aws

Guida per gli sviluppatori

Amazon Elastic Container Service

Amazon Elastic Container Service: Guida per gli sviluppatori

Table of Contents

Cos'è Amazon Elastic Container Service?

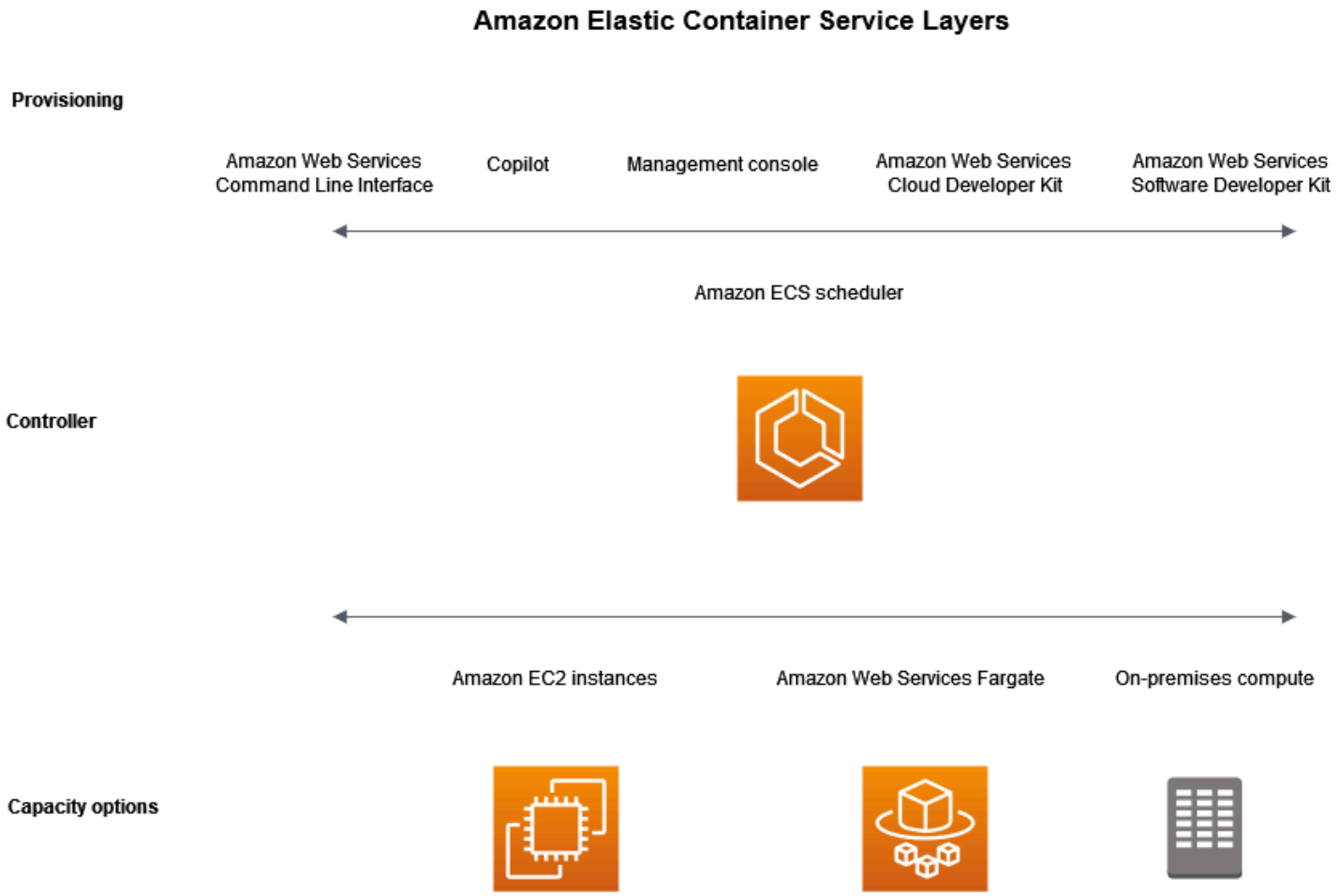
Amazon Elastic Container Service (Amazon ECS) è un servizio di orchestrazione di container completamente gestito che facilita l'implementazione, la gestione e il dimensionamento delle applicazioni containerizzate. Essendo un servizio completamente gestito, Amazon ECS include AWS configurazioni e best practice operative integrate. È integrato con strumenti AWS sia di terze parti, come Amazon Elastic Container Registry e 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 di Amazon ECS

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.



Capacità di Amazon ECS

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

- Istanze Amazon EC2 nel cloud AWS

  Scegli il tipo di istanza e il numero di istanze per gestire la capacità.
- Serverless (AWS Fargate (Fargate)) nel cloud AWS

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

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

La capacità può essere localizzata in una delle seguenti risorse: AWS

- Zone di disponibilità
- Zone locali
- Zone Wavelength
- Regioni AWS
- AWS Outposts

Controller di Amazon ECS

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

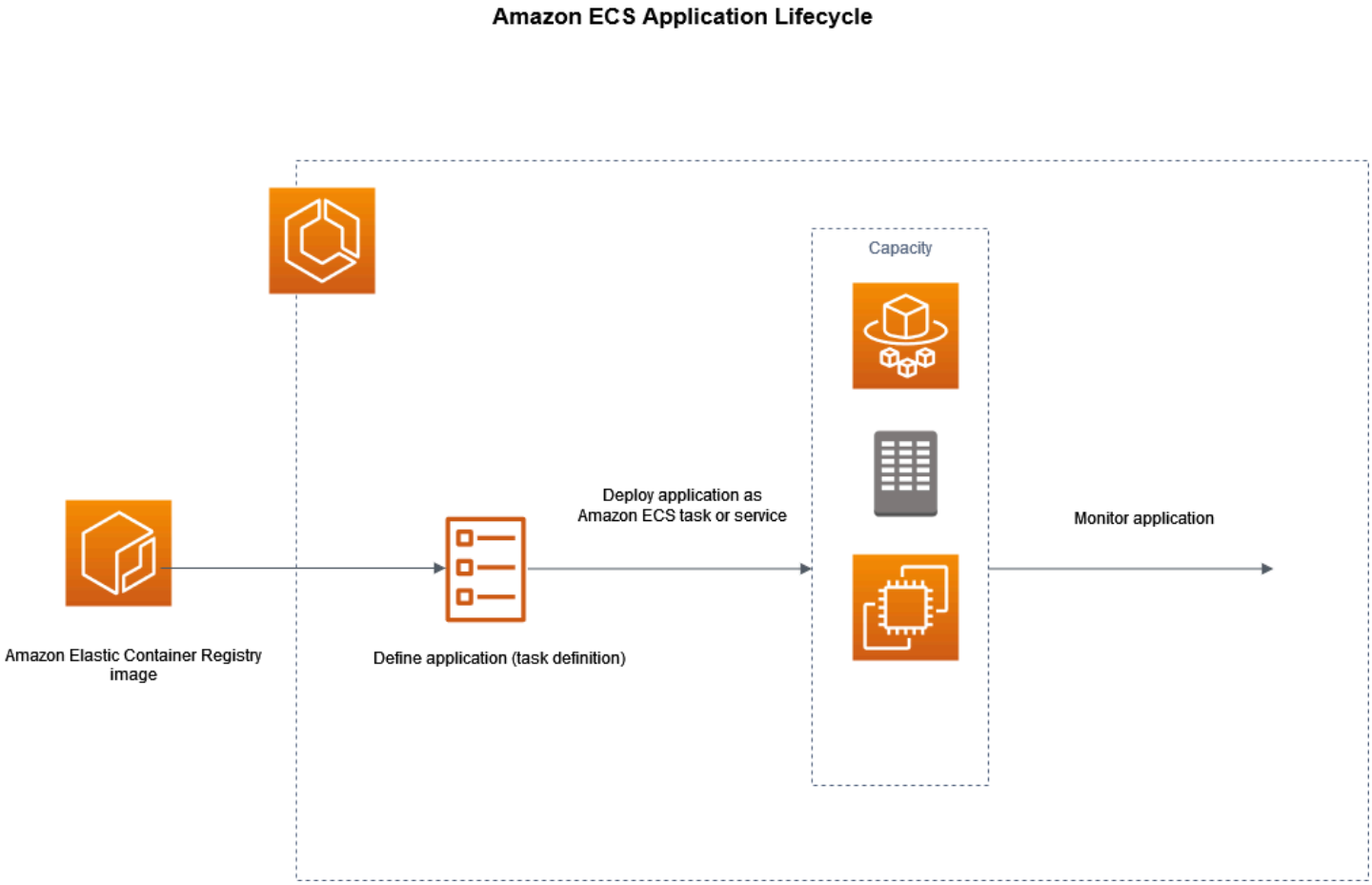
Provisioning di Amazon ECS

Esistono diverse opzioni per il provisioning di Amazon ECS:

- AWS Management Console: fornisce un'interfaccia Web che può essere utilizzata per accedere alle risorse Amazon ECS.
- AWS Command Line Interface (AWS CLI) — Fornisce comandi per un'ampia gamma di AWS servizi, tra cui Amazon ECS. È supportata su Windows, Mac e Linux. Per ulteriori informazioni, consulta AWS Command Line Interface.
- AWS SDK: fornisce API specifiche per la lingua 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 SDK di AWS.
- Copilot: fornisce uno strumento open source che consente agli sviluppatori di creare, rilasciare e gestire applicazioni containerizzate pronte per la produzione su Amazon ECS. Per ulteriori informazioni, consulta Copilot sul sito Web. GitHub
- 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.

Ciclo di vita delle applicazioni

Il diagramma seguente mostra il ciclo di vita dell'applicazione e il funzionamento con i componenti di Amazon ECS.



È necessario progettare le applicazioni in modo che possano essere eseguite su contenitori. Un container è un'unità di sviluppo software standardizzata che contiene tutto ciò che è necessario per l'esecuzione dell'applicazione software. Ciò include codice, runtime, strumenti di sistema e librerie di sistema pertinenti. I container vengono creati da un modello di sola lettura denominato immagine. Le immagini sono generalmente create da un Dockerfile. Un Dockerfile è un file di testo semplice che contiene le istruzioni per la creazione di un contenitore. In seguito alla creazione, tali immagini vengono archiviate in un registro, ad esempio Amazon ECR, da cui possono essere scaricate.

Dopo aver creato e archiviato l'immagine, puoi creare una definizione dell'attività di Amazon ECS. Una definizione di attività è uno schema dell'applicazione. Si tratta di un file di testo in formato JSON che descrive i parametri e uno o più container che formano l'applicazione. Ad esempio, puoi utilizzarla per specificare l'immagine e i parametri del sistema operativo, i container da utilizzare, le porte da aprire per l'applicazione e i volumi di dati da utilizzare con i container nell'attività. I parametri specifici disponibili per la definizione di attività dipendono dalle esigenze dell'applicazione specifica.

Dopo aver stabilito la definizione di attività, implementala come servizio o attività nel cluster. Un cluster è un raggruppamento logico di attività o servizi in esecuzione sull'infrastruttura di capacità registrata in un cluster.

Si definisce attività la creazione dell'istanza relativa a una definizione di attività all'interno di un cluster. Puoi eseguire un processo autonomo oppure eseguire un processo come parte di un servizio. Puoi utilizzare un servizio Amazon ECS per eseguire e mantenere simultaneamente il numero desiderato di processi in un cluster Amazon ECS. Il funzionamento è che, se uno dei processi non riesce o si interrompe per qualsiasi motivo, il pianificatore del servizio Amazon ECS lancia un'altra istanza in base alla definizione di attività. Lo fa per sostituirlo e quindi mantenere il numero desiderato di processi nel servizio.

L'agente del container funziona su ogni istanza di container in un cluster Amazon ECS. L'agente invia ad Amazon ECS informazioni sui processi correntemente in esecuzione e sull'utilizzo delle risorse dei tuoi container. Avvia e interrompe i processi ogni volta che riceve una richiesta da Amazon ECS.

Dopo aver implementato l'attività o il servizio, puoi utilizzare uno degli strumenti seguenti per monitorare l'implementazione e l'applicazione:

- CloudWatch
- Monitoraggio del runtime

Informazioni relative ad Amazon ECS

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

- AWS Fargate: panoramica delle funzionalità di Fargate.
- Windows on AWS: panoramica di Windows su AWS carichi di lavoro e servizi.
- Linux from AWS - Portafoglio di moderni sistemi operativi basati su Linux di. AWS

Tutorial per sviluppatori

- AWS Compute Blog: informazioni sulle nuove funzionalità e approfondimenti, esempi di codice e best practice.

AWS re:Post

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

Prezzi

- Prezzi di Amazon ECS: informazioni sui prezzi di Amazon ECS.
- AWS Fargate prezzi — Informazioni sui prezzi di Fargate.

Risorse generali AWS

Le seguenti risorse generali possono aiutarti a lavorare con AWS.

- Corsi e workshop: collegamenti a corsi specializzati e basati su ruoli, oltre a laboratori di autoapprendimento per aiutarti ad affinare le tue abilità e acquisire esperienza pratica. AWS
- AWS Developer Center: esplora i tutorial, scarica strumenti e scopri gli eventi per sviluppatori. AWS
- AWS Strumenti per sviluppatori: collegamenti a strumenti di sviluppo, SDK, toolkit IDE e strumenti a riga di comando per lo sviluppo e la gestione di applicazioni. AWS
- Centro risorse introduttivo: scopri come configurare Account AWS, entrare a far parte della AWS community e lanciare la tua prima applicazione.
- Tutorial pratici: segui i tutorial per avviare la step-by-step tua prima applicazione su. AWS
- AWS Whitepaper: collegamenti a un elenco completo di AWS white paper tecnici, su argomenti quali architettura, sicurezza ed economia e redatti da Solutions Architects o altri esperti tecnici. AWS
- AWS Support Center: l'hub per la creazione e la gestione dei casi. AWS Support Include anche collegamenti ad altre risorse utili, come forum, domande frequenti tecniche, stato di salute del servizio e AWS Trusted Advisor.
- AWS Support— La pagina web principale per informazioni su AWS Support one-on-one, un canale di supporto a risposta rapida per aiutarti a creare ed eseguire applicazioni nel cloud.
- Contatti - Un punto di contatto centrale per richieste relative a fatturazione, account, eventi, uso illecito e altre questioni relative ad AWS .
- AWS Termini del sito: informazioni dettagliate sul nostro copyright e marchio, sull'account, sulla licenza e sull'accesso al sito e altri argomenti.

Scopri come creare e utilizzare le risorse Amazon ECS

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

Indice

Configurazione per l'uso di Amazon ECS

Se hai già eseguito la registrazione ad Amazon Web Services (AWS) e hai già usato Amazon Elastic Compute Cloud (Amazon EC2), sei praticamente in grado di usare 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.

Iscriviti per 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 pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.
2. Segui le istruzioni online.

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

Quando ti iscrivi a un Account AWS, Utente root dell'account AWSviene creato un. L'utente root dispone dell'accesso a tutte le risorse e tutti i Servizi AWS nell'account. Come procedura consigliata in materia di sicurezza, assegnate l'accesso amministrativo a un utente e utilizzate solo l'utente root per eseguire attività che richiedono l'accesso da parte dell'utente root.

AWS ti invia un'e-mail di conferma dopo il completamento della procedura di registrazione. È possibile visualizzare l'attività corrente dell'account e gestire l'account in qualsiasi momento accedendo all'indirizzo https://aws.amazon.com/ e selezionando Il mio account.

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

1. Accedi AWS Management Consolecome proprietario dell'account scegliendo Utente root e inserendo il tuo indirizzo Account AWS email. Nella pagina successiva, inserisci la password.

   Per informazioni sull'accesso utilizzando un utente root, consulta la pagina Signing in as the root user della Guida per l'utente di Accedi ad AWS .
2. Abilita l'autenticazione a più fattori (MFA) per l'utente root.

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

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

   Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .
2. In IAM Identity Center, concedi 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 le impostazioni predefinite IAM Identity Center directory nella Guida per l'AWS IAM Identity Center utente.

Accedi come utente con accesso amministrativo

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

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

Assegna l'accesso ad altri utenti

1. In IAM Identity Center, crea un set di autorizzazioni che segua la migliore pratica di applicazione delle autorizzazioni con privilegi minimi.

   Per istruzioni, consulta Creare un set di autorizzazioni nella Guida per l'utente.AWS IAM Identity Center
2. Assegna gli utenti a un gruppo, quindi assegna l'accesso Single Sign-On al gruppo.

   Per istruzioni, consulta Aggiungere gruppi nella Guida per l'utente.AWS IAM Identity Center

Crea un cloud privato virtuale

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

Se è presente un VPC di default, puoi ignorare questa sezione e passare all'operazione successiva, Creazione di un gruppo di sicurezza. Per determinare se disponi di un VPC predefinito, consulta Supported Platforms in the Amazon EC2 Console nella Amazon EC2 User Guide. In caso contrario, puoi creare un VPC non predefinito nell'account utilizzando le fasi seguenti.

Per informazioni su come creare un VPC, consulta la sezione Creare solo un 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
Nome	Se lo desideri, puoi fornire un nome per il VPC.

Opzione	Valore
IPv4 CIDR block (Blocco CIDR IPv4)	Input manuale CIDR IPv4 La dimensione del blocco CIDR deve essere compresa tra /16 e /28.
IPv6 CIDR block (Blocco CIDR IPv6)	Nessun blocco CIDR IPv6
Tenancy	Predefinita

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

Creazione di un gruppo di sicurezza

I gruppi di sicurezza fungono da firewall per le istanze di container, controllando sia il traffico in entrata che quello in uscita a livello di istanza di container. È possibile aggiungere regole a un gruppo di sicurezza che consentono di connettersi all'istanza di container dall'indirizzo IP tramite SSH. È inoltre possibile aggiungere regole che consentono il traffico HTTP e HTTPS in entrata e in uscita da qualsiasi posizione. Aggiungi le regole per aprire le porte necessarie per le attività. Le istanze di container richiedono un accesso di rete esterno per comunicare con l'endpoint del servizio Amazon ECS.

Se prevedi di avviare istanze di container in più regioni, devi creare un gruppo di sicurezza in ogni regione. Per ulteriori informazioni, consulta Regioni e zone di disponibilità nella Guida per l'utente di Amazon EC2.

