gestione AWS

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 di più con le quote di servizio, Console di gestione AWS 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](#).

## Gestione delle service quotas Amazon ECS e dei 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 quali il bilanciamento del carico dei servizi, Service Connect, il networking delle attività e il dimensionamento automatico dei cluster. Amazon ECS e gli altri Servizi AWS con cui si integra mantengono tutti le quote di servizio e i limiti di frequenza 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 i costi.

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 la sezione relativa alla [Richiesta di limitazione delle richieste API Amazon ECS](#).

Quando ridimensioni i carichi di lavoro su Amazon ECS, ti consigliamo di tenere presente la quota di servizio seguente.

- AWS Fargate prevede quote che limitano il numero di attività simultanee in esecuzione in ciascuna di esse. Regione AWS Esistono quote per le attività on demand e Fargate Spot su Amazon ECS. Ogni quota di servizio comprende 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 usi il dimensionamento automatico del cluster Amazon ECS con un provider di capacità del gruppo Auto Scaling, quando scali i tuoi servizi considera la quota `Tasks in the PROVISIONING state per cluster`. Questa quota è il numero massimo di attività nello stato `PROVISIONING` per ogni cluster per le quali i provider di capacità possono aumentare la capacità. Quando avvii un gran numero di attività contemporaneamente, puoi facilmente raggiungere la quota. Un esempio è l'implementazione simultanea di decine di servizi, ognuno con centinaia di attività. Quando ciò accade, il provider di capacità deve avviare nuove istanze di container per collocare le attività quando il cluster non dispone di capacità sufficiente. 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 di una capacità insufficiente del cluster. Il pianificatore del servizio Amazon ECS implementa una strategia di back-off e limitazione esponenziale per riprovare il posizionamento delle attività man mano che vengono avviate nuove istanze di container. Di conseguenza, potrebbero verificarsi tempi di implementazione o di aumento orizzontale più lenti. Per evitare questa situazione, puoi pianificare le implementazioni dei servizi in uno dei seguenti modi. Implementa un gran numero di attività che non richiedono un aumento della capacità del cluster oppure mantieni la capacità del cluster di riserva per l'avvio 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 che sono integrati con Amazon ECS.

## Bilanciatore del carico elastico

Puoi configurare i servizi Amazon ECS in modo da usare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività. Per ulteriori informazioni e best practice consigliate su come scegliere un bilanciatore del carico, consulta [Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS](#).

### Quote di servizio Elastic Load Balancing

Quando scali i carichi di lavoro, tieni conto delle seguenti quote del servizio Elastic Load Balancing. La maggior parte delle quote del servizio Elastic Load Balancing sono regolabili ed è possibile richiedere un aumento nella console Service Quotas.

#### Application Load Balancer

Quando usi un Application Load Balancer, a seconda del caso d'uso, potresti dover richiedere un aumento della quota per:

- La quota **Targets** per **Application Load Balancer**, ossia il numero di destinazioni alla base dell'**Application Load Balancer**.
- La quota **Targets** per **Target Group** per **Region**, ossia il numero di destinazioni alla base dei gruppi di destinazioni.

Per ulteriori informazioni, consulta [Quote per Application Load Balancer](#) nella Guida per l'utente di **Application Load Balancer**.

## Network Load Balancer

Ci sono limitazioni più rigorose sul numero di destinazioni che è possibile registrare con un **Network Load Balancer**. Quando si usa un **Network Load Balancer**, spesso si vuole abilitare il supporto tra zone, che comporta limitazioni di dimensionamento aggiuntive sui **Targets** per **Availability Zone** Per **Network Load Balancer**, ovvero sul numero massimo di destinazioni per zona di disponibilità per ogni **Network Load Balancer**. Per ulteriori informazioni, consulta [Quote per i Network Load Balancer](#) nella Guida per l'utente per i **Network Load Balancer**.

## Limitazione dell'API Elastic Load Balancing

Quando configuri un servizio Amazon ECS per usare un bilanciatore del carico, i controlli dell'integrità del gruppo di destinazione devono essere superati prima che il servizio venga considerato integro. Per eseguire questi controlli dell'integrità, Amazon ECS invoca le operazioni dell'API Elastic Load Balancing per tuo conto. Se hai un gran numero di servizi configurati con bilanciatori del carico nel tuo account, potresti riscontrare implementazioni dei servizi più lente a causa di una potenziale limitazione specifica per le operazioni **RegisterTarget**, **DeregisterTarget** e **DescribeTargetHealth** dell'API Elastic Load Balancing. Quando si verifica una limitazione, compaiono 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 sulla risoluzione di tali errori di limitazione della larghezza di banda della rete, consulta [Gestisci i problemi di limitazione di Amazon ECS](#).

## Interfacce di rete elastiche

Con le tue attività, usa la modalità di rete **aws-vpc**, Amazon ECS fornisce un'interfaccia di rete elastica (ENI) unica per ogni attività. Quando i tuoi servizi Amazon ECS usano un bilanciatore del carico Elastic Load Balancing, queste interfacce di rete vengono registrate anche come destinazioni per il gruppo di destinazione appropriato definito nel servizio.



## Quote di servizio dell'interfaccia di rete elastica

Quando esegui attività che usano la modalità di rete `awsvpc`, a ciascuna attività viene collegata un'interfaccia elastica di rete unica. Se tali attività devono essere eseguite tramite Internet, assegna un indirizzo IP pubblico all'interfaccia di rete elastica per tali attività. Quando scali 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 quota `Elastic IP addresses per Region`, ossia il numero massimo di indirizzi IP elastici in una Regione AWS.

Entrambe le quote di servizio sono regolabili e puoi richiedere un aumento dalla tua console Service Quotas. Per ulteriori informazioni, consulta [Quote di servizio 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 possono essere eseguite su un'istanza. La quota non può essere modificata 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'istanza `c5.large` può avere fino a tre interfacce di rete. L'interfaccia di rete principale per l'istanza viene calcolata come unica. Quindi, puoi collegare due interfacce di rete aggiuntive all'istanza. Poiché ogni attività che usa la modalità di rete `awsvpc` richiede un'ENI, di solito si possono eseguire solo due attività su questo tipo di istanza. Questo può portare a un sottoutilizzo della capacità del cluster. Se abiliti il `trunking` dell'interfaccia di rete elastica, puoi aumentare la densità dell'interfaccia di rete per collocare un numero maggiore di attività su ciascuna istanza. Con il `trunking` attivo, un'istanza `c5.large` può avere fino a 12 interfacce di rete. L'istanza ha un'interfaccia di rete primaria e Amazon ECS crea e collega un'interfaccia di rete “trunk” all'istanza. Di conseguenza, con questa configurazione puoi eseguire 10 attività sull'istanza invece delle due attività predefinite. Per ulteriori informazioni, consulta [Aumento delle interfacce di rete delle istanze di container Linux di Amazon ECS](#).

## Limitazione della larghezza di banda della rete dell'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. A volte questi ritiri automatici possono causare un ritardo nell'avvio delle attività, con un conseguente rallentamento della velocità di implementazione. Quando si verifica la limitazione delle API, compare 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 sulla risoluzione di tali errori di limitazione della larghezza di banda della rete, consulta [Handling throttling issues](#).

## 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, valuta i seguenti suggerimenti.

### 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 del servizio AWS Cloud Map riduce il numero massimo di istanze eseguibili a 1.000 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 nel rilevamento servizi ed eseguono controlli dell'integrità quando si avvia un'attività. Quando più servizi che utilizzano il service discovery con un gran numero di attività vengono distribuiti contemporaneamente, 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 sulla risoluzione di tali errori di limitazione della larghezza di banda della rete, consulta [Gestisci i problemi di limitazione di Amazon ECS](#).

# Documentazione di riferimento dell'API Amazon ECS

Oltre alla Console di gestione AWS e all'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 sull'AWS CLI, consulta [Riferimento AWS Command Line Interface 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             |
|-------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Aggiungi le autorizzazioni alla policy ECSServiceRolePolicy gestita da <a href="#">Amazon</a> . | La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata per includere l'autorizzazione <code>ssmmessages:OpenDataChannel</code> . Questa autorizzazione consente ad Amazon ECS di aprire canali di dati per le sessioni ECS Exec.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     | 20 gennaio 2026  |
| Aggiorna le autorizzazioni a. AmazonECSInfrastructureRolePolicyForManagedInstances              | <p>La AmazonECSInfrastructureRolePolicyForManagedInstances politica è stata aggiornata per modificare le CreateFleet autorizzazioni. Le condizioni basate sulle risorse per sottoreti, gruppi di sicurezza e EC2 immagini sono state rimosse perché:</p> <ul style="list-style-type: none"> <li>Le sottoreti e i gruppi di sicurezza non vengono creati o gestiti da Amazon ECS, quindi non contengono i tag richiesti. AmazonECSManaged</li> <li>Sebbene Amazon ECS possieda e gestisca EC2 le immagini, i tag non sono visibili ai clienti, impedendo l'applicazione di politiche basate sulle risorse.</li> </ul> <p>Questa modifica garantisce il corretto funzionamento della policy con risorse prive dei tag di gestione ECS previsti.</p> | 15 dicembre 2025 |
| Politica di durata corretta <a href="#">delle istanze gestite</a>                               | Documentazione aggiornata sulle istanze gestite per chiarire la terminologia relativa alla durata delle istanze                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   | 11 dicembre 2025 |