> **Tip**
>
> È necessario l'indirizzo IP pubblico del computer locale, che puoi ottenere usando un servizio. Forniamo ad esempio il servizio seguente: 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 Create a security group nella Amazon EC2 User Guide e utilizza la tabella seguente per determinare quali opzioni selezionare.

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

> **Note**
>
> Se il tuo account supporta Amazon EC2 Classic, seleziona il VPC creato in precedenza.

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 Guida per l'utente di Amazon EC2.

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 Add rules to your security group nella Amazon EC2 User Guide.

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

Opzione	Valore
	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 consiglia bile 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.

Creazione delle credenziali per connettersi all'istanza EC2

Per Amazon ECS, una coppia di chiavi è necessaria solo se si intende utilizzare il tipo di avvio EC2.

AWS utilizza la crittografia a chiave pubblica per proteggere le informazioni di accesso per la tua 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 ancora creato una coppia di chiavi, puoi crearne una utilizzando la console Amazon EC2. 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 Guida per l'utente di Amazon EC2.

Per creare una coppia di chiavi

- Usa la console Amazon EC2 per creare una coppia di chiavi. 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 istance nella Amazon EC2 User Guide.

Installa il AWS CLI

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

Per utilizzarlo AWS CLI con Amazon ECS, installa la AWS CLI versione più recente. Per informazioni sull'installazione AWS CLI o sull'aggiornamento alla versione più recente, consulta Installazione dell'interfaccia a riga di AWS comando nella Guida per l'AWS Command Line Interface utente.

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

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

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

funzionamento. Per ulteriori informazioni su Docker, consulta Cos'è Docker? e la panoramica su 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 la sezione Configurazione di 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 2, consulta la sezione Installazione di Docker su AL2023. Per tutti gli altri sistemi operativi, consulta la documentazione su Docker all'indirizzo Panoramica di Docker Desktop.
- Hai AWS CLI installato e configurato. Per ulteriori informazioni, consulta Installazione dell' AWS Command Line Interface nella Guida per l'utente dell'AWS Command Line Interface .

Se non hai bisogno di un ambiente di sviluppo locale e preferisci usare un'istanza Amazon EC2 per usare Docker, di seguito elenchiamo i passaggi per avviare un'istanza Amazon EC2 e installare Docker Engine e la CLI di Docker utilizzando Amazon Linux 2.

Installazione di Docker su AL2023

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 già utilizzi Amazon EC2, puoi avviare un'istanza Amazon Linux 2023 e installare Docker per iniziare.

Se hai già installato Docker, passa a Creazione di un'immagine Docker.

Installazione di Docker su un'istanza Amazon EC2 con un'AMI Amazon Linux 2023

1. Avvia un'istanza con l'ultima AMI Amazon Linux 2023. Per ulteriori informazioni, consulta Launching an Instance nella Amazon EC2 User Guide.
2. Connettiti alla tua istanza. Per ulteriori informazioni, consulta Connect to Your Linux Instance nella Amazon EC2 User Guide.

3. Aggiorna i pacchetti installati e la cache dei pacchetti sulla tua istanza.

```
sudo yum update -y
```

4. Installa il pacchetto Docker Community Edition più recente.

```
sudo yum install docker
```

5. Avvia il servizio Docker.

```
sudo service docker start
```

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

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

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

```
docker info
```

> **Note**
>
> In alcuni casi, l'assegnazione delle autorizzazioni necessarie a `ec2-user` per accedere al daemon Docker può richiedere il riavvio dell'istanza. Prova a riavviare l'istanza se visualizzi questo errore:
>
> ```
> Cannot connect to the Docker daemon. Is the docker daemon running on this
>  host?
> ```

Creazione di un'immagine Docker

Le definizioni di processo di Amazon ECS utilizzano le immagini Docker per avviare i container sulle istanze di container nei tuoi cluster. In questa sezione viene descritto come creare un'immagine

Docker di una semplice applicazione Web e come testarla nel sistema locale o nell'istanza Amazon EC2, per poi inviarla a un registro del container Amazon ECR per poterla utilizzare in una definizione di attività di 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

   # Install dependencies
   RUN yum update -y && \
    yum install -y httpd

   # Install apache and write hello world message
   RUN echo 'Hello World!' > /var/www/html/index.html

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

   EXPOSE 80

   CMD /root/run_apache.sh
   ```

   Questo Dockerfile utilizza l'immagine pubblica di Amazon Linux 2 ospitata su Amazon ECR Public. Le istruzioni `RUN` aggiornano le cache dei pacchetti, installano alcuni pacchetti software per il server Web e infine scrivono il contenuto "Hello World!" nella root del documento del server Web. L'`EXPOSE`istruzione indica che la porta 80 sul contenitore è quella in ascolto e l'`CMD`istruzione 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.

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

4. Elenca l'immagine del container.

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

Output:

```
REPOSITORY          TAG                 IMAGE ID            CREATED
 SIZE
hello-world         latest              e9ffedc8c286        4 minutes ago
 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. Per ulteriori informazioni sul comando docker run, consulta la documentazione di riferimento di Docker run.

```
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'istanza EC2, si tratta del valore Public DNS (DNS pubblico) del server, che è lo stesso indirizzo utilizzato per la connessione all'istanza con SSH. Assicurati che il gruppo di sicurezza per l'istanza consenta il traffico in entrata sulla porta 80.
   - Se Docker è in esecuzione in locale, accedi con il browser a http://localhost/.

- Se lo utilizzi docker-machine su un computer Windows o Mac, trova l'indirizzo IP della macchina VirtualBox virtuale che ospita Docker con il docker-machine ip comando, sostituendo *machine-name con il nome* della macchina docker che stai utilizzando.

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

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

7. Interrompi il container Docker digitando Ctrl+c.

Invia l'immagine al registro del container di Amazon Elastic

Amazon ECR è un servizio di registro AWS Docker gestito. Puoi utilizzare la CLI di Docker per inviare, estrarre e gestire le immagini nei tuoi repository Amazon ECR. Per informazioni dettagliate su Amazon ECR, casi di studio di clienti in evidenza e domande frequenti, consulta le pagine dei dettagli di 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.

   Sostituiscilo`region`, 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 Installazione dell' AWS Command Line Interface 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 Passaggi successivi. Una volta completato l'esperimento con l'immagine Amazon ECR, puoi eliminare il repository in modo che non ti siano addebitati costi per l'archiviazione dell'immagine.

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

Passaggi successivi

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

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

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

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

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

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

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

Prerequisiti

Prima di iniziare, completa i passaggi indicati Configurazione per l'uso di Amazon ECS e assicurati che il tuo AWS utente 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 Add rules to your security group 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 all'indirizzo 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 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. Seleziona Crea.

Fase 2: Creazione di una definizione di attività

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

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

```
{
    "family": "sample-fargate",
    "networkMode": "awsvpc",
    "containerDefinitions": [
        {
            "name": "fargate-app",
            "image": "public.ecr.aws/docker/library/httpd:latest",
            "portMappings": [
                {
                    "containerPort": 80,
                    "hostPort": 80,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": [
```

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

4. Scegli Crea.

Fase 3: Creazione del servizio

Crea un servizio utilizzando la definizione di attività.

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

5. Scegli Crea.

Fase 4: visualizzazione del servizio

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

Fase 5: rimozione

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

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

1. Nel pannello di navigazione scegliere Clusters (Cluster).
2. Nella pagina Cluster, seleziona il cluster creato per questo tutorial.
3. Scegli la scheda Servizi.
4. Seleziona il servizio, quindi scegli Elimina.
5. Alla richiesta di conferma, immetti delete (elimina), quindi scegli Delete (Elimina). In alternativa, puoi utilizzare l'`Force delete`opzione che consente ad Amazon ECS di ridimensionare il servizio per tuo conto prima di eliminarlo.

   Attendi che il servizio venga eliminato.
6. Scegli Elimina Cluster. Alla richiesta di conferma, immetti delete ***`cluster-name`*** Elimina nome-cluster), quindi scegli Delete (Elimina). L'eliminazione del cluster rimuove le risorse associate

create con il cluster, inclusi i gruppi con dimensionamento automatico, i VPC o i sistemi di bilanciamento del carico.

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

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

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

Prerequisiti

Prima di iniziare, completa i passaggi indicati Configurazione per l'uso di Amazon ECS e assicurati che il tuo AWS utente 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 Add rules to your security group nella Amazon EC2 User Guide.

- Type (Tipo): HTTP

- Protocol (Protocollo): TCP
- Intervallo porte: 80
- Origine: Qualsiasi (`0.0.0.0/0`)

Fase 1: creazione di un cluster

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

Per creare un cluster con AWS Management Console

1. Apri la console all'indirizzo 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), 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. Seleziona Crea.

Fase 2: Registrazione di una definizione di attività di Windows

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

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

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

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

```
    "requiresCompatibilities": ["FARGATE"]
}
```

4. Verifica le informazioni e scegli Crea.

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

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

Per creare un servizio dalla definizione di attività con la console

1. Nel riquadro di navigazione scegli Clusters (Cluster), quindi seleziona il cluster creato in Fase 1: creazione di un cluster.
2. Nella scheda Services (Servizi), scegli Create (Crea).
3. In Deployment configuration (Configurazione dell'implementazione), specifica come viene implementata l'applicazione.
   a. In Task Definition (Definizione di attività), scegli la definizione di attività creata in Fase 2: Registrazione di una definizione di attività di Windows.
   b. In Service name (Nome servizio), specifica un nome per il servizio.
   c. Per Desired tasks (Attività desiderate), immetti 1.
4. In Reti, puoi creare un gruppo di sicurezza o sceglierne uno esistente. Assicurati che il gruppo di sicurezza utilizzato contenga la regola in entrata elencata in Prerequisiti.
5. Scegli 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. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Scegli il cluster in cui è stato eseguito il servizio.
4. Nella scheda Servizi, in Nome del servizio scegli il servizio creato in Fase 3: creazione 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 le istanze Amazon EC2, i sistemi di bilanciamento del carico Elastic Load Balancing e i gruppi Auto Scaling, devono essere pulite manualmente nella console Amazon EC2 o eliminando lo stack che le ha create. AWS CloudFormation

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

   Attendi che il servizio venga eliminato.
6. Scegli Elimina Cluster. Alla richiesta di conferma, immetti delete ***cluster-name*** Elimina nome-cluster), quindi scegli Delete (Elimina). L'eliminazione del cluster rimuove le risorse associate create con il cluster, inclusi i gruppi con dimensionamento automatico, i VPC o i sistemi di bilanciamento del carico.

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

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

Per iniziare a utilizzare Amazon ECS con il tipo di avvio EC2, completa le fasi descritte di seguito.

Prerequisiti

Prima di iniziare, completa i passaggi indicati Configurazione per l'uso di Amazon ECS e assicurati che il tuo AWS utente 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 Add rules to your security group 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.

Le seguenti fasi ti guideranno attraverso la creazione di un cluster con un'istanza Amazon EC2 registrata che permetterà di eseguire un processo. 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 all'indirizzo 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 istanze Amazon EC2 al cluster, espandi Infrastruttura e seleziona Istanze Amazon EC2. 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.
   - In EC2 instance type (Tipo di istanza EC2), scegli il tipo di istanza per i tuoi carichi di lavoro. Per ulteriori informazioni sui diversi tipi di istanza, consulta Istanze Amazon EC2.

     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. Le istanze Amazon EC2 comportano costi finché sono presenti nelle tue risorse. AWS Per ulteriori informazioni, consulta Prezzi di Amazon EC2.

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. Seleziona Crea.

Fase 2: Registrazione di una definizione di attività

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

1. Nel riquadro di navigazione, scegli Definizioni di attività.
2. Scegli Create new task definition (Crea nuova definizione di attività), Create new task definition with JSON (Crea nuova definizione di attività con JSON).

3. Copia e incolla la seguente definizione di attività di esempio nella casella, quindi scegli Salva.