| Modifica                                                                                                                           | Descrizione                                                                                                                                                                                                                                                                                                              | Data             |
|------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
|                                                                                                                                    | ed estendere la durata massima delle istanze da 14 giorni a 21 giorni.                                                                                                                                                                                                                                                   |                  |
| Aggiungi le autorizzazioni alla policy ECSServiceRolePolicy gestita da <a href="#">Amazon</a> .                                    | La policy IAM gestita AmazonECSServiceRolePolicy è stata aggiornata per includere l'autorizzazione ec2:DescribeInstanceEventWindows . Questa autorizzazione consente ad Amazon ECS di recuperare Amazon EC2 Event Windows per i servizi e i cluster associati a Event Windows.                                           | 20 novembre 2025 |
| Aggiunte autorizzazioni a AmazonECSServiceRolePolicy .                                                                             | AmazonECSServiceRolePolicy È stato aggiornato per includere le autorizzazioni ec2:AssociateTrunkInterface e le ec2:DissociateTrunkInterface autorizzazioni per consentire il provisioning e il ECSto de-provisioning da parte di Amazon del Task ENI.                                                                    | 14 novembre 2025 |
| Aggiorna la politica <a href="#">the section called “AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity”</a> | <a href="#">the section called “AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity”</a> Policy aggiornata per consentire ad Amazon ECS di monitorare il ciclo di vita dei segreti creati da Amazon ECS.                                                                                            | 13 novembre 2025 |
| Distribuzioni canary di Amazon ECS                                                                                                 | Le implementazioni di Amazon ECS Canary consentono di testare nuove versioni delle applicazioni indirizzando una piccola percentuale di traffico verso la nuova versione e mantenendo la maggior parte sulla versione stabile. Per ulteriori informazioni, consulta <a href="#">Distribuzioni canary di Amazon ECS</a> . | 28 ottobre 2025  |

| Modifica                                                                                               | Descrizione                                                                                                                                                                                                                                                                                                                                                | Data              |
|--------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Implementazioni lineari di Amazon ECS                                                                  | Le implementazioni lineari di Amazon ECS consentono di spostare gradualmente il traffico dalla revisione corrente del servizio a una nuova revisione con incrementi percentuali uguali, fornendo una distribuzione controllata con monitoraggio in ogni fase. Per ulteriori informazioni, consulta <a href="#">Implementazioni lineari di Amazon ECS</a> . | 28 ottobre 2025   |
| Regioni di istanze gestite di Amazon ECS                                                               | Amazon ECS Managed Instances è ora disponibile in regioni aggiuntive. Per ulteriori informazioni, consulta <a href="#">Architect for Amazon ECS Managed Instances</a> .                                                                                                                                                                                    | 27 ottobre 2025   |
| AWS per Fluent Bit versione 3.0.0                                                                      | AWS per Fluent Bit 3.0.0 è ora disponibile la versione, basata sulla versione Fluent Bit e 023. 4.1.1 AL2 Per ulteriori informazioni, consulta <a href="#">AWS per gli archivi di Fluent Bit immagini per Amazon ECS</a> .                                                                                                                                 | 14 ottobre 2025   |
| FireLens supporto per utenti non root in container                                                     | Amazon ECS supporta l'esecuzione del FireLens contenitore come utente non root. Per ulteriori informazioni, consulta <a href="#">Inviare i log di Amazon ECS a un servizio o AWSAWS Partner</a> .                                                                                                                                                          | 13 ottobre 2025   |
| Acquisizione di eventi Amazon ECS nella console                                                        | La console Amazon ECS fornisce funzionalità di acquisizione degli eventi che archiviano gli eventi generati da Amazon ECS, come le azioni di servizio e le modifiche allo stato delle attività, in Logs through. CloudWatch EventBridge <a href="#">Per ulteriori informazioni, consulta Task lifecycle events</a> .                                       | 2 ottobre 2025    |
| Amazon ECS supporta Amazon ECS Managed Instances per l'esecuzione di carichi di lavoro containerizzati | Amazon ECS supporta Amazon ECS Managed Instances per l'esecuzione di carichi di lavoro containerizzati. Per ulteriori informazioni, consulta <a href="#">Amazon ECS Managed Instances</a> .                                                                                                                                                                | 30 settembre 2025 |

| Modifica                                                                                                    | Descrizione                                                                                                                                                                                                                                                                                                                                                              | Data              |
|-------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| Aggiunto nuovo <a href="#">the section called “AmazonECS InstanceRolePolicy ForManagedInstances”</a>        | È stata aggiunta una nuova ECSInstance RolePolicy yForManagedInstances policy di Amazon che fornisce le autorizzazioni per la registrazione delle istanze gestite di Amazon ECS con i cluster Amazon ECS.                                                                                                                                                                | 30 settembre 2025 |
| Aggiunto nuovo <a href="#">the section called “AmazonECS InfrastructureRole PolicyFor ManagedInstances”</a> | È stata aggiunta una nuova ECSInfrastructure RolePolicyForManagedInstances policy di Amazon che fornisce l'accesso ad Amazon ECS per creare e gestire risorse EC2 gestite da Amazon.                                                                                                                                                                                     | 30 settembre 2025 |
| Support per l'esecuzione di carichi di lavoro Amazon ECS in una configurazione solo IPv6                    | I carichi di lavoro Amazon ECS possono ora essere eseguiti in una configurazione IPv6 solo che consente alle attività di comunicare esclusivamente tramite IPv6. Per ulteriori informazioni, consulta le opzioni di <a href="#">task networking di Amazon ECS per il tipo di EC2 avvio e le opzioni di task networking di Amazon ECS per il tipo di lancio Fargate</a> . | 29 settembre 2025 |
| Supporto per Amazon ECS Service Connect tramite namespace AWS Cloud Map condivisi                           | Amazon ECS Service Connect ora supporta l'uso di AWS Cloud Map namespace condivisi per la comunicazione tra account. service-to-service Per ulteriori informazioni, consulta <a href="#">Amazon ECS Service Connect con namespace AWS Cloud Map condivisi</a> .                                                                                                          | 12 settembre 2025 |
| Amazon ECS migliora la modifica delle definizioni delle attività nella AWS console con Amazon Q             | Per ulteriori informazioni, consulta <a href="#">Utilizzo di Amazon Q per fornire i suggerimenti sulla definizione delle attività nella console Amazon ECS</a>                                                                                                                                                                                                           | 10 settembre 2025 |



| Modifica                                                                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Data             |
|--------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per l'esecuzione di comandi ECS Exec dalla console                                                  | Ora puoi eseguire i comandi ECS Exec direttamente dalla console Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Esecuzione di comandi ECS Exec dalla console</a> .                                                                                                                                                                                                                                                                                                        | 4 settembre 2025 |
| Aggiunto nuovo <a href="#">the section called “AmazonECSInfrastructureRolePolicyForManagedInstances”</a>     | È stata aggiunta una nuova ECSInfrastructureRolePolicyForManagedInstances policy di Amazon che fornisce l'accesso ad Amazon ECS per creare e gestire risorse EC2 gestite da Amazon.                                                                                                                                                                                                                                                                                                    | 31 agosto 2025   |
| Aggiungi le autorizzazioni alla policy gestita di <a href="#">Amazon EC2 Container Servicefor EC2 Role</a> . | La policy IAM gestita AmazonEC2ContainerServiceforEC2Role è stata aggiornata per includere l'autorizzazione <code>ecs:ListTagsForResource</code> . Questa autorizzazione consente all'agente Amazon ECS di recuperare i tag delle attività e delle istanze di container tramite l'endpoint dei metadati delle attività ( <code>\${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags</code> ). Per ulteriori informazioni, consulta <a href="#">Amazon EC2 ContainerServicefor EC2 Role</a> . | 4 agosto 2025    |
| Aggiungi autorizzazioni ad <a href="#">Amazon ECSInfrastructure RolePolicyForLoadBalancers</a> .             | La policy IAM gestita AmazonECSInfrastructureRolePolicyForLoadBalancers è stata aggiornata con nuove autorizzazioni per la descrizione, l'annullamento della registrazione e la registrazione dei gruppi di destinazione.                                                                                                                                                                                                                                                              | 25 luglio 2025   |

| Modifica                                                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                        | Data           |
|-------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Supporto per l'aggiornamento di un controller di implementazione del servizio                         | Amazon ECS supporta l'aggiornamento di un controller di implementazione del servizio, che consente di passare da un tipo di implementazione all'altro per un servizio esistente. Per ulteriori informazioni, consulta <a href="#">Aggiornamento dei parametri del servizio di Amazon ECS</a> .                                                                                                                                                     | 15 luglio 2025 |
| Implementazioni native blue/green di Amazon ECS                                                       | Amazon ECS supporta le blue/green implementazioni native, consentendoti di convalidare nuove revisioni dei servizi prima di indirizzare il traffico di produzione e verso di esse. Per ulteriori informazioni, consulta le <a href="#">blue/green implementazioni di Amazon ECS</a> .                                                                                                                                                              | 15 luglio 2025 |
| Aggiungi una nuova policy ECSInfrastructureRolePolicyForLoadBalancers gestita da Amazon.              | Questa policy fornisce l'accesso ad altre risorse di AWS servizio necessarie per gestire i sistemi di bilanciamento del carico associati ai carichi di lavoro Amazon ECS per tuo conto. Per ulteriori informazioni, consulta <a href="#">Amazon ECSInfrastructureRolePolicyForLoadBalancers</a> .                                                                                                                                                  | 15 luglio 2025 |
| Supporto del manifesto dell'indice SOCI v2                                                            | I clienti Fargate che desiderano utilizzare SOCI potranno usare solo il manifesto dell'indice SOCI v2. I clienti esistenti che hanno già utilizzato SOCI su Fargate possono continuare a usare il manifesto dell'indice SOCI v1, tuttavia consigliamo vivamente di migrare al manifesto dell'indice SOCI v2. Per ulteriori informazioni, consulta <a href="#">Caricamento lento delle immagini del container utilizzando Seekable OCI (SOCI)</a> . | 2 luglio 2025  |
| Aggiunto l'ID attività negli eventi di azione del servizio generati a causa di problemi di integrità. | Per ulteriori informazioni, consulta <a href="#">Messaggi di eventi non integri del servizio di Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                   | 30 giugno 2025 |