```
{
    "containerDefinitions": [
        {
            "command": ["New-Item -Path C:\\inetpub\\wwwroot\\index.html
 -Type file -Value '<html> <head> <title>Amazon ECS Sample App</title>
 <style>body {margin-top: 40px; background-color: #333;} </style> </head><body>
 <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in
 Amazon ECS.</p>'; C:\\ServiceMonitor.exe w3svc"],
            "entryPoint": [
                "powershell",
                "-Command"
            ],
            "essential": true,
            "cpu": 2048,
            "memory": 4096,
            "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
            "name": "sample_windows_app",
            "portMappings": [
                {
                    "hostPort": 443,
                    "containerPort": 80,
                    "protocol": "tcp"
                }
            ]
        }
    ],
    "memory": "4096",
    "cpu": "2048",
    "family": "windows-simple-iis-2019-core",
    "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
    "runtimePlatform": {"operatingSystemFamily": "WINDOWS_SERVER_2019_CORE"},
    "requiresCompatibilities": ["EC2"]
}
```

4. Verifica le informazioni e scegli Crea.

Fase 3: creazione di un servizio

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

Come creare un servizio

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

Fase 4: visualizzazione del servizio

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

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Scegli il cluster in cui è stato eseguito il servizio.
4. Nella scheda Servizi, in Nome del servizio scegli il servizio creato in Fase 3: creazione di un servizio.

5. Scegli la scheda Attività, quindi scegli l'attività nel tuo servizio.
6. Nella pagina dell'attività, nella sezione Configurazione, per IP pubblico, scegli Apri indirizzo. La schermata riportata di seguito rappresenta 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 le istanze Amazon EC2, i sistemi di bilanciamento del carico Elastic Load Balancing e i gruppi Auto Scaling, devono essere pulite manualmente nella console Amazon EC2 o eliminando lo stack che le ha create. AWS CloudFormation

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

   Attendi che il servizio venga eliminato.
6. Scegli Elimina Cluster. Alla richiesta di conferma, immetti delete ***cluster-name*** Elimina nome-cluster), quindi scegli Delete (Elimina). L'eliminazione del cluster rimuove le risorse associate create con il cluster, inclusi i gruppi con dimensionamento automatico, i VPC o i sistemi di bilanciamento del carico.

Panoramica degli strumenti per sviluppatori di Amazon ECS

Che tu faccia parte di una grande azienda o di una startup, Amazon ECS offre una varietà di strumenti che possono aiutarti a far funzionare rapidamente i tuoi container, indipendentemente dal tuo livello di competenza. Puoi utilizzare Amazon ECS nei modi seguenti:

- Scopri, sviluppa, gestisci e visualizza le applicazioni e i servizi per container utilizzando la AWS Management Console.
- Esegui operazioni specifiche per le risorse Amazon ECS con implementazioni automatizzate tramite programmazione o script utilizzando la AWS Command Line Interface, AWS SDK o l'API ECS.
- Definisci e gestisci tutte le AWS risorse del tuo ambiente con la distribuzione automatizzata utilizzandoAWS CloudFormation.
- Utilizza il flusso di lavoro completo AWS CLI Copilot end-to-end degli sviluppatori per creare, rilasciare e gestire applicazioni container conformi alle AWS migliori pratiche per l'infrastruttura.
- Usando il linguaggio di programmazione preferito, definisci l'infrastruttura o l'architettura come codice con l'opzione AWS CDK.
- Containerizza le applicazioni ospitate su istanze on-premise o Amazon EC2 o entrambe utilizzando l'ecosistema integrato di portabilità e strumenti per container AWS App2Container.
- Implementa un'applicazione su Amazon ECS o esegui il test dei container locali con container in esecuzione in Amazon ECS, tramite il formato file Docker Compose con la CLI di Amazon ECS.
- Avvia i container da Integrazione di Docker Desktop con Amazon ECS utilizzando Amazon ECS in Docker Desktop.

AWS Management Console

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

Per un tutorial esplicativo della console, consulta Scopri come creare e utilizzare le risorse Amazon ECS.

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

Inizia con. AWS Management Console

AWS Command Line Interface

AWS Command Line Interface (AWS CLI) è uno strumento unificato che puoi utilizzare per gestire i tuoi AWS servizi. 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 fornisce due set di strumenti da riga di comando: the AWS Command Line Interface(AWS CLI) e the. AWS Tools for Windows PowerShell Per ulteriori informazioni, consulta la Guida per l'utente di AWS Command Line Interface e la Guida per l'utente di AWS Tools for Windows PowerShell.

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 acquisire familiarità con le API di Amazon ECS. 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 le API di Amazon ECS e i comandi CLI corrispondenti e desideri scrivere script automatici ed eseguire azioni specifiche sulle risorse Amazon ECS.

AWS CloudFormation

AWS CloudFormation e Terraform per Amazon ECS forniscono metodi efficaci per definire la tua infrastruttura come codice (Infrastructure as Code). Puoi facilmente monitorare la versione del tuo modello o stack AWS CloudFormation che è in esecuzione in qualsiasi momento ed eseguire il ripristino dello stato precedente a una versione precedente, se necessario. È possibile eseguire implementazioni di infrastrutture e applicazioni nello stesso modo automatizzato. Questa flessibilità e automazione sono ciò che rende AWS CloudFormation Terraform due formati popolari per la distribuzione di carichi di lavoro su Amazon ECS da pipeline di distribuzione continua.

Per ulteriori informazioni su, consulta. AWS CloudFormationCreazione di risorse Amazon ECS utilizzando AWS CloudFormation

Usa AWS CloudFormation or Terraform se desideri automatizzare le implementazioni e le applicazioni dell'infrastruttura su Amazon ECS e definire e gestire in modo esplicito tutte le risorse del tuo ambiente. AWS

AWS CLI Copilot

La AWS CLI (interfaccia a riga di comando) di Copilot è uno strumento completo che consente ai clienti di distribuire e gestire applicazioni confezionate in contenitori e ambienti su Amazon ECS direttamente dal loro codice sorgente. Quando usi AWS Copilot, puoi eseguire queste operazioni senza conoscere gli elementi di AWS Amazon ECS come Application Load Balancer, sottoreti pubbliche, attività, servizi e cluster. AWS Copilot crea AWS risorse per tuo conto partendo da modelli di servizio ponderati, come un servizio web con bilanciamento del carico o un servizio di backend, fornendo un ambiente di produzione immediato per applicazioni containerizzate. È possibile eseguire la distribuzione tramite una AWS CodePipeline pipeline su più ambienti, account o regioni, tutti gestibili all'interno della CLI. Utilizzando AWS Copilot è possibile anche eseguire attività operative, come la visualizzazione dei log e lo stato del servizio. AWS Copilot è all-in-one uno strumento che consente di gestire più facilmente le risorse cloud in modo da potersi concentrare sullo sviluppo e sulla gestione delle applicazioni.

Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.

Utilizza il flusso di lavoro completo AWS per end-to-end sviluppatori di Copilot per creare, rilasciare e gestire applicazioni container conformi alle AWS migliori pratiche per l'infrastruttura.

AWS CDK

AWS Cloud Development Kit (AWS CDK) Si tratta di un framework di sviluppo software open source che puoi utilizzare per modellare e fornire le risorse delle tue applicazioni cloud utilizzando linguaggi di programmazione familiari. AWS CDK fornisce le tue risorse in modo sicuro e ripetibile tramite AWS CloudFormation. Utilizzando il CDK, i clienti possono generare il proprio ambiente con meno righe di codice utilizzando lo stesso linguaggio utilizzato per creare la propria applicazione. Amazon ECS fornisce un modulo nel CDK denominato `ecs-patterns`, che crea architetture comuni. Un modello disponibile è `ApplicationLoadBalancedFargateService()`. Questo modello crea

un cluster, una definizione di attività e risorse aggiuntive per eseguire un servizio Amazon ECS con bilanciamento del carico su AWS Fargate.

Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando AWS CDK.

Utilizzatela AWS CDK se desiderate definire l'infrastruttura o l'architettura come codice nel vostro linguaggio di programmazione preferito. Ad esempio, puoi utilizzare lo stesso linguaggio utilizzato per scrivere le applicazioni.

AWS App2Container

A volte i clienti aziendali potrebbero già disporre di applicazioni ospitate in istanze on-premise o EC2 o entrambe. Sono interessati alla portabilità e all'ecosistema degli strumenti dei container, in particolare su Amazon ECS, e devono prima effettuare la containerizzazione. AWS App2Container permette di fare proprio questo. App2Container (A2C) è uno strumento a riga di comando per modernizzare le applicazioni .NET e Java in applicazioni containerizzate. A2C analizza e crea un inventario di tutte le applicazioni in esecuzione su macchine virtuali, on-premise o nel cloud. Dopo aver selezionato l'applicazione che desideri containerizzare, A2C impacchetta l'artefatto dell'applicazione e le dipendenze identificate in immagini del container. Configura quindi le porte di rete e genera il processo Amazon ECS. Infine, crea un CloudFormation modello che puoi distribuire o modificare se necessario.

Per ulteriori informazioni, consulta Nozioni di base su AWS App2Container.

Utilizza App2Container se disponi di applicazioni ospitate su istanze on-premise o Amazon EC2 o entrambe.

CLI di Amazon ECS

L'Amazon ECS CLI ti consente di eseguire le tue applicazioni su Amazon ECS AWS Fargate utilizzando il formato di file Docker Compose. Puoi eseguire rapidamente il provisioning delle risorse, eseguire push e pull delle immagini utilizzando Amazon ECR e monitorare le applicazioni in esecuzione su Amazon ECS o AWS Fargate. È inoltre possibile testare i container in esecuzione localmente insieme ai container nel cloud all'interno della CLI.

Per ulteriori informazioni, consulta Guida introduttiva all'interfaccia a riga di comando di Amazon ECS.

Utilizza la CLI di ECS se dispone di un'applicazione Compose e desideri distribuirla su Amazon ECS oppure testare container locali con container in esecuzione su Amazon ECS nel cloud.

Integrazione di Docker Desktop con Amazon ECS

AWS e Docker hanno collaborato per creare un'esperienza di sviluppo semplificata da utilizzare per distribuire e gestire contenitori su Amazon ECS direttamente utilizzando gli strumenti Docker. Ora puoi creare e testare i tuoi container in locale utilizzando Docker Desktop e Docker Compose e quindi implementarli su Amazon ECS su Fargate. Per iniziare a utilizzare l'integrazione di Amazon ECS e Docker, scarica Docker Desktop e, facoltativamente, registrati per ottenere un ID Docker. Per ulteriori informazioni, consulta Docker Desktop e Registrazione per l'ID Docker.

Gli utenti non esperti di container spesso iniziano a conoscerli utilizzando strumenti Docker come la CLI di Docker e Docker Compose. Ciò rende l'utilizzo del plug-in Docker Compose CLI per Amazon ECS un passaggio successivo naturale nell'esecuzione dei contenitori dopo i test locali. AWS Docker fornisce una spiegazione passo per passo dell'implementazione dei container su Amazon ECS. Per ulteriori informazioni, consulta Docker Compose CLI - Amazon ECS.

Puoi sfruttare le funzionalità aggiuntive di Amazon ECS, come il rilevamento dei servizi, il bilanciamento del carico e altre AWS risorse da utilizzare con le loro applicazioni con Docker Desktop.

Puoi anche scaricare il plug-in Docker Compose CLI per Amazon ECS direttamente da. GitHub Per ulteriori informazioni, consulta il plug-in Docker Compose CLI per Amazon ECS on. GitHub

AWS SDK

Puoi anche utilizzare AWS gli SDK per gestire le risorse e le operazioni di Amazon ECS da una varietà di linguaggi di programmazione. Gli SDK forniscono moduli che aiutano a gestire le attività, incluse le attività nell'elenco seguente.

- Firma crittografica delle richieste di servizio
- Nuovi tentativi di richiesta
- Gestione delle risposte di errore

Per ulteriori informazioni sugli SDK disponibili, consulta Strumenti per Amazon Web Services.

Riepilogo

Con le molte opzioni tra cui scegliere, è possibile scegliere le opzioni più adatte a voi. Considera le seguenti opzioni:

- Se si è orientati visivamente, è possibile creare e gestire visivamente i container utilizzando l'opzione AWS Management Console.
- Se preferisci le CLI, prendi in considerazione l'utilizzo di AWS Copilot o di. AWS CLI In alternativa, se preferisci l'ecosistema Docker, è possibile sfruttare le funzionalità di ECS dall'interno della CLI di Docker per l'implementazione in AWS. Una volta implementate queste risorse, è possibile continuare a gestirle tramite la CLI o visivamente tramite la console.
- Se sei uno sviluppatore, puoi utilizzare il AWS CDK per definire la tua infrastruttura nella stessa lingua dell'applicazione. È possibile utilizzare CDK e AWS Copilot per esportare in CloudFormation modelli in cui è possibile modificare le impostazioni granulari, aggiungere altre AWS risorse e automatizzare le distribuzioni tramite script o una pipeline CI/CD, ad esempio. AWS CodePipeline

Gli SDK o AWS CLI l'API ECS sono strumenti utili per automatizzare le azioni sulle risorse ECS, il che li rende ideali per l'implementazione. Per distribuire le applicazioni utilizzando, AWS CloudFormation puoi utilizzare una varietà di linguaggi di programmazione o un semplice file di testo per modellare e fornire tutte le risorse necessarie per le tue applicazioni. È quindi possibile implementare l'applicazione in più regioni e account in modo automatico e sicuro. Ad esempio, puoi definire il cluster ECS, i servizi, le definizioni delle attività o i fornitori di capacità come codice in un file e distribuirli tramite i comandi. AWS CLI CloudFormation

Per eseguire attività operative, è possibile visualizzare e gestire le risorse in modo programmatico utilizzando l'SDK o l' AWS CLI API ECS. Comandi come `describe-tasks` o `list-services` visualizzano i metadati più recenti o un elenco di tutte le risorse. Analogamente alle implementazioni, i clienti possono scrivere un'automazione che include comandi come `update-service` per fornire azioni correttive all'individuazione di una risorsa che si è arrestata in modo imprevisto. Puoi anche gestire i tuoi servizi utilizzando Copilot. AWS Comandi come `copilot svc logs` o `copilot app show` forniscono dettagli su ciascuno dei tuoi microservizi o sulla tua applicazione nel suo complesso.

I clienti possono utilizzare uno qualsiasi degli strumenti disponibili menzionati in questo documento e utilizzarli in varie combinazioni. Gli strumenti ECS offrono diversi percorsi per passare da determinati strumenti ad altri più evoluti che si adattano alle esigenze in continua evoluzione. Ad esempio, è possibile scegliere un controllo più granulare sulle risorse o una maggiore automazione in base alle

esigenze. ECS offre anche una vasta gamma di strumenti per una vasta gamma di esigenze e livelli di competenza.

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

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

AWS Attualmente Copilot supporta i sistemi Linux, macOS e Windows. Per ulteriori informazioni sulla versione più recente della AWS CLI di Copilot, consulta 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 GitHubdove puoi aprire problemi, fornire feedback e segnalare bug.

Per informazioni sull'installazione della AWS CLI di Copilot, vedere. Installazione della CLI di AWS Copilot Per informazioni sulla distribuzione di un'app di esempio, consulta. Distribuzione di un'applicazione Amazon ECS di esempio utilizzando la AWS CLI Copilot La documentazione aggiuntiva per la AWS CLI di Copilot è disponibile sul sito Web di Copilot.AWS

Installazione della CLI di AWS Copilot

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

Usa Homebrew

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

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

Scarica il file binario

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

macOS

Per macOS:

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

Per sistemi macOS ARM:

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

Linux

Per sistemi Linux x86 (64 bit):

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

Per i sistemi Linux ARM:

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

Windows

Tramite Powershell, esegui il comando riportato:

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

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

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

1. Scarica e installa GnuPG. Per ulteriori informazioni, consulta il sito Web GnuPG .

   macOS

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

   ```
   brew install gnupg
   ```

   Linux

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

   Windows

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

   Per ulteriori informazioni, consulta la pagina relativa al download di GnuPG.

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

   macOS

   ```
   echo $PATH
   ```

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

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

   Linux

   ```
   echo $PATH
   ```

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

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

   Windows

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

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

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

3. Crea un file di testo locale normale.

   macOS

   Sul terminale, inserisci:

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

   Apri il file con. TextEdit

Linux

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

Windows

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

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

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v2

mQINBFq1SasBEADliGcT1NVJ1ydfN8DqebYYe9ne3dt6jqKFmKowLmm6LLGJe7HU
jGtqhCWRDkN+qPpHqdArRgDZAtn2pXY5fEipHgar4CP8QgRnRMO2fl74lmavr4Vg
7K/KH8VHlq2uRw32/B94XLEgRbGTMdWFdKuxoPCttBQaMj3LGn6Pe+6xVWRkChQu
BoQAhjBQ+bEm0kNy0LjNgjNlnL3UMAG56t8E3LANIgGgEnpNsB1UwfWluPoGZoTx
N+6pHBJrKIL/1v/ETU4FXpYw2zvhWNahxeNRnoYj3uycHkeliCrw4kj0+skizBgO
2K7oVX8Oc3j5+ZilhL/qDLXmUCb2az5cMM1mOoF8EKX5HaNuq1KfwJxqXE6NNIcO
lFTrT7QwD5fMNld3FanLgv/ZnIrsSaqJOL6zRSq8O4LN1OWBVbndExk2Kr+5kFxn
5lBPgfPgRj5hQ+KTHMa9Y8Z7yUc64BJiN6F9Nl7FJuSsfqbdkvRLsQRbcBG9qxX3
rJAEhieJzVMEUNl+EgeCkxj5xuSkNU7zw2c3hQZqEcrADLV+hvFJktOz9Gm6xzbq
lTnWWCz4xrIWtuEBA2qE+MlDheVd78a3gIsEaSTfQq0osYXaQbvlnSWOoc1y/5Zb
zizHTJIhLtUyls9WisP2s0emeHZicVMfW61EgPrJAiupgc7kyZvFt4YwfwARAQAB
tCRBbWF6b24gRUNTIDxlY3Mtc2VjdXJpdHlAYW1hem9uLmNvbT6JAhwEEAECAAYF
AlrjL0YACgkQHivRXs0TaQrg1g/+JppwPqHnlVPmv7lessB8I5UqZeD6p6uVpHd7
Bs3pcPp8BV7BdRbs3sPLt5bV1+rkqOlw+0gZ4Q/ue/YbWtOAt4qY0OcEo0HgcnaX
lsB827QIfZIVtGWMhuh94xzm/SJkvngml6KB3YJNnWP61A9qJ37/VbVVLzvcmazA
McWB4HUMNrhd0JgBCo0gIpqCbpJEvUc02Bjn23eEJsS9kC7OUAHyQkVnx4d9UzXF
4OoISF6hmQKIBoLnRrAlj5Qvs3GhvHQ0ThYq0Grk/KMJJX2CSqt7tWJ8gk1n3H3Y
SReRXJRnv7DsDDBwFgT6r5Q2HW1TBUvaoZy5hF6maD09nHcNnvBjqADzeT8Tr/Qu
bBCLzkNSYqqkpgtwv7seoD2P4n1giRvDAOEfMZpVkUr+C252IaH1HZFEz+TvBVQM
Y8OWWxmIJW+J6evjo3N1eO19UHv71jvoF8zljbI4bsL2c+QTJmOv7nRqzDQgCWyp
Id/v2dUVVTk1j9omuLBBwNJzQCB+72LcIzJhYmaP1HC4LcKQG+/f41exuItenatK
lEJQhYtyVXcBlh6Yn/wzNg2NWOwb3vqY/F7m6u9ixAwgtIMgPCDE4aJ86zrrXYFz
N2HqkTSQh77Z8KPKmyGopsmN/reMuilPdINb249nA0dzoN+nj+tTFOYCIaLaFyjs
Z0r1QAOJAjkEEwECACMFAlq1SasCGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIX
gAAKCRC86dmkLVF4T9iFEACEnkm1dNXsWUx34R3c0vamHrPxvfkyI1FlEUen8D1h
uX9xy6jCEROHWEp0rjGK4QDPgM93sWJ+s1UAKg214QRVzft0y9/DdR+twApA0fzy
uavIthGd6+03jAAo6udYDE+cZC3P7XBbDiYEWk4XAF9I1JjB8hTZUgvXBL046JhG
eM17+crgUyQeetkiOQemLbsbXQ40Bd9V7zf7XJraFd8VrwNUwNb+9KFtgAsc9rk+
YIT/PEf+YOPysgcxI4sTWghtyCulVnuGoskgDv4v73PALU0ieUrvvQVqWMRvhVx1
```

```
0X90J7cC1KOyhlEQQ1aFTgmQjmXexVTwIBm8LvysFK6YXM41KjOrlz3+6xBIm/qe
bFyLUnf4WoiuOplAaJhK9pRY+XEnGNxdtN4D26Kd0F+PLkm3Tr3Hy3b1Ok34FlGr
KVHUq1TZD7cvMnnNKEELTUcKX+1mV3an16nmAg/my1JSUt6BNK2rJpY1s/kkSGSE
XQ4zuF2IGCpvBFhYAlt5Un5zwqkwwQR3/n2kwAoDzonJcehDw/C/cGos5D0aIU7I
K2X2aTD3+pA7Mx3IMe2hqmYqRt9X42yF1PIEVRneBRJ3HDezAgJrNh0GQWRQkhIx
gz6/cTR+ekr5TptVszS9few2GpI5bCgBKBisZIssT89aw7mAKWut0Gcm4qM9/yK6
1bkCDQRatUmrARAAxNPvVwreJ2yAiFcUpdRlVhsuOgnxvs1QgsIw3H7+Pacr9Hpe
8uftYZqdC82KeSKhpHq7c8gMTMucIINtH25x9BCc73E33EjCL9Lqov1TL7+QkgHe
T+JIhZwdD8Mx2K+LVVVu/aWkNrfMuNwyDUciSI4D5QHa8T+F8fgN4OTpwYjirzel
5yoICMr9hVcbzDNv/ozKCxjx+XKgnFc3wrnDfJfntfDAT7ecwbUTL+viQKJ646s+
psiqXRYtVvYInEhLVrJ0aV6zHFoigE/Bils6/g7ru1Q6CEHqEw++APs5CcE8VzJu
WAGSVHZgun5Y9N4quR/M9Vm+IPMhTxrAg7rOvyRN9cAXfeSMf77I+XTifigNna8x
t/MOdjXr1fjF4pThEi5u6WsuRdFwjY2azEv3vevodTi4HoJReH6dFRa6y8c+UDgl
2iHiOKIpQqLbHEfQmHcDd2fix+AaJKMnPGNku9qCFEMbgSRJpXz6BfwnY1QuKE+I
R6jA0frUNt2jhiGG/F8RceXzohaaC/Cx7LUCUFWc0n7z32C9/Dtj7I1PMOacdZzz
bjJzRKO/ZDv+UN/c9dwAkllzAyPMwGBkUaY68EBstnIliW34aWm6IiHhxioVPKSp
VJfyiXPO0EXqujtHLAeChfjcns3I12YshT1dv2PafG53fp33ZdzeUgsBo+EAEQEA
AYkCHwQYAQIACQUCWrVJqwIbDAAKCRC86dmkLVF4T+ZdD/9x/8APzgNJF3o3STrF
jvnV1ycyhWYGAeBJiu7wjsNWwzMFOv15tLjB7AqeVxZn+WKDD/mIOQ45OZvnYZuy
X7DR0JszaH9wrYTxZLVruAu+t6UL0y/XQ4L1GZ9QR6+r+7t1Mvbfy7BlHbvX/gYt
Rwe/uwdibI0CagEzyX+2D3kTOlHO5XThbXaNf8AN8zha91Jt2Q2UR2X5T6JcwtMz
FBvZnl3LSmZyE0EQehS2iUurU4uWOpGppuqVnbi0jbCvCHKgDGrqZ0smKNAQng54
F365W3g8AfY48s8XQwzmcliowYX9bT8PZiEi0J4QmQh0aXkpqZyFefuWeOL2R94S
XKzr+gRh3BAULoqF+qK+IUMxTip9KTPNvYDpiC66yBiT6gFDji5Ca9pGpJXrC3xe
TXiKQ8DBWDhBPVPrruLIaenTtZEOsPc4I85yt5U9RoPTStcOr34s3w5yEaJagt6S
Gc5r9ysjkfH6+6rbi1ujxMgROSqtqr+RyB+V9A5/OgtNZc8llK6u4UoOCde8jUUW
vqWKvjJB/Kz3u4zaeNu2ZyyHaOqOuH+TETcW+jsY9IhbEzqN5yQYGi4pVmDkY5vu
lXbJnbqPKpRXgM9BecV9AMbPgbDq/5LnHJJXg+G8YQOgp4lR/hC1TEFdIp5wM8AK
CWsENyt2o1rjgMXiZOMF8A5oBLkCDQRatUuSARAAr77kj7j2QR2SZeOSlFBvV7oS
mFeSNnz9xZssqrsm6bTwSHM6YLDwc7Sdf2esDdyzONETwqrVCg+FxgL8hmo9hS4c
rR6tmrP0mOmptr+xLLsKcaP7ogIXsyZnrEAEsvW8PnfayoiPCdc3cMCR/lTnHFGA
7EuR/XLBmi7Qg9tByVYQ5Yj5wB9V4B2yeCt3XtzPqeLKvaxl7PNelaHGJQY/xo+m
V0bndxf9IY+4oFJ4blD32WqvyxESo7vW6WBh7oqv3Zbm0yQrr8a6mDBpqLkvWwNI
3kpJR974tg5o5LfDu1BeeyHWPSGm4U/G4JB+JIG1ADy+RmoWEt4BqTCZ/knnoGvw
D5sTCxbKdmuOmhGyTssoG+3OOcGYHV7pWYPhazKHMPm201xKCjH1RfzRULzGKjD+
yMLT1I3AXFmLmZJXikAOlvE3/wgMqCXscbycbLjLD/bXIuFWo3rzoezeXjgi/DJx
jKBAyBTYO5nMcth1O9oaFd9d0HbsOUDkIMnsgGBE766Piro6MHo0T0rXl07Tp4pI
rwuSOsc6XzCzdImj0Wc6axS/HeUKRXWdXJwno5awTwXKRJMXGfhCvSvbcbc2Wx+L
IKvmB7EB4K3fmjFFE67yolmiw2qRcUBfygtH3eL5XZU28MiCpue8Y8GKJoBAUyvf
KeM1rO8Jm3iRAc5a/D0AEQEAAYkEPgQYAQIACQUCWrVLkgIbAgIpCRC86dmkLVF4
T8FdIAQZAQIABgUCWrVLkgAKCRDePL1hra+LjtHYD/9MucxdFe6bXO1dQR4tKhhQ
P0LRqy6zlBY9ILCLowNdGZdqorogUiUymgn3VhEhVtxTOoHcN7qOuM01PNsRnOeS
EYjf8Xrb1clzkD6xULwmOclTb9bBxnBc/4PFvHAbZW3QzusaZniNgkuxt6BTfloS
Of4inq71kjmGK+TlzQ6mUMQUg228NUQC+a84EPqYyAeY1sgvgB7hJBhYL0QAxhcW
```

```
6m20Rd8iEc6HyzJ3yCOCsKip/nRWAbf0OvfHfRBp0+m0ZwnJM8cPRFjOqqzFpKH9
HpDmTrC4wKP1+TL52LyEqNh4yZitXmZNV7giSRIkk0eDSko+bFy6VbMzKUMkUJK3
D3eHFAMkujmbfJmSMTJOPGn5SB1HyjCZNx6bhIIbQyEUB9gKCmUFaqXKwKpF6rj0
iQXAJxLR/shZ5Rk96VxzOphUl7T90m/PnUEEPwq8KsBhnMRgxa0RFidDP+n9fgtv
HLmrOqX9zBCVXh0mdWYLrWvmzQFWzG7AoE55fkf8nAEPsalrCdtaNUBHRXA0OQxG
AHMOdJQQvBsmqMvuAdjkDWpFu5y0My5ddU+hiUzUyQLjL5Hhd5LOUDdewlZgIw1j
xrEAUzDKetnemM8GkHxDgg8koev5frmShJuce7vSjKpCNg3EIJSgqMOPFjJuLWtZ
vjHeDNbJy6uNL65ckJy6WhGjEADS2WAW1D6Tfekkc21SsIXk/LqEpLMR/0g5OUif
wcEN1rS9IJXBwIy8MelN9qr5KcKQLmfdfBNEyyceBhyVl0MDyHOKC+7PofMtkGBq
13QieRHv5GJ8LB3fclqHV8pwTTo3Bc8z2g0TjmUYAN/ixETdReDoKavWJYSE9yoM
aaJu279ioVTrwpECse0XkiRyKToTjwOb73CGkBZZpJyqux/rmCV/fp4ALdSW8zbz
FJVORaivhoWwzjpfQKhwcU9lABXi2UvVm14v0AfeI7oiJPSU1zM4fEny4oiIBXlR
zhFNih1UjIu82X16mTm3BwbIga/s1fnQRGzyhqUIMii+mWra23EwjChaxpvjjcUH
5ilLc5Zq781aCYRygYQw+hu5nFkOH1R+Z50Ubxjd/aqUfnGIAX7kPMD3Lof4KldD
Q8ppQriUvxVo+4nPV6rpTy/PyqCLWDjkguHpJsEFsMkwajrAz0QNSAU5CJ0G2Zu4
yxvYlumHCEl7nbFrm0vIiA75Sa8KnywTDsyZsu3XcOcf3g+g1xWTpjJqy2bYXlqz
9uDOWtArWHOis6bq8l9RE6xr1RBVXS6uqgQIZFBGyq66b0dIq4D2JdsUvgEMaHbc
e7tBfeB1CMBdA64e9Rq7bFR7Tvt8gasCZYlNr3lydh+dFHIEkH53HzQe6l88HEic
+0jVnLkCDQRa55wJARAAyLya2Lx6gyoWoJN1a6740q3o8e9d4KggQOfGMTCflmeq
ivuzgN+3DZHN+9ty2KxXMtn0mhHBerZdbNJyjMNT1gAgrhPNB4HtXBXum2wS57WK
DNmade914L7FWTPAWBG2Wn448OEHTqsClICXXWy9IICgclAEyIq0Yq5mAdTEgRJS
Z8t4GpwtDL9gNQyFXaWQmDmkAsCygQMvhAlmu9xOIzQG5CxSnZFk7zcuL60k14Z3
Cmt49k4T/7ZU8goWi8tt+rU78/IL3J/fF9+1civ1OwuUidgfPCSvOUW1JojsdCQA
L+RZJcoXq7lfOFj/eNjeOSstCTDPfTCL+kThE6E5neDtbQHBYkEX1BRiTedsV4+M
ucgiTrdQFWKf89G72xdv8ut9AYYQ2BbEYU+JAYhUH8rYYui2dHKJIgjNvJscuUWb
+QEqJIRleJRhrO+/CHgMs4fZAkWF1VFhKBkcKmEjLn1f7EJJUUW84ZhKXjO/AUPX
1CHsNjziRceuJCJYox1cwsoq6jTE50GiNzcIxTn9xUc0UMKFeggNAFys1K+TDTm3
Bzo8H5ucjCUEmUm9lhkGwqTZgOlRX5eqPX+JBoSaObqhgqCa5IPinKRa6MgoFPHK
6sYKqroYwBGgZm6Js5chpNchvJMs/3WXNOEVg0J3z3vP0DMhxqWm+r+n9zlW8qsA
EQEAAYkEPgQYAQgACQUCWuecCQIbAgIpCRC86dmkLVF4T8FdIAQZAQgABgUCWuec
CQAKCRBQ3szEcQ5hr+ykD/4tOLRHFHXuKUcxgGaubUcVtsFrwBKma1cYjqaPms8u
6Sk0wfGRI32G/GhOrp0Ts/MOkbObq6VLTh8N5Yc/53MEl8zQFw9Y5AmRoW4PZXER
ujs5s7p4oR7xHMihMjCCBn1bvrR+34YPfgzTcgLiOEFHYT8UTxwnGmXOvNkMM7md
xD3CV5q6VAte8WKBo/220II3fcQlc9r/oWX4kXXkb0v9hoGwKbDJ1tzqTPrp/xFt
yohqnvImpnlz+Q9zXmbrWYL9/g8VCmW/NN2gju2G3Lu/TlFUWIT4v/5OPK6TdeNb
VKJO4+S8bTayqSG9CML1S57KSgCo5HUhQWeSNHI+fpe5oX6FALPT9JLDce8OZz1i
cZZ0MELP37mOOQun0AlmHm/hVzf0f311PtbzcqWaE51tJvgUR/nZFo6Ta3O5Ezhs
3VlEJNQ1Ijf/6DH87SxvAoRIARCuZd0qxBcDK0avpFzUtbJd24lRA3WJpkEiMqKv
RDVZkE4b6TW61f0o+LaVfK6E8oLpixegS4fiqC16mFrOdyRk+RJJfIUyz0WTDVmt
g0U1CO1ezokMSqkJ7724pyjr2xf/r9/sC6aOJwB/lKgZkJfC6NqL7TlxVA31dUga
LEOvEJTTE4gl+tYtfsCDvALCtqL0jduSkUo+RXcBItmXhA+tShW0pbS2Rtx/ixua
KohVD/0R4QxiSwQmICNtm9mw9ydIl1yjYXX5a9x4wMJracNY/LBybJPFnZnT4dYR
z4XjqysDwvvYZByaWoIe3QxjX84V6MlI2IdAT/xImu8gbaCI8tmyfpIrLnPKiR9D
VFYfGBXuAX7+HgPPSFtrHQONCALxxzlbNpS+zxt9r0MiLgcLyspWxSdmoYGZ6nQP
```

```
RO5Nm/ZVS+u2imPCRzNUZEMa+dlE6kHx0rS0dPiuJ4O7NtPeYDKkoQtNagspsDvh
cK7CSqAiKMq06UBTxqlTSRkm62eOCtcs3p3OeHu5GRZF1uzTET0ZxYkaPgdrQknx
ozjP5mC7X+45lcCfmcVt94TFNL5HwEUVJpmOgmzILCI8yoDTWzloo+i+fPFsXX4f
kynhE83mSEcr5VHFYrTY3mQXGmNJ3bCLuc/jq7ysGq69xiKmTlUeXFm+aojcRO5i
zyShIRJZ0GZfuzDYFDbMV9amA/YQGygLw//zP5ju5SW26dNxlf3MdFQE5JJ86rn9
MgZ4gcpazHEVUsbZsgkLizRp9imUiH8ymLqAXnfRGlU/LpNSefnvDFTtEIRcpOHc
bhayG0bk51Bd4mioOXnIsKy4j63nJXA27x5EVVHQ1sYRN8Ny4Fdr2tMAmj2O+X+J
qX2yy/UX5nSPU492e2CdZ1UhoU0SRFY3bxKHKB7SDbVeav+K5g==
=Gi5D
-----END PGP PUBLIC KEY BLOCK-----
```