| Modifica                                                                                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                        | Data           |
|-----------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Aggiorna alla modalità predefinita del driver di log.                                               | La modalità di log predefinita per Amazon ECS è stata aggiornata da blocking a non-blocking. Puoi sovrascrivere questa modalità predefinita specificando una mode nella definizione del container di logConfiguration o configurarne una personalizzata modificando l'impostazione dell'account defaultLogDriverMode utilizzando la console Amazon ECS o l' AWS CLI. Per ulteriori informazioni, consulta <a href="#">Modalità predefinita dei driver di log</a> . | 25 giugno 2025 |
| Aggiungi autorizzazioni ad <a href="#">Amazon ECSServiceRolePolicy</a> .                            | La policy IAM AmazonECSServiceRolePolicy gestita è stata aggiornata con nuove AWS Cloud Map autorizzazioni che Amazon ECS può aggiornare e gli attributi del AWS Cloud Map servizio per i servizi gestiti da Amazon ECS.                                                                                                                                                                                                                                           | 24 giugno 2025 |
| Supporto per le AMI Windows Server 2025 Full e Windows Server 2025 Core ottimizzate per Amazon ECS. | Per ulteriori informazioni, consulta Windows <a href="#">ottimizzato per Amazon ECS</a> . AMIs                                                                                                                                                                                                                                                                                                                                                                     | 20 giugno 2025 |
| Supporto dell'aggiornamento della strategia di un provider di capacità.                             | Ora puoi aggiornare la strategia di un provider di capacità del servizio. Per ulteriori informazioni, consulta <a href="#">Aggiornamento dei parametri di servizio</a> .                                                                                                                                                                                                                                                                                           | 12 giugno 2025 |
| <a href="#">Aggiungi autorizzazioni ad Amazon ECSInfrastructureRolePolicyForVolumes</a>             | La policy AmazonECSInfrastructureRolePolicyForVolumes è stata aggiornata per aggiungere l'autorizzazione ec2:DescribeInstances. L'autorizzazione aiuta a prevenire la collisione dei nomi dei dispositivi per i volumi Amazon EBS collegati alle attività di Amazon ECS eseguite sulla stessa istanza di container.                                                                                                                                                | 2 giugno 2025  |

| Modifica                                                                                                               | Descrizione                                                                                                                                                                                                                                                                                                                              | Data             |
|------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Supporto per la velocità di inizializzazione del volume per i 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 collegare alle attività di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">volumeInitializationRate</a> . | 13 maggio 2025   |
| Supporto per l'interruzione dell'implementazione di un servizio.                                                       | Ora puoi interrompere l'implementazione di un servizio. Per ulteriori informazioni, consulta <a href="#">Arresto delle implementazioni del servizio di Amazon ECS</a> .                                                                                                                                                                  | 2 maggio 2025    |
| Supporto per gli endpoint dual stack di 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="#">Utilizzo degli endpoint dual stack di Amazon ECS</a> .      | 30 aprile 2025   |
| Supporto per la configurazione di una modalità di driver di log predefinita a livello di account in una Regione AWS.   | Ora puoi configurare una modalità di consegna predefinita dei messaggi di log da un container al driver di log specificato in <code>logConfiguration</code> utilizzando l'impostazione dell'account <code>defaultLogDriverMode</code> . Per ulteriori informazioni, consulta <a href="#">Modalità predefinita dei driver di log</a> .    | 17 aprile 2025   |
| Supporto per la migrazione di un servizio con un formato ARN breve al pacchetto in formato ARN lungo.                  | Ora puoi migrare un servizio con un formato ARN breve al pacchetto in formato ARN lungo. Per ulteriori informazioni, consulta <a href="#">Migrazione di un ARN di servizio breve di Amazon ECS in un ARN lungo</a> .                                                                                                                     | 13 febbraio 2025 |
| Supporto per tre tipi di iniezione di guasti su Amazon ECS e Fargate.                                                  | Ora puoi utilizzare la porta blackhole di rete, la latenza di rete e l'iniezione di guasti per la perdita di pacchetti di rete con Amazon ECS e Fargate. Per ulteriori informazioni, consulta <a href="#">Utilizzo di AWS Fault Injection Service per testare i carichi di lavoro Amazon ECS</a> .                                       | 19 dicembre 2024 |