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

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

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

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

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

   macOS

   Per i sistemi macOS, utilizza il seguente comando.

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

   Linux

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

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

Per i sistemi ARM Linux, esegui il comando riportato.

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

Windows

Tramite Powershell, esegui il comando riportato.

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

7. Verifica la firma utilizzando il seguente comando:

   - Per sistemi macOS e Linux:

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

   - Per i sistemi Windows:

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

   Output previsto:

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

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

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

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

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

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

Prerequisiti

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

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

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

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

```
git clone https://github.com/aws-samples/aws-copilot-sample-service.git demo-app &&
 \
cd demo-app &&                                  \
```

```
copilot init --app demo                        \
  --name api                                   \
  --type 'Load Balanced Web Service'           \
  --dockerfile './Dockerfile'                  \
  --port 80                                    \
  --deploy
```

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

   - Elenca tutte le tue applicazioni AWS Copilot.

     ```
     copilot app ls
     ```

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

     ```
     copilot app show
     ```

   - Mostra informazioni sugli ambienti in uso.

     ```
     copilot env ls
     ```

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

     ```
     copilot svc show
     ```

   - Elenca tutti i servizi in un'applicazione.

     ```
     copilot svc ls
     ```

   - Mostra i log di un servizio implementato.

     ```
     copilot svc logs
     ```

   - Mostra lo stato del servizio.

     ```
     copilot svc status
     ```

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

```
copilot app delete
```

Creazione di risorse Amazon ECS utilizzando AWS CDK

AWS Cloud Development Kit (AWS CDK) Si tratta di un framework Infrastructure-as-Code (IAC) che puoi utilizzare per definire l'infrastruttura AWS cloud utilizzando un linguaggio di programmazione a tua scelta. Per definire la tua infrastruttura cloud, devi innanzitutto scrivere un'app (in uno dei linguaggi supportati dal CDK) contenente uno o più stack. Quindi, lo sintetizzi in un modello e distribuisci le tue risorse sul tuo. AWS CloudFormation 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 modulo `aws-ecs-patterns` include anche costrutti che utilizzano un Network Load Balancer e sono eseguiti su Amazon EC2.

Prima di iniziare questa attività, configurate l'ambiente di AWS CDK sviluppo e installatelo AWS CDK eseguendo il comando seguente. Per istruzioni su come configurare l'ambiente di AWS CDK sviluppo, consulta Getting Started With the AWS CDK - Prerequisiti.

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

> **Note**
>
> Queste istruzioni presuppongono che tu stia usando AWS CDK v2.

Argomenti

- Fase 1: Configura il tuo progetto AWS CDK

Fase 1: Configura il tuo progetto AWS CDK

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

TypeScript

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

JavaScript

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

Python

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

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

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

Java

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

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

C#

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

Go

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

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

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

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

TypeScript

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

JavaScript

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

Python

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

Java

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

C#

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

Go

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

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

Usa l'immagine del contenitore da. `amazon-ecs-sample` DockerHub Questa immagine contiene un'app Web PHP in esecuzione su Amazon Linux 2.

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

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

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

TypeScript

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

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

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

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

JavaScript

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

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

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

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

```
      },
      publicLoadBalancer: true
    });
  }
}

module.exports = { HelloEcsStack }
```

Python

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

```
import aws_cdk as cdk
from constructs import Construct

import aws_cdk.aws_ecs as ecs
import aws_cdk.aws_ecs_patterns as ecsp

class HelloEcsStack(cdk.Stack):

    def __init__(self, scope: Construct, construct_id: str, **kwargs) -> None:
        super().__init__(scope, construct_id, **kwargs)

        ecsp.ApplicationLoadBalancedFargateService(self, "MyWebServer",
            task_image_options=ecsp.ApplicationLoadBalancedTaskImageOptions(
                image=ecs.ContainerImage.from_registry("amazon/amazon-ecs-sample")),
            public_load_balancer=True
        )
```

Java

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

```
package com.myorg;

import software.constructs.Construct;
import software.amazon.awscdk.Stack;
import software.amazon.awscdk.StackProps;

import software.amazon.awscdk.services.ecs.ContainerImage;
import
 software.amazon.awscdk.services.ecs.patterns.ApplicationLoadBalancedFargateService;
```

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

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

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

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

C#

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

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

```
                    PublicLoadBalancer = true
                });
        }
    }
}
```

Go

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

```
package main

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

type HelloEcsStackProps struct {
 awscdk.StackProps
}

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

 // The code that defines your stack goes here

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

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

 return stack
}

func main() {
 defer jsii.Close()

 app := awscdk.NewApp(nil)

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

 app.Synth(nil)
}

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

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

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

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

Il frammento breve precedente include quanto segue:

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

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

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

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

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

```
Resources:
  MyWebServerLB3B5FD3AB:
    Type: AWS::ElasticLoadBalancingV2::LoadBalancer
```

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

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

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

```
Outputs:
HelloEcsStack.MyWebServerLoadBalancerDNSXXXXXXX = Hello-MyWeb-ZZZZZZZZZZZZZ-
ZZZZZZZZZZ.us-west-2.elb.amazonaws.com
HelloEcsStack.MyWebServerServiceURLYYYYYYYY = http://Hello-MyWeb-ZZZZZZZZZZZZZ-
ZZZZZZZZZZ.us-west-2.elb.amazonaws.com
```

Fase 3: Test del server Web

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

Fase 4: pulizia

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

Passaggi successivi

Per ulteriori informazioni su come sviluppare l' AWS infrastruttura utilizzando 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 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. In questo modo, puoi dedicare meno tempo alla creazione e alla gestione delle risorse e dell'infrastruttura. Utilizzando AWS CloudFormation, puoi creare un modello che descriva tutte le AWS risorse che desideri, ad esempio cluster Amazon ECS specifici. Quindi, AWS CloudFormation effettuerà il provisioning e la configurazione di tali risorse.

Quando lo usi AWS CloudFormation, puoi riutilizzare il modello per configurare le risorse Amazon ECS in modo coerente e ripetibile. Descrivi le tue risorse una sola volta, quindi esegui nuovamente il provisioning delle stesse risorse su più e. Account AWS Regioni AWS

AWS CloudFormation modelli

Per effettuare il provisioning e configurare le risorse per Amazon ECS e i servizi correlati, assicurati di conoscere i AWS CloudFormation modelli. AWS CloudFormation i modelli sono file di testo in formato JSON o YAML che descrivono le risorse che desideri fornire nei tuoi stack. AWS CloudFormation

Se non conosci il formato JSON o YAML o entrambi, puoi usare Designer per iniziare a utilizzare i modelli. AWS CloudFormation AWS CloudFormation Per ulteriori informazioni, consulta Cos'è Designer? AWS CloudFormation nella Guida AWS CloudFormation per l'utente.

Amazon ECS supporta la creazione di cluster, definizioni di processo, servizi e set di processi in AWS CloudFormation. I seguenti esempi illustrano come creare risorse con questi modelli utilizzando la AWS CLI. Puoi creare queste risorse anche utilizzando la console AWS CloudFormation . Per ulteriori informazioni su come creare risorse utilizzando la console AWS CloudFormation , consulta laGuida per l'utente di AWS CloudFormation.

Modelli di esempio

Creazione di risorse Amazon ECS mediante stack separati

Negli esempi seguenti viene illustrato come creare risorse Amazon ECS utilizzando stack separati per ciascuna risorsa.

Definizioni di processi

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

JSON

```
  {
    "AWSTemplateFormatVersion": "2010-09-09",
    "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": "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::aws_account_id:role/
ecsTaskExecutionRole",
                "Family": "task-definition-cfn",
                "Memory": 512,
                "NetworkMode": "awsvpc",
                "RequiresCompatibilities": [
                    "FARGATE"
                ],
                "RuntimePlatform": {
                    "OperatingSystemFamily": "LINUX"
                }
            }

        }
        }
    }
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
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: '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::aws_account_id:role/ecsTaskExecutionRole'
    Family: task-definition-cfn
    Memory: 512
    NetworkMode: awsvpc
    RequiresCompatibilities:
      - FARGATE
    RuntimePlatform:
      OperatingSystemFamily: LINUX
```

Cluster

Per creare un cluster vuoto, 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
```

Creazione di più risorse Amazon ECS in un unico stack

Per creare più risorse Amazon ECS in un unico stack, puoi utilizzare il seguente modello di esempio. Il modello crea un cluster Amazon ECS denominato `CFNCluster`. Il cluster contiene una definizione di attività Linux Fargate che configura un server Web. Il modello crea anche un servizio denominato `cfn-service` che avvia e mantiene la definizione di attività. Prima di utilizzare questo modello, assicurati che la sottorete e gli ID del gruppo di sicurezza nella `NetworkConfiguration` del servizio appartengano tutti allo stesso VPC e che il gruppo di sicurezza disponga delle regole necessarie. Per ulteriori informazioni sulle regole dei gruppi di sicurezza, consulta Regole del gruppo di sicurezza nella Guida per l'utente di Amazon VPC.

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "ECSCluster": {
```

```
            "Type": "AWS::ECS::Cluster",
            "Properties": {
                "ClusterName": "CFNCluster"
            }
        },
        "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": "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::aws_account_id::role/
ecsTaskExecutionRole",
                "Family": "task-definition-cfn",
                "Memory": 512,
                "NetworkMode": "awsvpc",
                "RequiresCompatibilities": [
                    "FARGATE"
                ],
                "RuntimePlatform": {
                    "OperatingSystemFamily": "LINUX"
                }
            }
        },
        "ECSService": {
            "Type": "AWS::ECS::Service",
            "Properties": {
                "ServiceName": "cfn-service",
                "Cluster": {
                    "Ref": "ECSCluster"
                },
                "DesiredCount": 1,
                "LaunchType": "FARGATE",
                "NetworkConfiguration": {
                    "AwsvpcConfiguration": {
                        "AssignPublicIp": "ENABLED",
                        "SecurityGroups": [
                            "sg-abcdef01234567890"
                        ],
                        "Subnets": [
                            "subnet-abcdef01234567890"
                        ]
                    }
                },
                "TaskDefinition": {
                    "Ref": "ECSTaskDefinition"
                }
            }
        }
    }
}
```

YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  ECSCluster:
    Type: 'AWS::ECS::Cluster'
    Properties:
      ClusterName: CFNCluster
  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: '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::aws_account_id:role/ecsTaskExecutionRole'
      Family: task-definition-cfn
      Memory: 512
      NetworkMode: awsvpc
      RequiresCompatibilities:
        - FARGATE
```

```
      RuntimePlatform:
        OperatingSystemFamily: LINUX
  ECSService:
    Type: 'AWS::ECS::Service'
    Properties:
      ServiceName: cfn-service
      Cluster: !Ref ECSCluster
      DesiredCount: 1
      LaunchType: FARGATE
      NetworkConfiguration:
        AwsvpcConfiguration:
          AssignPublicIp: ENABLED
          SecurityGroups:
            - sg-abcdef01234567890
          Subnets:
            - subnet-abcdef01234567890
      TaskDefinition: !Ref ECSTaskDefinition
```

Utilizzo di AWS CLI per creare risorse a partire da modelli

Il comando seguente crea uno stack denominato `ecs-stack` utilizzando un file del corpo del modello denominato `ecs-template-body.json`. Assicurati che il file del corpo del modello sia in formato JSON o YAML. La posizione del file è specificata nel parametro `--template-body`. In questo caso, il file del corpo del modello si trova nella directory corrente.

```
aws cloudformation create-stack \
      --stack-name ecs-stack \
      --template-body file://ecs-template-body.json
```

Per assicurarti che le risorse siano create correttamente, controlla la console Amazon ECS o, in alternativa, usa i seguenti comandi:

- Il comando seguente elenca tutte le definizioni di attività.

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

- Il comando seguente elenca tutti i cluster.

```
aws ecs list-clusters
```

- Il comando seguente elenca tutti i servizi definiti nel cluster *CFNCluster*. Sostituisci *CFNCluster* con il nome del cluster in cui desideri creare il servizio.

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

Scopri di più su AWS CloudFormation

Per ulteriori informazioni AWS CloudFormation, consulta le seguenti risorse:

- AWS CloudFormation
- AWS CloudFormation Guida per l'utente
- AWS CloudFormation Guida per l'utente dell'interfaccia a riga di comando

Guida introduttiva all'interfaccia a riga di comando di Amazon ECS

Amazon ECS ha rilasciato AWS Copilot, uno strumento di interfaccia a riga di comando (CLI) che semplifica la creazione, il rilascio e il funzionamento di applicazioni containerizzate pronte per la produzione su Amazon ECS da un ambiente di sviluppo locale. Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.

L'interfaccia a riga di comando (CLI) di Amazon Elastic Container Service (Amazon ECS) offre comandi di alto livello per semplificare la creazione, l'aggiornamento e il monitoraggio di cluster e processi in un ambiente di sviluppo locale. La CLI di Amazon ECS supporta i file di Docker Compose, una specifica open source molto diffusa per la definizione e l'esecuzione di applicazioni multi-container. Utilizza la CLI ECS come parte del ciclo di sviluppo e test quotidiano in alternativa alla AWS Management Console.

Al momento, la versione più recente della CLI di Amazon ECS supporta solo le versioni principali della sintassi del file di Docker Compose versioni 1, 2 e 3. La versione specificata nel file Compose deve essere la stringa `"1"`, `"1.0"`, `"2""2.0"`, `"3"` o `"3.0"`. Le versioni secondarie di Docker Compose non sono supportate.

Il codice sorgente per la CLI di Amazon ECS è disponibile su. GitHub Questo strumento non è più sviluppato attivamente.

Installazione della CLI di Amazon ECS

Amazon ECS ha rilasciato AWS Copilot, uno strumento di interfaccia a riga di comando (CLI) che semplifica la creazione, il rilascio e il funzionamento di applicazioni containerizzate pronte per la produzione su Amazon ECS da un ambiente di sviluppo locale. Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.

Segui queste istruzioni per installare la CLI di Amazon ECS su sistemi macOS, Linux o Windows.

Installazione della CLI di Amazon ECS

1. Scarica il file binario della CLI di Amazon ECS.

   macOS

   ```
   sudo curl -Lo /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/
   ecs-cli-darwin-amd64-latest
   ```

   Linux

   ```
   sudo curl -Lo /usr/local/bin/ecs-cli https://amazon-ecs-cli.s3.amazonaws.com/
   ecs-cli-linux-amd64-latest
   ```

   Windows

   Apri Windows e inserisci i seguenti comandi. PowerShell

   > **Note**
   >
   > Se riscontri problemi di autorizzazione, assicurati di disporre dell'accesso come amministratore su Windows e di PowerShell lavorare come amministratore.

   ```
   New-Item -Path 'C:\Program Files\Amazon\ECSCLI' -ItemType Directory
   Invoke-WebRequest -OutFile 'C:\Program Files\Amazon\ECSCLI\ecs-cli.exe' https://
   amazon-ecs-cli.s3.amazonaws.com/ecs-cli-windows-amd64-latest.exe
   ```

2. Verifica la CLI di Amazon ECS tramite le firme PGP. I file eseguibili della CLI di Amazon ECS vengono firmati crittograficamente tramite firme PGP. Le firme PGP possono essere utilizzate per verificare la validità dell'eseguibile della CLI di Amazon ECS. Puoi completare le fasi seguenti per verificare le firme tramite lo strumento GnuPG.

   a. 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.

   b. 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>"
```

c. 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`

d. 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+qPpHqdArRgDZAtn2pXY5fEipHgar4CP8QgRnRMO2fl74lmavr4Vg
7K/KH8VHlq2uRw32/B94XLEgRbGTMdWFdKuxoPCttBQaMj3LGn6Pe+6xVWRkChQu
BoQAhjBQ+bEm0kNy0LjNgjNlnL3UMAG56t8E3LANIgGgEnpNsB1UwfWluPoGZoTx
N+6pHBJrKIL/1v/ETU4FXpYw2zvhWNahxeNRnoYj3uycHkeliCrw4kj0+skizBgO
2K7oVX8Oc3j5+ZilhL/qDLXmUCb2az5cMM1mOoF8EKX5HaNuq1KfwJxqXE6NNIcO
lFTrT7QwD5fMNld3FanLgv/ZnIrsSaqJOL6zRSq8O4LN1OWBVbndExk2Kr+5kFxn
5lBPgfPgRj5hQ+KTHMa9Y8Z7yUc64BJiN6F9Nl7FJuSsfqbdkvRLsQRbcBG9qxX3
rJAEhieJzVMEUNl+EgeCkxj5xuSkNU7zw2c3hQZqEcrADLV+hvFJktOz9Gm6xzbq
lTnWWCz4xrIWtuEBA2qE+MlDheVd78a3gIsEaSTfQq0osYXaQbvlnSWOoc1y/5Zb
zizHTJIhLtUyls9WisP2s0emeHZicVMfW61EgPrJAiupgc7kyZvFt4YwfwARAQAB
tCRBbWF6b24gRUNTIDxlY3Mtc2VjdXJpdHlAYW1hem9uLmNvbT6JAhwEEAECAAYF
AlrjL0YACgkQHivRXs0TaQrg1g/+JppwPqHnlVPmv7lessB8I5UqZeD6p6uVpHd7
Bs3pcPp8BV7BdRbs3sPLt5bV1+rkqOlw+0gZ4Q/ue/YbWtOAt4qY0OcEo0HgcnaX
lsB827QIfZIVtGWMhuh94xzm/SJkvngml6KB3YJNnWP61A9qJ37/VbVVLzvcmazA
McWB4HUMNrhd0JgBCo0gIpqCbpJEvUc02Bjn23eEJsS9kC7OUAHyQkVnx4d9UzXF
4OoISF6hmQKIBoLnRrAlj5Qvs3GhvHQ0ThYq0Grk/KMJJX2CSqt7tWJ8gk1n3H3Y
SReRXJRnv7DsDDBwFgT6r5Q2HW1TBUvaoZy5hF6maD09nHcNnvBjqADzeT8Tr/Qu
bBCLzkNSYqqkpgtwv7seoD2P4n1giRvDAOEfMZpVkUr+C252IaH1HZFEz+TvBVQM
Y8OWWxmIJW+J6evjo3N1eO19UHv71jvoF8zljbI4bsL2c+QTJmOv7nRqzDQgCWyp
Id/v2dUVVTk1j9omuLBBwNJzQCB+72LcIzJhYmaP1HC4LcKQG+/f41exuItenatK
lEJQhYtyVXcBlh6Yn/wzNg2NWOwb3vqY/F7m6u9ixAwgtIMgPCDE4aJ86zrrXYFz
N2HqkTSQh77Z8KPKmyGopsmN/reMuilPdINb249nA0dzoN+nj+tTFOYCIaLaFyjs
Z0r1QAOJAjkEEwECACMFAlq1SasCGwMHCwkIBwMCAQYVCAIJCgsEFgIDAQIeAQIX
gAAKCRC86dmkLVF4T9iFEACEnkm1dNXsWUx34R3c0vamHrPxvfkyI1FlEUen8D1h
uX9xy6jCEROHWEp0rjGK4QDPgM93sWJ+s1UAKg214QRVzft0y9/DdR+twApA0fzy
uavIthGd6+03jAAo6udYDE+cZC3P7XBbDiYEWk4XAF9I1JjB8hTZUgvXBL046JhG
eM17+crgUyQeetkiOQemLbsbXQ40Bd9V7zf7XJraFd8VrwNUwNb+9KFtgAsc9rk+
YIT/PEf+YOPysgcxI4sTWghtyCulVnuGoskgDv4v73PALU0ieUrvvQVqWMRvhVx1
0X90J7cC1KOyhlEQQ1aFTgmQjmXexVTwIBm8LvysFK6YXM41KjOrlz3+6xBIm/qe
bFyLUnf4WoiuOplAaJhK9pRY+XEnGNxdtN4D26Kd0F+PLkm3Tr3Hy3b1Ok34FlGr
KVHUq1TZD7cvMnnNKEELTUcKX+1mV3an16nmAg/my1JSUt6BNK2rJpY1s/kkSGSE
XQ4zuF2IGCpvBFhYAlt5Un5zwqkwwQR3/n2kwAoDzonJcehDw/C/cGos5D0aIU7I
K2X2aTD3+pA7Mx3IMe2hqmYqRt9X42yF1PIEVRneBRJ3HDezAgJrNh0GQWRQkhIx
gz6/cTR+ekr5TptVszS9few2GpI5bCgBKBisZIssT89aw7mAKWut0Gcm4qM9/yK6
1bkCDQRatUmrARAAxNPvVwreJ2yAiFcUpdRlVhsuOgnxvs1QgsIw3H7+Pacr9Hpe
8uftYZqdC82KeSKhpHq7c8gMTMucIINtH25x9BCc73E33EjCL9Lqov1TL7+QkgHe
T+JIhZwdD8Mx2K+LVVVu/aWkNrfMuNwyDUciSI4D5QHa8T+F8fgN4OTpwYjirzel
5yoICMr9hVcbzDNv/ozKCxjx+XKgnFc3wrnDfJfntfDAT7ecwbUTL+viQKJ646s+
psiqXRYtVvYInEhLVrJ0aV6zHFoigE/Bils6/g7ru1Q6CEHqEw++APs5CcE8VzJu
```

```
WAGSVHZgun5Y9N4quR/M9Vm+IPMhTxrAg7rOvyRN9cAXfeSMf77I+XTifigNna8x
t/MOdjXr1fjF4pThEi5u6WsuRdFwjY2azEv3vevodTi4HoJReH6dFRa6y8c+UDgl
2iHiOKIpQqLbHEfQmHcDd2fix+AaJKMnPGNku9qCFEMbgSRJpXz6BfwnY1QuKE+I
R6jA0frUNt2jhiGG/F8RceXzohaaC/Cx7LUCUFWc0n7z32C9/Dtj7I1PMOacdZzz
bjJzRKO/ZDv+UN/c9dwAkllzAyPMwGBkUaY68EBstnIliW34aWm6IiHhxioVPKSp
VJfyiXPO0EXqujtHLAeChfjcns3I12YshT1dv2PafG53fp33ZdzeUgsBo+EAEQEA
AYkCHwQYAQIACQUCWrVJqwIbDAAKCRC86dmkLVF4T+ZdD/9x/8APzgNJF3o3STrF
jvnV1ycyhWYGAeBJiu7wjsNWwzMFOv15tLjB7AqeVxZn+WKDD/mIOQ45OZvnYZuy
X7DR0JszaH9wrYTxZLVruAu+t6UL0y/XQ4L1GZ9QR6+r+7t1Mvbfy7BlHbvX/gYt
Rwe/uwdibI0CagEzyX+2D3kTOlHO5XThbXaNf8AN8zha91Jt2Q2UR2X5T6JcwtMz
FBvZnl3LSmZyE0EQehS2iUurU4uWOpGppuqVnbi0jbCvCHKgDGrqZ0smKNAQng54
F365W3g8AfY48s8XQwzmcliowYX9bT8PZiEi0J4QmQh0aXkpqZyFefuWeOL2R94S
XKzr+gRh3BAULoqF+qK+IUMxTip9KTPNvYDpiC66yBiT6gFDji5Ca9pGpJXrC3xe
TXiKQ8DBWDhBPVPrruLIaenTtZEOsPc4I85yt5U9RoPTStcOr34s3w5yEaJagt6S
Gc5r9ysjkfH6+6rbi1ujxMgROSqtqr+RyB+V9A5/OgtNZc8llK6u4UoOCde8jUUW
vqWKvjJB/Kz3u4zaeNu2ZyyHaOqOuH+TETcW+jsY9IhbEzqN5yQYGi4pVmDkY5vu
lXbJnbqPKpRXgM9BecV9AMbPgbDq/5LnHJJXg+G8YQOgp4lR/hC1TEFdIp5wM8AK
CWsENyt2o1rjgMXiZOMF8A5oBLkCDQRatUuSARAAr77kj7j2QR2SZeOSlFBvV7oS
mFeSNnz9xZssqrsm6bTwSHM6YLDwc7Sdf2esDdyzONETwqrVCg+FxgL8hmo9hS4c
rR6tmrP0mOmptr+xLLsKcaP7ogIXsyZnrEAEsvW8PnfayoiPCdc3cMCR/lTnHFGA
7EuR/XLBmi7Qg9tByVYQ5Yj5wB9V4B2yeCt3XtzPqeLKvaxl7PNelaHGJQY/xo+m
V0bndxf9IY+4oFJ4blD32WqvyxESo7vW6WBh7oqv3Zbm0yQrr8a6mDBpqLkvWwNI
3kpJR974tg5o5LfDu1BeeyHWPSGm4U/G4JB+JIG1ADy+RmoWEt4BqTCZ/knnoGvw
D5sTCxbKdmuOmhGyTssoG+3OOcGYHV7pWYPhazKHMPm201xKCjH1RfzRULzGKjD+
yMLT1I3AXFmLmZJXikAOlvE3/wgMqCXscbycbLjLD/bXIuFWo3rzoezeXjgi/DJx
jKBAyBTYO5nMcth1O9oaFd9d0HbsOUDkIMnsgGBE766Piro6MHo0T0rXl07Tp4pI
rwuSOsc6XzCzdImj0Wc6axS/HeUKRXWdXJwno5awTwXKRJMXGfhCvSvbcbc2Wx+L
IKvmB7EB4K3fmjFFE67yolmiw2qRcUBfygtH3eL5XZU28MiCpue8Y8GKJoBAUyvf
KeM1rO8Jm3iRAc5a/D0AEQEAAYkEPgQYAQIACQUCWrVLkgIbAgIpCRC86dmkLVF4
T8FdIAQZAQIABgUCWrVLkgAKCRDePL1hra+LjtHYD/9MucxdFe6bXO1dQR4tKhhQ
P0LRqy6zlBY9ILCLowNdGZdqorogUiUymgn3VhEhVtxTOoHcN7qOuM01PNsRnOeS
EYjf8Xrb1clzkD6xULwmOclTb9bBxnBc/4PFvHAbZW3QzusaZniNgkuxt6BTfloS
Of4inq71kjmGK+TlzQ6mUMQUg228NUQC+a84EPqYyAeY1sgvgB7hJBhYL0QAxhcW
6m20Rd8iEc6HyzJ3yCOCsKip/nRWAbf0OvfHfRBp0+m0ZwnJM8cPRFjOqqzFpKH9
HpDmTrC4wKP1+TL52LyEqNh4yZitXmZNV7giSRIkk0eDSko+bFy6VbMzKUMkUJK3
D3eHFAMkujmbfJmSMTJOPGn5SB1HyjCZNx6bhIIbQyEUB9gKCmUFaqXKwKpF6rj0
iQXAJxLR/shZ5Rk96VxzOphUl7T90m/PnUEEPwq8KsBhnMRgxa0RFidDP+n9fgtv
HLmrOqX9zBCVXh0mdWYLrWvmzQFWzG7AoE55fkf8nAEPsalrCdtaNUBHRXA0OQxG
AHMOdJQQvBsmqMvuAdjkDWpFu5y0My5ddU+hiUzUyQLjL5Hhd5LOUDdewlZgIw1j
xrEAUzDKetnemM8GkHxDgg8koev5frmShJuce7vSjKpCNg3EIJSgqMOPFjJuLWtZ
vjHeDNbJy6uNL65ckJy6WhGjEADS2WAW1D6Tfekkc21SsIXk/LqEpLMR/0g5OUif
wcEN1rS9IJXBwIy8MelN9qr5KcKQLmfdfBNEyyceBhyVl0MDyHOKC+7PofMtkGBq
13QieRHv5GJ8LB3fclqHV8pwTTo3Bc8z2g0TjmUYAN/ixETdReDoKavWJYSE9yoM
aaJu279ioVTrwpECse0XkiRyKToTjwOb73CGkBZZpJyqux/rmCV/fp4ALdSW8zbz
```

```
FJVORaivhoWwzjpfQKhwcU9lABXi2UvVm14v0AfeI7oiJPSU1zM4fEny4oiIBXlR
zhFNih1UjIu82X16mTm3BwbIga/s1fnQRGzyhqUIMii+mWra23EwjChaxpvjjcUH
5ilLc5Zq781aCYRygYQw+hu5nFkOH1R+Z50Ubxjd/aqUfnGIAX7kPMD3Lof4KldD
Q8ppQriUvxVo+4nPV6rpTy/PyqCLWDjkguHpJsEFsMkwajrAz0QNSAU5CJ0G2Zu4
yxvYlumHCEl7nbFrm0vIiA75Sa8KnywTDsyZsu3XcOcf3g+g1xWTpjJqy2bYXlqz
9uDOWtArWHOis6bq8l9RE6xr1RBVXS6uqgQIZFBGyq66b0dIq4D2JdsUvgEMaHbc
e7tBfeB1CMBdA64e9Rq7bFR7Tvt8gasCZYlNr3lydh+dFHIEkH53HzQe6l88HEic
+0jVnLkCDQRa55wJARAAyLya2Lx6gyoWoJN1a6740q3o8e9d4KggQOfGMTCflmeq
ivuzgN+3DZHN+9ty2KxXMtn0mhHBerZdbNJyjMNT1gAgrhPNB4HtXBXum2wS57WK
DNmade914L7FWTPAWBG2Wn448OEHTqsClICXXWy9IICgclAEyIq0Yq5mAdTEgRJS
Z8t4GpwtDL9gNQyFXaWQmDmkAsCygQMvhAlmu9xOIzQG5CxSnZFk7zcuL60k14Z3
Cmt49k4T/7ZU8goWi8tt+rU78/IL3J/fF9+1civ1OwuUidgfPCSvOUW1JojsdCQA
L+RZJcoXq7lfOFj/eNjeOSstCTDPfTCL+kThE6E5neDtbQHBYkEX1BRiTedsV4+M
ucgiTrdQFWKf89G72xdv8ut9AYYQ2BbEYU+JAYhUH8rYYui2dHKJIgjNvJscuUWb
+QEqJIRleJRhrO+/CHgMs4fZAkWF1VFhKBkcKmEjLn1f7EJJUUW84ZhKXjO/AUPX
1CHsNjziRceuJCJYox1cwsoq6jTE50GiNzcIxTn9xUc0UMKFeggNAFys1K+TDTm3
Bzo8H5ucjCUEmUm9lhkGwqTZgOlRX5eqPX+JBoSaObqhgqCa5IPinKRa6MgoFPHK
6sYKqroYwBGgZm6Js5chpNchvJMs/3WXNOEVg0J3z3vP0DMhxqWm+r+n9zlW8qsA
EQEAAYkEPgQYAQgACQUCWuecCQIbAgIpCRC86dmkLVF4T8FdIAQZAQgABgUCWuec
CQAKCRBQ3szEcQ5hr+ykD/4tOLRHFHXuKUcxgGaubUcVtsFrwBKma1cYjqaPms8u
6Sk0wfGRI32G/GhOrp0Ts/MOkbObq6VLTh8N5Yc/53MEl8zQFw9Y5AmRoW4PZXER
ujs5s7p4oR7xHMihMjCCBn1bvrR+34YPfgzTcgLiOEFHYT8UTxwnGmXOvNkMM7md
xD3CV5q6VAte8WKBo/220II3fcQlc9r/oWX4kXXkb0v9hoGwKbDJ1tzqTPrp/xFt
yohqnvImpnlz+Q9zXmbrWYL9/g8VCmW/NN2gju2G3Lu/TlFUWIT4v/5OPK6TdeNb
VKJO4+S8bTayqSG9CML1S57KSgCo5HUhQWeSNHI+fpe5oX6FALPT9JLDce8OZz1i
cZZ0MELP37mOOQun0AlmHm/hVzf0f311PtbzcqWaE51tJvgUR/nZFo6Ta3O5Ezhs
3VlEJNQ1Ijf/6DH87SxvAoRIARCuZd0qxBcDK0avpFzUtbJd24lRA3WJpkEiMqKv
RDVZkE4b6TW61f0o+LaVfK6E8oLpixegS4fiqC16mFrOdyRk+RJJfIUyz0WTDVmt
g0U1CO1ezokMSqkJ7724pyjr2xf/r9/sC6aOJwB/lKgZkJfC6NqL7TlxVA31dUga
LEOvEJTTE4gl+tYtfsCDvALCtqL0jduSkUo+RXcBItmXhA+tShW0pbS2Rtx/ixua
KohVD/0R4QxiSwQmICNtm9mw9ydIl1yjYXX5a9x4wMJracNY/LBybJPFnZnT4dYR
z4XjqysDwvvYZByaWoIe3QxjX84V6MlI2IdAT/xImu8gbaCI8tmyfpIrLnPKiR9D
VFYfGBXuAX7+HgPPSFtrHQONCALxxzlbNpS+zxt9r0MiLgcLyspWxSdmoYGZ6nQP
RO5Nm/ZVS+u2imPCRzNUZEMa+dlE6kHx0rS0dPiuJ4O7NtPeYDKkoQtNagspsDvh
cK7CSqAiKMq06UBTxqlTSRkm62eOCtcs3p3OeHu5GRZF1uzTET0ZxYkaPgdrQknx
ozjP5mC7X+45lcCfmcVt94TFNL5HwEUVJpmOgmzILCI8yoDTWzloo+i+fPFsXX4f
kynhE83mSEcr5VHFYrTY3mQXGmNJ3bCLuc/jq7ysGq69xiKmTlUeXFm+aojcRO5i
zyShIRJZ0GZfuzDYFDbMV9amA/YQGygLw//zP5ju5SW26dNxlf3MdFQE5JJ86rn9
MgZ4gcpazHEVUsbZsgkLizRp9imUiH8ymLqAXnfRGlU/LpNSefnvDFTtEIRcpOHc
bhayG0bk51Bd4mioOXnIsKy4j63nJXA27x5EVVHQ1sYRN8Ny4Fdr2tMAmj2O+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
```