| Modifica                                                                                                      | Descrizione                                                                                                                                                                                                                                                                                                    | Data             |
|---------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Support per CloudWatch Container Insights con osservabilità migliorata.                                       | Puoi utilizzare CloudWatch Container Insights con una migliore osservabilità. Per ulteriori informazioni, consulta <a href="#">Monitoraggio dei container di Amazon ECS utilizzando Container Insights con osservabilità migliorata</a> .                                                                      | 2 dicembre 2024  |
| Supporto per il dimensionamento predittivo.                                                                   | Puoi utilizzare il dimensionamento predittivo 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 |
| Supporto per il ribilanciamento del servizio.                                                                 | Ora puoi fare in modo che Amazon ECS bilanci automaticamente le attività del servizio tra le zone di disponibilità. Per ulteriori informazioni, consulta <a href="#">Bilanciamento di un servizio Amazon ECS tra zone di disponibilità</a> .                                                                   | 19 novembre 2024 |
| Supporto per la configurazione della risoluzione dei tag delle immagini del container nei digest di immagini. | Ora puoi configurare la risoluzione dei tag delle immagini del container nei digest per ogni container nelle attività che fanno parte di un servizio. Per ulteriori informazioni, consultare <a href="#">Risoluzione dell'immagine del container</a> e <a href="#">versionConsistency</a> .                    | 19 novembre 2024 |
| Aggiunta nuova policy IAM AmazonECSInfrastructureRolePolicyForVpcLattice.                                     | La <a href="#">AmazonECSInfrastructureRolePolicyForVpcLattice</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 di Amazon ECS.                                           | Ora puoi utilizzare VPC Lattice per standardizzare la service-to-service connettività, la sicurezza e l'osservabilità. Per ulteriori informazioni, consulta <a href="#">Utilizzo di Amazon VPC Lattice per connettere, osservare e proteggere i servizi di Amazon ECS</a> .                                                                                   | 18 novembre 2024 |
| Supporto per la cronologia dei servizi utilizzando implementazioni e revisioni dei servizi.          | Puoi utilizzare le implementazioni e le revisioni dei servizi per visualizzare la cronologia delle implementazioni. Per ulteriori informazioni, consulta <a href="#">Visualizzazione della cronologia del servizio utilizzando le implementazioni dei servizi di Amazon ECS</a> e <a href="#">Revisioni dei servizi di Amazon ECS</a> .                       | 30 ottobre 2024  |
| Supporto 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="#">Utilizzo dei volumi Amazon EBS con Amazon ECS</a> .                                                                                                                              | 24 ottobre 2024  |
| Aggiungi autorizzazioni ad Amazon ECSInfrastructureRolePolicyForVolumes.                             | La policy AmazonECSInfrastructureRolePolicyForVolumes 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 |
| Policy di riavvio per container individuali nelle attività di Amazon ECS.                            | Puoi abilitare le policy di riavvio per i container essenziali e non indicati nella definizione delle attività per superare più rapidamente i guasti temporanei e mantenere la disponibilità delle attività. Per ulteriori informazioni, consulta <a href="#">Riavviare singoli container nelle attività Amazon ECS con policy di riavvio dei container</a> . | 15 agosto 2024   |

| Modifica                                                                                                                             | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                     | Data           |
|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Autorizzazioni aggiunte a <a href="#">the section called “Amazon ECS_FullAccess”</a>                                                 | La policy AmazonECS_FullAccess è stata aggiornata per aggiungere le autorizzazioni iam:PassRole per i ruoli IAM per un ruolo denominato ecsInfrastructureRole . Questo è il ruolo IAM predefinito creato da Console di gestione AWS 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 delle immagini del container per i servizi che utilizzano il tipo di implementazione di aggiornamento in sequenza (ECS). | Per garantire che tutte le attività in un servizio che usa il controller di implementazione di aggiornamento in sequenza utilizzino la stessa immagine del container, Amazon ECS risolve i nomi delle immagini del container e tutti i tag di immagine specificati nella definizione delle attività nei digest di tali immagini. Per ulteriori informazioni, consulta <a href="#">Risoluzione dell'immagine del container</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  |

| Modifica                                                                      | Descrizione                                                                                                                                                                                                                                                                                                   | Data            |
|-------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| Supporto per gMSA per container Linux su Fargate                              | Amazon ECS supporta l'autenticazione di Active Directory per i container Linux su Fargate tramite un tipo speciale di account di servizio denominato account del servizio gestito del gruppo (gMSA). Per ulteriori informazioni, consulta <a href="#">Utilizzo di gMSA per i container Linux su Fargate</a> . | 5 marzo 2024    |
| 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 |
| TLS di Service Connect                                                        | Ora puoi utilizzare <a href="#">TLS con Service Connect</a> .                                                                                                                                                                                                                                                 | 22 gennaio 2024 |
| Policy gestita di 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 |
| Svuotamento delle istanze gestite da Amazon ECS                               | Puoi utilizzare lo <a href="#">svuotamento gestito delle istanze</a> di Amazon ECS per facilitare la chiusura ordinata delle istanze Amazon ECS.                                                                                                                                                              | 19 gennaio 2024 |
| Aggiunto il supporto per Ubuntu 22 a 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 |



| Modifica                                                              | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           | Data             |
|-----------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Aggiunta la policy IAM AmazonECSInfrastructureRolePolicyForVolumes    | 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  |
| Volume di dati Amazon EBS per attività di Amazon ECS                  | Puoi configurare un <a href="#">volume di dati Amazon EBS</a> per attività durante l'implementazione per il collegamento ad attività di Amazon ECS autonome o attività gestite da un servizio di ECS. La configurazione di un volume al momento dell'implementazione consente di creare definizioni delle attività riutilizzabili non limitate a tipi o impostazioni di volume specifici. I volumi Amazon EBS forniscono un'archiviazione 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 Amazon ECS classica ha raggiunto la fine del ciclo di vita | La console Amazon ECS ha raggiunto la fine del ciclo di vita.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 04 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.                                                                                                                                                                                                                                                                                                                                                                                                 | 04 dicembre 2023 |
| Supporto del Monitoraggio del runtime                                 | Puoi utilizzare Monitoraggio del runtime per monitorare i carichi di lavoro Amazon ECS e identificare comportamenti dannosi o non autorizzati. Per ulteriori informazioni, consulta <a href="#">Monitoraggio del runtime</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   |

| Modifica                                                                            | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Data              |
|-------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|
| AWS Fargate configura zione 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>                                                                                                                                                                                                                                                                                 | 05 settembre 2023 |
| 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="#">Caricamento lento delle immagini del container utilizzando Seekable OCI (SOCI)</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, consulta <a href="#">Usare g MSAs per Linux Containers e Using g for Windows Containers</a> . MSAs                                 | 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 il ridimensionamento di 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                                         | È possibile utilizzare gMSA per autenticarsi in contenitori Active Directory per Linux su EC2. Per ulteriori informazioni, consulta <a href="#">Using g MSAs for Linux Containers</a> .                                                                                                                                                                                                                                                                                                                           | 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 aggiornamenti in sequenza"</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 rilevamento 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'implementazione di aggiornamenti continui”</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="#">Amazon ECS endpoints and quotas</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 delle istanze preinizializzate per il tuo gruppo Auto Scaling Amazon ECS</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 istanze esterne</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 container 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 i carichi di lavoro Fargate</a> , <a href="#">Eliminare un cluster Amazon ECS</a> , <a href="#">Aggiornamento di una definizione di attività Amazon ECS attraverso la console</a> e <a href="#">Annullamento della registrazione di una revisione di definizione dell'attività con 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 di Amazon ECS attraverso la nuova 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 di 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 l'integrità delle istanze di 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 container 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="#">Log 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 su 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> .       | 08 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-premises o di una macchina virtuale (VM) con il cluster. Per ulteriori informazioni, consulta <a href="#">Cluster Amazon ECS per istanze esterne</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 container 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'implementazione di aggiornamenti continui di Amazon ECS</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> .                                                                               | 05 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="#">Versione 4 degli endpoint dei metadati delle attività di Amazon ECS</a> .                                                                                                                                              | 05 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, zone locali e zone Wavelength</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 <code>bridge</code> rete <code>awsipc</code> o ospitate su EC2 istanze Amazon che eseguono almeno una versione <code>1.43.0</code> dell'agente contenitore. Per ulteriori informazioni, consulta <a href="#">Versione 4 degli endpoint dei metadati delle attività di Amazon ECS</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 dell'utente. Per ulteriori informazioni, consulta <a href="#">Parametri di utilizzo</a> .                                                                                                                                                                                                                 | 03 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 temporanea delle attività di 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="#">Passare una singola variabile di ambiente a un container 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="#">Amazon ECS endpoints and quotas</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="#">Usare 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 temporanea delle attività di 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 <a href="#">Opzioni di networking delle attività di Amazon ECS per il tipo di avvio Fargate</a>.</li><li>• Task ENIs aggiunge il supporto per i frame jumbo. 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 container di 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="#">Versione 4 degli endpoint dei metadati delle attività di Amazon ECS.</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="#">Usare 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="#">Versione 4 degli endpoint dei metadati delle attività di Amazon ECS</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="#">Trasferimento di dati sensibili a un container 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="#">CodeDeploy implementazioni blu/verdi per Amazon ECS</a> .                                                                                                                                                                                                                             | 06 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="#">Usare 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 log 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 il dimensionamento automatico dei 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 operazioni di servizi 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 container di 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 un potenziamento delle interfacce di rete dei container Amazon ECS.</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 di destinazione con un servizio Amazon ECS.</a>                                                                                                                                                                                                                                                                                                                  | 30 luglio 2019 |

| Modifica                                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    | Data           |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| 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="#">Specifica di dati sensibili utilizzando segreti di Secrets Manager in Amazon ECS</a> .                                                                                                                                                                                                                                                                                                                                                            | 20 luglio 2019 |
| CloudWatch Container Insights                                                | Amazon ECS ha aggiunto il supporto per CloudWatch Container Insights. Per ulteriori informazioni, consulta <a href="#">Monitora i container di Amazon ECS utilizzando Container Insights con osservabilità migliorata</a> .                                                                                                                                                                                                                                                                                                                                                                    | 09 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>awsvpcTrunking</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 delle istanze di container Linux di Amazon ECS</a> . | 06 giugno 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 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 |
| 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="#">Trasferimento di dati sensibili a un container 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 Gestione dei segreti AWS segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta <a href="#">Trasferimento di dati sensibili a un container 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 |