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

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

f. Scarica le firme della CLI di Amazon ECS. 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

```
curl -Lo ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-darwin-amd64-latest.asc
```

Linux

```
curl -Lo ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-linux-amd64-latest.asc
```

Windows

```
Invoke-WebRequest -OutFile ecs-cli.asc https://amazon-ecs-cli.s3.amazonaws.com/ecs-cli-windows-amd64-latest.exe.asc
```

g. Verifica la firma.

macOS and Linux

```
gpg --verify ecs-cli.asc /usr/local/bin/ecs-cli
```

Windows

```
gpg --verify ecs-cli.asc 'C:\Program Files\Amazon\ECSCLI\ecs-cli.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.

3. Applica le autorizzazioni di esecuzione al file binario.

   macOS and Linux

   ```
   sudo chmod +x /usr/local/bin/ecs-cli
   ```

   Windows

   Modifica le variabili di ambiente e aggiungi `C:\Program Files\Amazon\ECSCLI` al campo della variabile `PATH`, separandolo dalle voci esistenti tramite un punto e virgola. Per esempio:

   ```
   setx path "%path%;C:\Program Files\Amazon\ECSCLI"
   ```

   Riavvia PowerShell in modo che le modifiche abbiano effetto.

> **Note**
>
> Dopo aver impostato la PATH variabile, la CLI di Amazon ECS può essere utilizzata da PowerShell Windows o dal prompt dei comandi.

4. Verifica il corretto funzionamento della CLI.

```
ecs-cli --version
```

Passa a Configurazione della CLI di Amazon ECS.

> **Important**
>
> Devi configurare l'Amazon ECS CLI con le AWS tue credenziali, AWS una regione e un nome di cluster Amazon ECS prima di poterla utilizzare. Per ulteriori informazioni, consulta Configurazione della CLI di Amazon ECS.

Configurazione della CLI di Amazon ECS

Amazon ECS ha rilasciato AWS Copilot, uno strumento di interfaccia a riga di comando (CLI) che semplifica la creazione, il rilascio e il funzionamento di applicazioni containerizzate pronte per la produzione su Amazon ECS da un ambiente di sviluppo locale. Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.

L'Amazon ECS CLI richiede alcune informazioni di configurazione di base prima di poterla utilizzare, come AWS le credenziali, AWS la regione in cui creare il cluster e il nome del cluster Amazon ECS da utilizzare. Le informazioni di configurazione vengono memorizzate nella directory `~/.ecs` sui sistemi macOS e Linux e in `C:\Users\<username>\AppData\local\ecs` sui sistemi Windows.

Come configurare la CLI di Amazon ECS

1. Imposta un profilo CLI con il seguente comando, sostituendolo con il nome del profilo desiderato *$AWS_ACCESS_KEY_ID* e le variabili di *$AWS_SECRET_ACCESS_KEY* ambiente *profile_name*con le tue credenziali. AWS

```
ecs-cli configure profile --profile-name profile_name --access-
key $AWS_ACCESS_KEY_ID --secret-key $AWS_SECRET_ACCESS_KEY
```

2. Completa la configurazione con il comando seguente, sostituendo *launch_type* con il tipo di avvio del processo che si desidera utilizzare come impostazione predefinita, *region_name* con la regione AWS desiderata, *cluster_name* con il nome di un cluster Amazon ECS esistente o nuovo da utilizzare e *configuration_name* per il nome che si desidera assegnare a questa configurazione.

```
ecs-cli configure --cluster cluster_name --default-launch-type launch_type --
region region_name --config-name configuration_name
```

Utilizzo dei profili

La CLI di Amazon ECS supporta la configurazione di più set AWS di credenziali come profili denominati utilizzando il comando. ecs-cli configure profile Puoi configurare un profilo predefinito tramite il comando ecs-cli configure profile default. Quando esegui comandi della CLI di Amazon ECS che richiedono le credenziali, puoi fare riferimento a tali profili tramite il flag `--ecs-profile` altrimenti, viene utilizzato il profilo di default.

Utilizzo delle configurazioni cluster

Una configurazione cluster è un insieme di campi che descrive un cluster Amazon ECS con valori quali il nome del cluster e la regione. Puoi impostare una configurazione cluster predefinita tramite il comando ecs-cli configure default. Tramite l'opzione `--config-name`, la CLI di Amazon ECS supporta la configurazione di più configurazioni cluster.

Informazioni sull'ordine di precedenza

Sono disponibili più modi per trasmettere credenziali e regione insieme in un comando della CLI di Amazon ECS. Di seguito è riportato l'ordine di precedenza per ciascuno di essi.

L'ordine di precedenza per le credenziali è:

1. Flag del profilo della CLI di Amazon ECS
   a. Profilo Amazon ECS (`--ecs-profile`)
   b. AWS profilo () `--aws-profile`

2. Variabili di ambiente:
   a. `ECS_PROFILE`
   b. `AWS_PROFILE`
   c. `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY` e `AWS_SESSION_TOKEN`
3. Tentativi della configurazione ECS di recuperare le credenziali dal profilo ECS predefinito.
4. AWS Profilo predefinito: tenta di utilizzare le credenziali (`aws_access_key_id`,`aws_secret_access_key`) o `assume_role` (`role_arn`,`source_profile`) dal nome del AWS profilo.
   a. Variabile di ambiente `AWS_DEFAULT_PROFILE` (con impostazione predefinita su `default`).
5. Ruolo dell'istanza EC2

L'ordine di precedenza per la regione è:

1. Flag della CLI di Amazon ECS:
   a. Contrassegno della regione (`--region`)
   b. Contrassegno di configurazione del cluster (`--cluster-config`)
2. Tentativi della configurazione ECS di recuperare la regione dal profilo ECS predefinito.
3. Variabili d'ambiente: i tentativi di recuperare la regione dalle seguenti variabili di ambiente:
   a. `AWS_REGION`
   b. `AWS_DEFAULT_REGION`
4. AWS profile - tenta di utilizzare la regione contenuta nel nome del AWS profilo:
   a. Variabile di ambiente `AWS_PROFILE`
   b. Variabile di ambiente `AWS_DEFAULT_PROFILE` (con impostazione predefinita su `default`)

AWS Fargateper Amazon ECS

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

Fargate offre versioni di piattaforma per le edizioni Amazon Linux 2 e Microsoft Windows 2019 Server Full e Core. Se non diversamente specificato, le informazioni contenute in questa pagina si applicano a tutte le piattaforme Fargate.

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

Per informazioni sulle regioni che supportano container Linux su Fargate, consulta the section called “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 utilizzare la console, consulta:

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

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

- Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI

- Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI

Provider di capacità

Sono disponibili i seguenti provider di capacità:

- Fargate
- Fargate Spot: esegui attività Amazon ECS tolleranti alle interruzioni a un prezzo scontato rispetto al prezzo. 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 il tipo di lancio Fargate .

  È possibile utilizzare Fargate Spot per Linux solo per attività che utilizzano l'architettura X86.

Definizioni di processi

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

Versioni della piattaforma

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

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

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

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

Inoltre, quando crei gruppi di destinazione per questi servizi, devi scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2. 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 le metriche di CloudWatch utilizzo per fornire visibilità sull'utilizzo delle risorse da parte dei tuoi account. Utilizza queste metriche per visualizzare l'utilizzo corrente del servizio su CloudWatch grafici e dashboard.

AWS Fargate le metriche di utilizzo corrispondono alle quote di servizio. AWS È possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sulle quote di servizio per AWS Fargate, consulta AWS Fargate quote di servizio.

Per ulteriori informazioni sui parametri di AWS Fargate utilizzo, consulta i parametri di AWS Fargateutilizzo nella Guida per l'utente di Amazon Elastic Container Service per. AWS Fargate

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

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

Le migliori pratiche di sicurezza di Fargate in Amazon ECS

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

Utilizzato AWS KMS per crittografare lo storage temporaneo per Fargate

Dovresti avere la tua memoria effimera crittografata da. AWS KMS Per le attività ospitate su Fargate utilizzando una versione della piattaforma `1.4.0` o successiva, ogni attività riceve 20 GiB di spazio di archiviazione temporaneo. La quantità totale di spazio di archiviazione temporaneo può essere aumentata, fino ad un massimo di 200 GiB, specificando il parametro `ephemeralStorage` nella definizione di attività. Per tali attività lanciate il 28 maggio 2020 o successivamente, l'archiviazione temporanea è crittografata con un algoritmo di crittografia AES-256 utilizzando una chiave di crittografia gestita da Fargate.

Per ulteriori informazioni, consulta Uso dei volumi di dati nelle attività.

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

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

```
aws ecs run-task --cluster clustername \
  --task-definition taskdefinition:version \
```

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

Funzionalità SYS_PTRACE per il tracciamento delle syscall del kernel con Fargate

La configurazione predefinita delle funzionalità di Linux che vengono aggiunte o rimosse dal container è fornita da Docker. Per ulteriori informazioni sulle funzionalità predefinite, consulta Privilegi del runtime e funzionalità di Linux nella documentazione di esecuzione di Docker.

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

Di seguito è riportato il video tutorial che 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 Runtime Monitoring nella Guida per l'utente. GuardDuty GuardDuty

Considerazioni sulla sicurezza di Fargate per Amazon ECS

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

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

L'ambiente di runtime in Fargate impedisce l'uso di determinate funzionalità del controller supportate sulle 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.

- Reti: puoi utilizzare gruppi di sicurezza e ACL di rete 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 Linux 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 AWS Domande frequenti sulla manutenzione delle attività di Fargate su Amazon ECS.

Considerazioni

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

- Quando specifichi una versione della piattaforma, puoi usare un numero di versione specifico, ad esempio `1.4.0`, o `LATEST`.

  Quando è selezionata l'ULTIMA versione della piattaforma, viene utilizzata la versione della piattaforma `1.4.0`.
- 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

della piattaforma Linux `1.4.0`, puoi aggiornare il servizio e specificare una nuova versione della piattaforma. Le attività vengono implementate nuovamente con la versione e la revisione della versione della piattaforma più recenti. Per ulteriori informazioni sulle implementazioni, consulta Servizi Amazon ECS.

- Se la capacità del servizio viene ridotta senza aggiornare la versione della piattaforma, le attività ricevono la versione della piattaforma specificata nell'implementazione corrente del servizio. Ad esempio, supponiamo che tu disponga di un servizio che esegue attività nella versione della piattaforma Linux `1.3.0`. Se aumenti il conteggio desiderato del servizio, il pianificatore del servizio avvia le nuove attività utilizzando la revisione più recente della versione della piattaforma `1.3.0`.
- Le nuove attività vengono sempre eseguite sulla revisione più recente della versione della piattaforma. Ciò garantisce che le attività vengano sempre avviate su un'infrastruttura protetta e con patch applicate.
- 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.

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

1.4.0

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

- A partire dal 5 novembre 2020, qualsiasi nuovo processo Amazon ECS avviato su Fargate tramite la versione della piattaforma `1.4.0` potrà utilizzare le seguenti funzioni:
  - Quando utilizzi Secrets Manager per archiviare dati sensibili, è possibile inserire una chiave JSON specifica o una versione specifica di un segreto come variabile di ambiente o in una configurazione di log. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.
  - Specifica le variabili di ambiente in blocco utilizzando il parametro di definizione del container `environmentFiles`. Per ulteriori informazioni, consulta Passa una singola variabile di ambiente a un contenitore Amazon ECS.

  - Ai processi eseguiti in un VPC e alla sottorete abilitata per IPv6 verrà assegnato sia un indirizzo IPv4 privato che un indirizzo IPv6. Per ulteriori informazioni, consulta Reti di processi Fargate nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate.
  - L'endpoint dei metadati dei processi versione 4 fornisce metadati aggiuntivi relativi al processo e al container, inclusi il tipo di avvio del processo, l'Amazon Resource Name (ARN) del container e il driver di log e le relative opzioni utilizzate. Quando si esegue una query sull'endpoint `/stats` si ricevono anche le statistiche sulla velocità di rete per i container. Per ulteriori informazioni, consulta Task metadata endpoint versione 4.
- A partire dal 30 luglio 2020, qualsiasi nuovo processo Amazon ECS avviato su Fargate tramite la versione della piattaforma `1.4.0` sarà in grado di instradare il traffico UDP utilizzando un Network Load Balancer ai suoi processi Amazon ECS su Fargate. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS.
- A partire dal 28 maggio 2020, qualsiasi nuova attività Amazon ECS lanciata su Fargate utilizzando la `1.4.0` versione della piattaforma avrà lo storage temporaneo crittografato con un algoritmo di crittografia AES-256 utilizzando una chiave di crittografia proprietaria. AWS Per ulteriori informazioni, consulta Archiviazione effimera delle attività Fargate per Amazon ECS e Opzioni di storage per le attività di Amazon ECS.
- Aggiunto il supporto per l'utilizzo dei volumi del file system Amazon EFS per l'archiviazione dei processi persistenti. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.
- Lo storage temporaneo delle attività è stato aumentato fino a un minimo di 20 GB per ogni attività. Per ulteriori informazioni, consulta Archiviazione effimera delle attività Fargate per Amazon ECS.
- Il comportamento del traffico di rete da e verso le attività è stato aggiornato. A partire dalla versione 1.4.0 della piattaforma, tutti i processi Fargate ricevono un'unica interfaccia di rete elastica (denominata ENI di processo) e tutto il traffico di rete scorre attraverso tale ENI all'interno del VPC e sarà visibile all'utente attraverso i log di flusso del VPC. Per ulteriori informazioni sul networking per il tipo di lancio di Amazon EC2, consulta Fargate Task Networking. Per ulteriori informazioni sulla rete per il tipo di lancio Fargate, vedere. Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate
- I task ENI aggiungono il supporto per i frame jumbo. Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimensione del payload più grande che si adatta all'interno di un singolo frame. Più grande è l'MTU, più il payload dell'applicazione può essere adattato all'interno di un singolo fotogramma, riducendo il sovraccarico per fotogramma e aumentando l'efficienza. Il supporto dei frame jumbo riduce il sovraccarico quando il percorso

di rete tra l'attività e la destinazione supporta frame jumbo, ad esempio tutto il traffico che rimane all'interno del VPC.

- CloudWatch Container Insights includerà metriche delle prestazioni di rete per le attività di Fargate. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.
- 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, vedere > Endpoint di metadati delle attività Amazon ECS versione 4 e. Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate
- Aggiunto il supporto per il parametro Linux `SYS_PTRACE` nelle definizioni dei container. Per ulteriori informazioni, consulta Parametri Linux.
- L'agente del container Fargate sostituisce l'uso dell'agente del container Amazon ECS per tutti i processi di Fargate. Spesso, questa modifica non dovrebbe avere effetto sull'esecuzione delle attività.
- Il runtime del container ora utilizza Containerd invece di Docker. Più frequentemente, questa modifica non dovrebbe avere effetto sull'esecuzione delle attività. Si noterà che alcuni messaggi di errore che hanno origine dal runtime del container passano dal menzionare Docker a errori più generici. Per ulteriori informazioni, consulta Codici di errore dei processi arrestati nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate.
- Basato su Amazon Linux 2.

1.3.0

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

- A partire dal 30 settembre 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log `awsfirelens`. Configura Amazon ECS FireLens per utilizzare i parametri di definizione delle attività per indirizzare i log verso un AWS servizio o una destinazione AWS Partner Network (APN) per l'archiviazione e l'analisi dei log. Per ulteriori informazioni, consulta Inviare i log di Amazon ECS a un servizio o AWSAWS Partner.
- Aggiunta la funzione di riavvio dei processi per i processi Fargate, che corrisponde al processo di aggiornamento dei processi che sono parte di un servizio Amazon ECS. Per ulteriori informazioni, consulta Manutenzione dei processi nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate.

- A partire dal 27 marzo 2019, ogni nuovo processo Fargate che viene avviato può utilizzare ulteriori parametri di definizione dei processi che consentono di definire una configurazione proxy, le dipendenze per l'avvio e l'arresto di container, nonché un valore di timeout di avvio e arresto per container. Per ulteriori informazioni, consulta Configurazione del proxy, Dipendenze per i container e Timeout del container.
- A partire dal 2 aprile 2019, qualsiasi nuova attività di Fargate che verrà lanciata supporta l'iniezione di dati sensibili nei contenitori archiviando i dati sensibili in AWS Secrets Manager segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.
- A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta il riferimento ai dati sensibili nella configurazione dei log di un container utilizzando il parametro di definizione del container `secretOptions`. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.
- A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log `splunk` 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 contenitori Amazon ECS utilizzando Container Insights.
- A partire dal 3 dicembre 2019, è supportato il Capacity Provider Fargate Spot. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio Fargate .
- Basato su Amazon Linux 2.

Migrazione alla versione 1.4.0 della piattaforma Linux

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. Una best practice consiste nel verificare che il processo funzioni correttamente sulla versione della piattaforma `1.4.0` prima di eseguire la migrazione dei processi.

- 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 Amazon ECS su Fargate ricevono un'unica interfaccia di rete elastica (denominata ENI attività) e tutto il traffico di rete scorre attraverso tale ENI all'interno

del VPC e sarà visibile all'utente attraverso i log di flusso VPC. Per ulteriori informazioni, consulta Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate.

- Se si utilizzano endpoint VPC dell'interfaccia, è necessario considerare quanto segue.
  - Quando si utilizzano immagini del container ospitate con Amazon ECR, insieme all'endpoint gateway Amazon S3 sono richiesti entrambi gli endpoint VPC di Amazon ECR com.amazonaws.region.ecr.dkr e com.amazonaws.region.ecr.api. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Se utilizzi una definizione di attività che 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 AWS Secrets Manager .
  - 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 Utilizzo di Systems Manager con endpoint VPC nella Guida per l'utente di AWS Systems Manager .
  - Assicurarsi che il gruppo di sicurezza nell'interfaccia di rete elastica (ENI) associato all'attività disponga delle regole del gruppo di sicurezza create per consentire il traffico tra l'attività e gli endpoint VPC in uso.

AWS Deprecazione della versione della piattaforma Fargate Linux

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

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

Consigliamo di aggiornare i processi autonomi dei servizi in modo da utilizzare la versione più recente della piattaforma. Per ulteriori informazioni sulla migrazione alla versione più recente della piattaforma, consulta Migrazione alla versione 1.4.0 della piattaforma Linux.

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

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

Per informazioni sulle versioni correnti della piattaforma, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.

Changelog per le versioni obsolete di Fargate Linux AWS

1.2.0

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

> **Note**
>
> La versione della piattaforma `1.2.0` non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta AWS Deprecazione della versione della piattaforma Fargate Linux.

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

1.1.0

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

> **Note**
>
> La versione della piattaforma `1.1.0` non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta AWS Deprecazione della versione della piattaforma Fargate Linux.

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

1.0.0

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

> **Note**
>
> La versione della piattaforma `1.0.0` non è più disponibile. Per informazioni sulla dichiarazione delle versioni della piattaforma come obsolete, consulta AWS Deprecazione della versione della piattaforma Fargate Linux.

- Basato su Amazon Linux 2017.09
- Versione iniziale.

Comportamento dell'immagine dei contenitori Linux su Fargate Container Pull per Amazon ECS

Ogni attività Fargate viene eseguita su una propria istanza monouso con tenant singolo. Quando si eseguono contenitori Linux su Fargate, le immagini dei contenitori o i livelli di immagini dei contenitori

non vengono memorizzati nella cache dell'istanza. Pertanto, per ogni immagine del contenitore definita nell'operazione, l'intera immagine del contenitore deve essere estratta dal registro delle immagini del contenitore per ogni attività Fargate. Il tempo necessario per estrarre le immagini è direttamente correlato al tempo impiegato per avviare un'attività Fargate.

Tenete conto di quanto segue per ottimizzare il tempo di caricamento dell'immagine.

Prossimità dell'immagine del contenitore

Per ridurre il tempo necessario per scaricare le immagini dei container, posiziona i dati il più vicino possibile al calcolo. L'estrazione dell'immagine di un contenitore da Internet o dall'altra parte Regioni AWS potrebbe influire sui tempi di download. Ti consigliamo di archiviare l'immagine del contenitore nella stessa regione in cui verrà eseguita l'attività. Se memorizzi l'immagine del contenitore in Amazon ECR, utilizza un endpoint di interfaccia VPC per ridurre ulteriormente il tempo di recupero dell'immagine. Per ulteriori informazioni, consulta gli endpoint VPC dell'interfaccia Amazon ECR AWS PrivateLink() nella Amazon ECR User Guide.

Riduzione delle dimensioni dell'immagine del contenitore

Le dimensioni di un'immagine del contenitore influiscono direttamente sul tempo di download. La riduzione delle dimensioni dell'immagine del contenitore o del numero di livelli dell'immagine del contenitore può ridurre il tempo necessario per il download di un'immagine. Le immagini di base leggere (come l'immagine minima del contenitore Amazon Linux 2023) possono essere significativamente più piccole di quelle basate sulle immagini di base del sistema operativo tradizionale. Per ulteriori informazioni sull'immagine minima, consulta l'immagine minima del contenitore AL2023 nella Amazon Linux 2023 User Guide.

Algoritmi di compressione alternativi

I livelli di immagini dei contenitori vengono spesso compressi quando vengono inseriti in un registro di immagini dei contenitori. La compressione del livello di immagine del contenitore riduce la quantità di dati che devono essere trasferiti attraverso la rete e archiviati nel registro delle immagini del contenitore. Dopo che un livello di immagine del contenitore è stato scaricato su un'istanza dal runtime del contenitore, tale livello viene decompresso. L'algoritmo di compressione utilizzato e la quantità di vCPU disponibili 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 ztsd on. GitHub Per informazioni su come implementare le immagini per Fargate, vedere Riduzione dei tempi di AWS Fargate avvio con le immagini dei container compressi zstd.

Lazy Loading delle immagini dei contenitori

Per immagini di contenitori di grandi dimensioni (> 250 MB), potrebbe essere ottimale caricare lentamente un'immagine del contenitore anziché scaricarla tutta. Su Fargate, puoi utilizzare Seekable OCI (SOCI) per caricare lentamente un'immagine del contenitore da un registro delle immagini del contenitore. Per ulteriori informazioni, consulta soci-snapshotter on GitHub e Lazy loading container images using Seekable OCI (SOCI).

Versioni della piattaforma Fargate Windows 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 AWS Domande frequenti sulla manutenzione delle attività di Fargate su Amazon ECS.

Considerazioni relative alla versione della piattaforma

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

- Quando specifichi una versione della piattaforma, puoi usare un numero di versione specifico, ad esempio `1.0.0`, o `LATEST`.

  Quando è selezionata l'ULTIMA versione della piattaforma, viene utilizzata la piattaforma `1.0.0`.

- Le nuove attività vengono sempre eseguite sulla revisione più recente della versione della piattaforma. Ciò garantisce che le attività vengano sempre avviate su un'infrastruttura protetta e con patch applicate.
- 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.
- 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.

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

1.0.0

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

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

Contenitori Windows su Fargate: considerazioni per Amazon ECS

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

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

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

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

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

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

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

- Account del servizio gestito del gruppo (gMSA)
- Amazon FSx
- Trunking ENI
- Servizio App Mesh e integrazione proxy per i processi
- Integrazione del router di log Firelens per i processi
- Volumi EFS
- I parametri di definizione delle attività riportati di seguito:
  - `maxSwap`
  - `swappiness`
  - `environmentFiles`
- Il provider di capacità Fargate Spot
- Volumi di immagine

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

Contenitori Windows su Fargate Container Image Pull: comportamento per Amazon ECS

Fargate Windows memorizza nella cache l'immagine di base servercore del mese più recente e quella del mese precedente fornita da Microsoft. Queste immagini corrispondono alle patch con numero KB/Build aggiornate ogni Patch Tuesday. Ad esempio, il 4/09/2024 Microsoft ha rilasciato KB5036896 (17763.5696) per Windows Server 2019. Il mese precedente il KB del 03/12/2024 era KB5035849 (17763.5576). Quindi, per le piattaforme e i seguenti contenitori, le immagini sono state memorizzate nella cache: `WINDOWS_SERVER_2019_CORE` `WINDOWS_SERVER_2019_FULL`

- `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 04/09/2024 Microsoft ha rilasciato KB5036909 (20348.2402) per Windows Server 2022. Il mese precedente il KB del 12/3/2024 era KB5035857 (20348.2340). Quindi, per le piattaforme e i seguenti contenitori, le immagini sono state memorizzate nella cache: `WINDOWS_SERVER_2022_CORE` `WINDOWS_SERVER_2022_FULL`

- `mcr.microsoft.com/windows/servercore:ltsc2022`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2402`
- `mcr.microsoft.com/windows/servercore:10.0.20348.2340`

Archiviazione effimera delle attività Fargate per Amazon ECS

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

Versioni della piattaforma container Linux Fargate

Versione 1.4.0 o successiva

Per impostazione predefinita, le attività Amazon ECS ospitate su Fargate che utilizzano la piattaforma versione `1.4.0` o successiva ricevono almeno 20 GiB di storage temporaneo. La quantità totale di

storage temporaneo può essere aumentata, fino a un massimo di 200 GiB. Puoi eseguire questa operazione specificando il parametro `ephemeralStorage` nella definizione di attività.

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

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

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

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

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

Versione 1.3.0 o precedente

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

- 10 GB di storage di livello Docker

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

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

Versioni della piattaforma container Windows Fargate

Versione 1.0.0 o successiva

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

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

Per ulteriori informazioni, consulta Usa i bind mount con Amazon ECS.

Chiavi gestite dal cliente per AWS Fargate lo storage temporaneo

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

È possibile importare o creare chiavi personalizzate in. AWS KMS AWS KMS Queste chiavi autogestite vengono archiviate AWS KMS ed eseguono azioni standard AWS KMS del ciclo di vita come la rotazione, la disabilitazione e l'eliminazione. È possibile controllare l'accesso e l'utilizzo delle chiavi nei log. CloudTrail

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

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

Argomenti

- Crea una chiave di crittografia per l'archiviazione effimera Fargate
- Gestione delle AWS KMS chiavi per l'archiviazione effimera Fargate

Crea una chiave di crittografia per l'archiviazione effimera Fargate

> **Note**
>
> La crittografia dello storage temporaneo Fargate con chiavi gestite dal cliente non è disponibile per i cluster di attività di Windows.
> La crittografia dello storage temporaneo Fargate con chiavi gestite dal cliente non era disponibile prima di. `platformVersions` `1.4.0`
> Fargate riserva spazio su uno spazio di archiviazione temporaneo utilizzato solo da Fargate e non ti viene addebitato alcun costo per lo spazio. L'allocazione potrebbe differire dalle attività chiave non gestite dal cliente, ma lo spazio totale rimane lo stesso. È possibile visualizzare questa modifica in strumenti come. `df`

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

1. Accedere a https://console.aws.amazon.com/kms.
2. Segui le istruzioni per la creazione di chiavi nella Guida per AWS Key Management Service gli sviluppatori.
3. Durante la creazione della AWS KMS chiave, assicuratevi di fornire al servizio Fargate le autorizzazioni AWS KMS operative pertinenti nelle politiche chiave. Le seguenti operazioni API devono essere consentite nella policy per utilizzare la chiave gestita dal cliente con le risorse del cluster Amazon ECS.

- `kms:GenerateDataKeyWithoutPlainText`- Chiama `GenerateDataKeyWithoutPlainText` per generare una chiave dati crittografata dalla AWS KMS chiave fornita.
- `kms:CreateGrant`- Aggiunge una concessione a una chiave gestita dal cliente. Concede l'accesso di controllo a una AWS KMS chiave specificata, che consente l'accesso alle operazioni di concessione richieste da Amazon ECS Fargate. Per ulteriori informazioni sull'utilizzo di Grants, consulta la Guida per gli sviluppatori.AWS Key Management Service Ciò consente ad Amazon ECS Fargate di effettuare le seguenti operazioni:
  - Chiama `Decrypt` per AWS KMS ottenere la chiave di crittografia per decrittografare i dati di archiviazione temporanei.
  - Configura un preside in pensione per consentire al servizio di farlo. `RetireGrant`
- `kms:DescribeKey`- Fornisce i dettagli chiave gestiti dal cliente per consentire ad Amazon ECS di convalidare la chiave se è simmetrica e abilitata.

L'esempio seguente mostra una politica AWS KMS chiave da applicare alla chiave di destinazione per la crittografia. Per utilizzare le istruzioni politiche di esempio, sostituite i *segnaposto di input dell'utente* con le vostre informazioni. Come sempre, configura solo le autorizzazioni di cui hai bisogno.