| Modifica                                                                     | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                | Data            |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|
| 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="#">Implementare i servizi Amazon ECS utilizzando un controller di terze parti</a> .                                                                                                                        | 27 marzo 2019   |
| 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 <code>PutAccountSettingDefault</code> che consente a un utente di impostare lo stato di attivazione del ARN/ID formato 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 |

| Modifica                                      | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                     | Data             |
|-----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|
| Amazon ECS supporta carichi di lavoro per GPU | <p>Amazon ECS introduce il supporto per i carichi di lavoro della GPU consentendo di creare cluster con istanze di container abilitate per GPU. Nella definizione di un'attività, puoi specificare il numero di richieste GPUs e l'agente ECS aggiungerà l'operazione fisica GPUs al contenitore.</p> <p>Per ulteriori informazioni, consulta <a href="#">Definizioni di attività Amazon ECS per carichi di lavoro GPU</a>.</p> | 4 febbraio 2019  |
| Amazon ECS ha esteso il supporto dei segreti  | <p>Amazon ECS ha esteso il supporto per l'utilizzo di Gestione dei segreti AWS segreti direttamente nelle definizioni delle attività per iniettare dati sensibili nei contenitori.</p> <p>Per ulteriori informazioni, consulta <a href="#">Trasferimento di dati sensibili a un container 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 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="#">Trasferimento di dati sensibili a un container 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 Amazon ECS blue/green                            | <p>Amazon ECS ha aggiunto il supporto per le blue/green distribuzioni 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="#">CodeDeploy implementazioni blu/verdi per Amazon ECS</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="#">Trasferimento di dati sensibili a un container 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="#">Aggiunta di tag alle 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="#">Progettazione per 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="#">Progettazione per 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 ECS_CONTAINER_META_DATA_URI 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 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> . | 18 ottobre 2018   |
| Espansione della regione per il rilevamento servizi di Amazon ECS              | Il rilevamento 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="#">Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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 Gestione dei segreti AWS. 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 il rilevamento servizi di Amazon ECS                                        | <p>Il servizio di rilevamento 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="#">Usare il rilevamento servizi per connettere i servizi Amazon ECS con nomi DNS</a>.</p>                                                                                                                 | 30 agosto 2018    |
| Processi pianificati con supporto per i processi Fargate                                                 | Amazon ECS ha introdotto il supporto per i processi pianificati per il tipo di avvio Fargate.                                                                                                                                                                                                                                                                                                                                                                             | 28 agosto 2018    |
| Autenticazione del registro privato tramite supporto Gestione dei segreti AWS                            | <p>Amazon ECS ha introdotto il supporto per l'autenticazione di registri privati tramite Gestione dei segreti AWS. 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 archiviazione 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="#">Progettazione per 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, consultare <a href="#">Strategia di pianificazione delle repliche</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, consultare <a href="#">Strategia di pianificazione dei daemon</a>.</li></ul> | 12 giugno 2018 |

 **Note**

I processi Fargate non supportano la strategia di pianificazione DAEMON

| Modifica                                                                                                                        | Descrizione                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      | Data           |
|---------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| Agente del container<br>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 il rilevamento servizi | <p>Aggiunto il supporto per la configurazione del rilevamento 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="#">Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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 dei metadati dell'AMI Linux ottimizzata 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="#">Progettazione per 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 dei metadati dell'AMI Linux ottimizzata 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 dell'integrità</a>.</li> <li>• Aggiunto supporto per <a href="#">Usare il rilevamento servizi per connettere i servizi Amazon ECS con nomi DNS</a></li> </ul> <p>Per ulteriori informazioni, consulta <a href="#">Versioni della piattaforma Fargate per Amazon ECS</a>.</p> | 26 marzo 2018  |
| Rilevamento servizi di Amazon ECS                         | <p>Aggiunta dell'integrazione con Route 53 per supportar e il rilevamento servizi di Amazon ECS. Per ulteriori informazioni, consulta <a href="#">Usare il rilevamento servizi per connettere i servizi Amazon ECS con 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 dell'integrità</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="#">Progettazione per 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 dell'evento di servizio documentato che si attiva quando un'attività del 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="#">Progetta la tua soluzione per 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 EC2</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.                                                                                                                                                                                                                                                                                              | 07<br>settembre<br>2017 |
| RunTask sovrascrive                                                           | 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.                                                                                                                                                                                                             | 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="#">Drenare le istanze di container di 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="#">Usare le strategie per definire il posizionamento delle attività su 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="#">Pianificare i container 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="#">Automatizzare 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 bilanciatore del carico 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 bilanciatore del carico. 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="#">Scalabilità automatica del 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. Console di gestione AWS 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 bilanciatore del carico 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.