```
{
      "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/
ClusterOperatorRole" },
      "Action": [
        "kms:DescribeKey"
      ],
      "Resource": "*"
    }
```

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

Per ulteriori informazioni, consultare Contesto della crittografia nella Guida per gli sviluppatori di AWS KMS.

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

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:UpdateCluster"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "ecs:fargate-ephemeral-storage-kms-key": "arn:aws:kms:us-
west-2:111122223333:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
      }
    }
  ]
}
```

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

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
```

```
        "ecs:CreateCluster",
        "ecs:UpdateCluster"
    ],
    "Resource": "*",
    "Condition": {
      "Null": {
        "ecs:fargate-ephemeral-storage-kms-key": "true"
      }
    }
  }
}
```

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

Per ulteriori informazioni, consulta Condition keys AWS KMS nella AWS KMS Developer Guide.

AWS Management Console

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

AWS CLI

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

```
aws ecs create-cluster --cluster clusterName \
--configuration '{"managedStorageConfiguration":
{"fargateEphemeralStorageKmsKeyId":"arn:aws:kms:us-
west-2:012345678901:key/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}}'
```

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

AWS CloudFormation

Di seguito è riportato un modello di esempio per la creazione di un cluster e la configurazione dello storage temporaneo Fargate utilizzando (sostituisci AWS CloudFormation i valori rossi 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 l'archiviazione effimera Fargate

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

AWS KMS Rotazione automatica dei tasti

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

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

Disattivazione o AWS KMS revoca delle chiavi

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

Eliminazione delle chiavi AWS KMS

L'eliminazione delle chiavi dovrebbe sempre essere l'ultima risorsa e dovrebbe essere eseguita solo se si è certi che la chiave eliminata non sia mai più necessaria. Le nuove attività che tentano di utilizzare la chiave eliminata falliranno perché non possono accedervi. AWS KMS consiglia di disabilitare una chiave anziché eliminarla. Se ritieni necessario eliminare una chiave, ti consigliamo 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 `GenerateDataKeyWithoutPlaintextGenerateDataKeyWithoutPlaintext (DryRun)`,`CreateGrant`,`CreateGrant (DryRun)`, e `RetireGrant` (sostituisci i valori *rossi* 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-EXAMPLEaaaaa",
    "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-EXAMPLEaaaaa",
    "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":
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855",
        "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-EXAMPLEaaaaa",
    "eventCategory": "Management"
}
```

CreateGrant (DryRun)

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

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

RetireGrant

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

```
        "keyId": "arn:aws:kms:us-west-2:account-id:key/a1b2c3d4-5678-90ab-cdef-
EXAMPLE11111"
    },
    "additionalEventData": {
        "grantId":
 "e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855"
    },
    "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"
}
```


AWS Domande frequenti sulla manutenzione delle attività di Fargate su Amazon ECS

Che cos'è la manutenzione e il pensionamento delle attività di Fargate?

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

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

Le attività di Amazon ECS possono essere classificate come attività di servizio o come attività autonome. Le attività di servizio vengono distribuite come parte di un servizio e controllate dalla pianificazione di Amazon ECS. Per ulteriori informazioni, consulta Servizi Amazon ECS. Le attività autonome sono attività avviate dall'`RunTask`API Amazon ECS, direttamente o da uno scheduler esterno, come le attività pianificate (che vengono avviate da Amazon EventBridge), AWS Batch oppure. AWS Step Functions

Per le attività di servizio, non è necessario intraprendere alcuna azione a meno che non si desideri sostituirle prima. AWS Quando lo scheduler di Amazon ECS interrompe le attività, utilizza la percentuale minima di integrità e avvia una nuova attività nel tentativo di mantenere il numero desiderato per il servizio. Per impostazione predefinita, la percentuale minima di attività integre di un servizio è pari al 100%, quindi prima di arrestare un'attività ne viene avviata un'altra. Le attività di servizio vengono regolarmente sostituite allo stesso modo quando si ridimensiona il servizio, si implementano modifiche alla configurazione o si distribuiscono revisioni delle definizioni delle attività. Per preparare il processo di ritiro, consigliamo di testare il comportamento dell'applicazione simulando questo scenario. A tale scopo, è possibile interrompere una singola attività nel servizio per testare la resilienza.

Per il ritiro delle attività autonome, AWS interrompe l'attività alla data di ritiro dell'attività o dopo tale data. Non lanciamo un'attività sostitutiva quando un'attività viene interrotta. Se è necessario che queste attività continuino a essere eseguite, è necessario interrompere le attività in esecuzione e avviare un'attività sostitutiva prima del tempo indicato nella notifica. Pertanto, consigliamo ai clienti di monitorare lo stato delle attività autonome e, se necessario, di implementare una logica per sostituire quelle interrotte.

Quando un'attività viene interrotta in uno degli scenari seguenti, puoi eseguire il comando `describe-tasks`. Il `stoppedReason` nella risposta è `ECS is performing maintenance on the underlying infrastructure hosting the task`.

La manutenzione delle attività si applica quando è presente una nuova versione della piattaforma, la revisione deve essere sostituita con una nuova revisione. In caso di problemi con un host Fargate sottostante, Amazon ECS sostituisce l'host senza un avviso di ritiro dell'attività.

Cosa contiene l'avviso di ritiro dell'attività?

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, gli ID delle attività.
- Per le attività del servizio, l'ID del cluster in cui viene eseguito il servizio e gli ID del servizio.
- I passaggi successivi da eseguire.

In genere, inviamo una notifica ciascuna per le attività di servizio e le attività autonome. Regione AWS Tuttavia, in alcuni casi potresti ricevere più di un evento per ogni tipo di attività, ad esempio quando ci sono troppe attività per essere ritirate e superano i limiti previsti dai nostri meccanismi di notifica.

Puoi identificare le attività pianificate per il ritiro nei modi seguenti:

- Il AWS Health Dashboard

  AWS Health le notifiche possono essere inviate tramite Amazon EventBridge a sistemi di archiviazione come Amazon Simple Storage Service, eseguire azioni automatiche come eseguire una AWS Lambda funzione o altri sistemi di notifica come Amazon Simple Notification Service. Per ulteriori informazioni, consulta Monitoraggio AWS Health degli eventi con Amazon EventBridge. Per una configurazione di esempio per inviare notifiche ad Amazon Chime, Slack o Microsoft Teams, consulta l'archivio AWS Health Aware su. GitHub

  Di seguito è riportato un esempio di evento. EventBridge

```
{
    "version": "0",
    "id": "3c268027-f43c-0171-7425-1d799EXAMPLE",
    "detail-type": "AWS Health Event",
    "source": "aws.health",
    "account": "123456789012",
    "time": "2023-08-16T23:18:51Z",
    "region": "us-east-1",
    "resources": [
        "cluster/service",
        "cluster/service"
```

```
    ],
    "detail": {
        "eventArn": "arn:aws:health:us-east-1::event/ECS/
AWS_ECS_TASK_PATCHING_RETIREMENT/AWS_ECS_TASK_PATCHING_RETIREMENT_test1",
        "service": "ECS",
        "eventScopeCode": "ACCOUNT_SPECIFIC",
        "communicationId":
 "7988399e2e6fb0b905ddc88e0e2de1fd17e4c9fa60349577446d95a18EXAMPLE",
        "lastUpdatedTime": "Wed, 16 Aug 2023 23:18:52 GMT",
        "eventRegion": "us-east-1",
        "eventTypeCode": "AWS_ECS_TASK_PATCHING_RETIREMENT",
        "eventTypeCategory": "scheduledChange",
        "startTime": "Wed, 16 Aug 2023 23:18:51 GMT",
        "endTime": "Fri, 18 Aug 2023 23:18:51 GMT",
        "eventDescription": [
            {
                "language": "en_US",
                "latestDescription": "\\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\\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:\\nPlease
 refer to create-deployment document (2) and create new deployment using same task
 definition revision.\\n\\nFor further details on ECS deployment types, please
 refer to ECS Deployment Developer Guide (1).\\nFor further details on Fargate's
 update process, please refer to the AWS Fargate User Guide (3).\\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\\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"
            }
        ],
        "affectedEntities": [
            {
                "entityValue": "cluster/service"
            },
            {
                "entityValue": "cluster/service"
            }
        ]
    }
}
```

- E-mail

  Viene inviata un'e-mail all'indirizzo di posta elettronica registrato per l' Account AWS ID.

Posso modificare il tempo di attesa per il ritiro dell'attività?

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (`0`). Quando hai bisogno di un maggiore controllo, ad esempio, quando un'attività può essere interrotta solo durante una determinata finestra, configura l'opzione 7 (`7`) o 14 giorni (`14`).

Consigliamo di scegliere un periodo di attesa più breve per ricevere prima le revisioni delle versioni più recenti della piattaforma.

Configura il periodo di attesa eseguendo `put-account-setting-default` o `put-account-setting` come utente root o utente amministrativo. Utilizza l'opzione `fargateTaskRetirementWaitPeriod` per il `name` e l'opzione `value` impostata su uno dei seguenti valori:

- `0`- AWS invia la notifica e inizia immediatamente a ritirare le attività interessate.
- `7`- AWS invia la notifica e attende 7 giorni di calendario prima di iniziare a ritirare le attività interessate.

- 14: AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

L'impostazione di default è 7 giorni.

Per ulteriori informazioni, consulta put-account-setting-default e put-account-setting nella Documentazione di riferimento delle API di Amazon Elastic Container Service.

Per ulteriori informazioni, consulta AWS Fargate tempo di attesa per il ritiro dell'attività.

Posso ricevere notifiche di ritiro delle attività tramite altri servizi? AWS

AWS invia una notifica di ritiro dell'attività al AWS Health Dashboard e al contatto di posta elettronica principale su. Account AWS AWS Health Dashboard Fornisce una serie di integrazioni in altri AWS servizi, tra cui. EventBridge È possibile EventBridge utilizzarlo per automatizzare la visibilità degli avvisi (ad esempio inoltrando il messaggio a uno strumento). ChatOps Per ulteriori informazioni, vedere Panoramica della soluzione: acquisizione delle notifiche di ritiro delle attività.

Posso modificare il ritiro di un'attività dopo averla pianificata?

No. La pianificazione si basa sul tempo di attesa per il ritiro dell'attività, che ha un valore predefinito di 7 giorni. Se hai bisogno di più tempo, puoi scegliere di configurare il periodo di attesa su 14 giorni. Per ulteriori informazioni, consulta Posso modificare il tempo di attesa per il ritiro dell'attività?. La modifica di questa configurazione si applica ai pensionamenti che verranno programmati in futuro. I pensionamenti attualmente programmati non sono influenzati. Se hai ulteriori dubbi, contatta. AWS Support

Posso controllare i tempi di sostituzione di un'attività?

Per i servizi che utilizzano la distribuzione continua, è necessario aggiornare il servizio utilizzando `update-service` l'`force-deployment`opzione prima dell'inizio del pensionamento.

L'`update-service`esempio seguente utilizza l'`force-deployment`opzione.

```
aws ecs update-service —-service service_name \
    --cluster cluster_name \
     --force-new-deployment
```

Per i servizi che utilizzano la distribuzione blu/verde, è necessario creare una nuova distribuzione in. AWS CodeDeployPer informazioni su come creare la distribuzione, consulta create-deployment nella Guida di riferimento.AWS Command Line Interface

In che modo Amazon ECS gestisce le attività che fanno parte di un servizio?

Amazon ECS sostituisce gradualmente le attività interessate del tuo servizio all'inizio del periodo di pensionamento di Fargate. Quando Amazon ECS interrompe un'attività, utilizza la percentuale minima di integrità del servizio e avvia una nuova attività per mantenere il numero di attività desiderato per il servizio. Una nuova attività viene avviata prima che un'attività venga interrotta perché la percentuale di integrità minima predefinita è 100. Le attività di servizio vengono regolarmente sostituite allo stesso modo quando si ridimensiona il servizio, si implementano modifiche alla configurazione o si distribuiscono revisioni delle definizioni delle attività. Per ulteriori informazioni sulla percentuale minima di salute, consulta. Configurazione dell'implementazione

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.

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. Se applicabile, vengono annotati gli ID zona di disponibilità supportati.

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 `usw1-az1` e `usw1-az3`)

Nome della regione	Regione
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacifico (Tokyo)	ap-northeast-1 (solo `apne1-az1` , `apne1-az2` e `apne1-az4`)
Asia Pacifico (Seul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (solo `cnn1-az1` e `cnn1-az2`)
Cina (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2

Nome della regione	Regione
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2
Europa (Stoccolma)	eu-north-1
South America (São Paulo)	sa-east-1
Israele (Tel Aviv)	il-central-1
Medio Oriente (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. Se applicabile, vengono annotati gli ID zona di disponibilità supportati.

Nome della regione	Regione
US East (Ohio)	us-east-2
Stati Uniti orientali (Virginia settentrionale)	us-east-1 (solo `use1-az1`, `use1-az2`, `use1-az4`, `use1-az5` e `use1-az6`)
Stati Uniti occidentali (California settentrionale)	us-west-1 (solo `usw1-az1` e `usw1-az3`)
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1

Nome della regione	Regione
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacifico (Tokyo)	ap-northeast-1 (solo `apne1-az1` , `apne1-az2` e `apne1-az4`)
Canada (Centrale)	ca-central-1 (solo `cac1-az1` e `cac1-az2`)
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (solo `cnn1-az1` e `cnn1-az2`)
Cina (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2

Nome della regione	Regione
Europa (Stoccolma)	eu-north-1
South America (São Paulo)	sa-east-1
Israele (Tel Aviv)	il-central-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Medio Oriente (Bahrein)	me-south-1

Progetta la tua soluzione per Amazon ECS

Prima di utilizzare Amazon ECS, devi prendere decisioni su capacità, rete, impostazioni dell'account e registrazione in modo da poter configurare correttamente le tue risorse Amazon ECS.

Capacità

La capacità è l'infrastruttura in cui vengono eseguiti i contenitori. Le opzioni sono le seguenti:

- Istanze Amazon EC2
- Senza server ()AWS Fargate (Fargate)
- Macchine virtuali (VM) o server on-premise

L'infrastruttura viene specificata quando si crea un cluster. È inoltre possibile specificare il tipo di infrastruttura quando si registra una definizione di attività. La definizione dell'attività si riferisce all'infrastruttura come al «tipo di avvio». Il tipo di avvio viene utilizzato anche quando si esegue un'attività autonoma o si distribuisce un servizio. Per informazioni sulle opzioni del tipo di avvio, consulta. Tipi di avvio di Amazon ECS

Rete

AWS le risorse vengono create in sottoreti. Quando utilizzi istanze EC2, Amazon ECS avvia le istanze nella sottorete specificata al momento della creazione di un cluster. Le tue attività vengono eseguite nella sottorete dell'istanza. Per le macchine virtuali Fargate o locali, si specifica la sottorete quando si esegue un'attività o si crea un servizio.

A seconda dell'applicazione, la sottorete può essere una sottorete privata o pubblica e può trovarsi in una delle seguenti risorse: AWS

- Zone di disponibilità
- Zone locali
- Zone Wavelength
- Regioni AWS
- AWS Outposts

Per ulteriori informazioni, consulta Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones o Amazon Elastic Container Service su AWS Outposts.

È possibile connettere l'applicazione a Internet utilizzando uno dei seguenti metodi:

- Una sottorete pubblica con un gateway Internet

  Utilizza le sottoreti pubbliche quando hai applicazioni pubbliche che richiedono grandi quantità di larghezza di banda o una latenza minima. Gli scenari applicabili includono lo streaming video e i servizi di gioco.

- Una sottorete privata con un gateway NAT

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

Accesso alle funzionalità

Puoi utilizzare le impostazioni del tuo account Amazon ECS per accedere alle seguenti funzionalità:

- Container Insights

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

- `awsvpctrunking`

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

- Autorizzazione all'assegnazione di tag

  Gli utenti devono disporre delle autorizzazioni per le azioni che creano una risorsa, ad esempio. `ecsCreateCluster` Se i tag sono specificati nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva sull'azione per verificare se gli utenti o i ruoli dispongono delle `ecs:TagResource` autorizzazioni per creare tag.

- Conformità FIPS-140 di Fargate

Fargate supporta il Federal Information Processing Standard (FIPS-140) che specifica i requisiti di sicurezza previsti per i moduli crittografici che proteggono le informazioni sensibili. È l'attuale standard governativo degli Stati Uniti e del Canada ed è applicabile ai sistemi che devono essere conformi al Federal Information Security Management Act (FISMA) o al Federal Risk and Authorization Management Program (FedRAMP).

- Modifiche all'orario di pensionamento delle attività di Fargate

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

- VPC a doppio stack

  Consenti alle attività di comunicare tramite IPv4, IPv6 o entrambi.

- Formato Amazon Resource Name (ARN)

  Alcune funzionalità, come l'autorizzazione all'etichettatura, richiedono un nuovo formato Amazon Resource Name (ARN).

Per ulteriori informazioni, consulta Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account.

Ruoli IAM

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

Alcune funzionalità di Amazon ECS richiedono ruoli. Per ulteriori informazioni, consulta Ruoli IAM per Amazon ECS.

Registrazione

La registrazione e il monitoraggio sono aspetti importanti per mantenere l'affidabilità, la disponibilità e le prestazioni dei carichi di lavoro Amazon ECS. Sono disponibili le seguenti opzioni:

- Amazon CloudWatch log: indirizza i log verso Amazon CloudWatch
- FireLensper 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.

Tipi di avvio di Amazon ECS

Il tipo di avvio della definizione dell'attività definisce, ad esempio, la capacità con cui l'attività può essere eseguita AWS Fargate.

Dopo aver scelto il tipo di avvio, Amazon ECS verifica che i parametri di definizione delle attività configurati funzionino con il tipo di avvio.

Fargate

Fargate è un motore di pay-as-you-go elaborazione senza server che ti consente di concentrarti sulla creazione di applicazioni senza gestire i server. Scegliendo Fargate, non è necessario gestire un'infrastruttura EC2. Tutto ciò che devi fare è creare l'immagine del contenitore e definire su quale cluster vuoi eseguire le tue applicazioni. Fargate ha un'integrazione nativa con AWS servizi tra cui:

- Amazon VPC
- Auto Scaling
- Sistema di bilanciamento del carico elastico
- IAM
- Secrets Manager

Con Fargate hai un maggiore controllo rispetto a EC2 perché selezioni esattamente la CPU e la memoria di cui la tua applicazione ha bisogno. Fargate gestisce la scalabilità della tua capacità, quindi non devi preoccuparti dei picchi di traffico. Ciò significa che l'impegno operativo con Fargate è minore.

Fargate soddisfa gli standard per i programmi di conformità tra cui PCI, FIPS 140-2, FedRAMP e HIPAA. Per ulteriori informazioni, vedere Services in Scope by Compliance Program.AWS

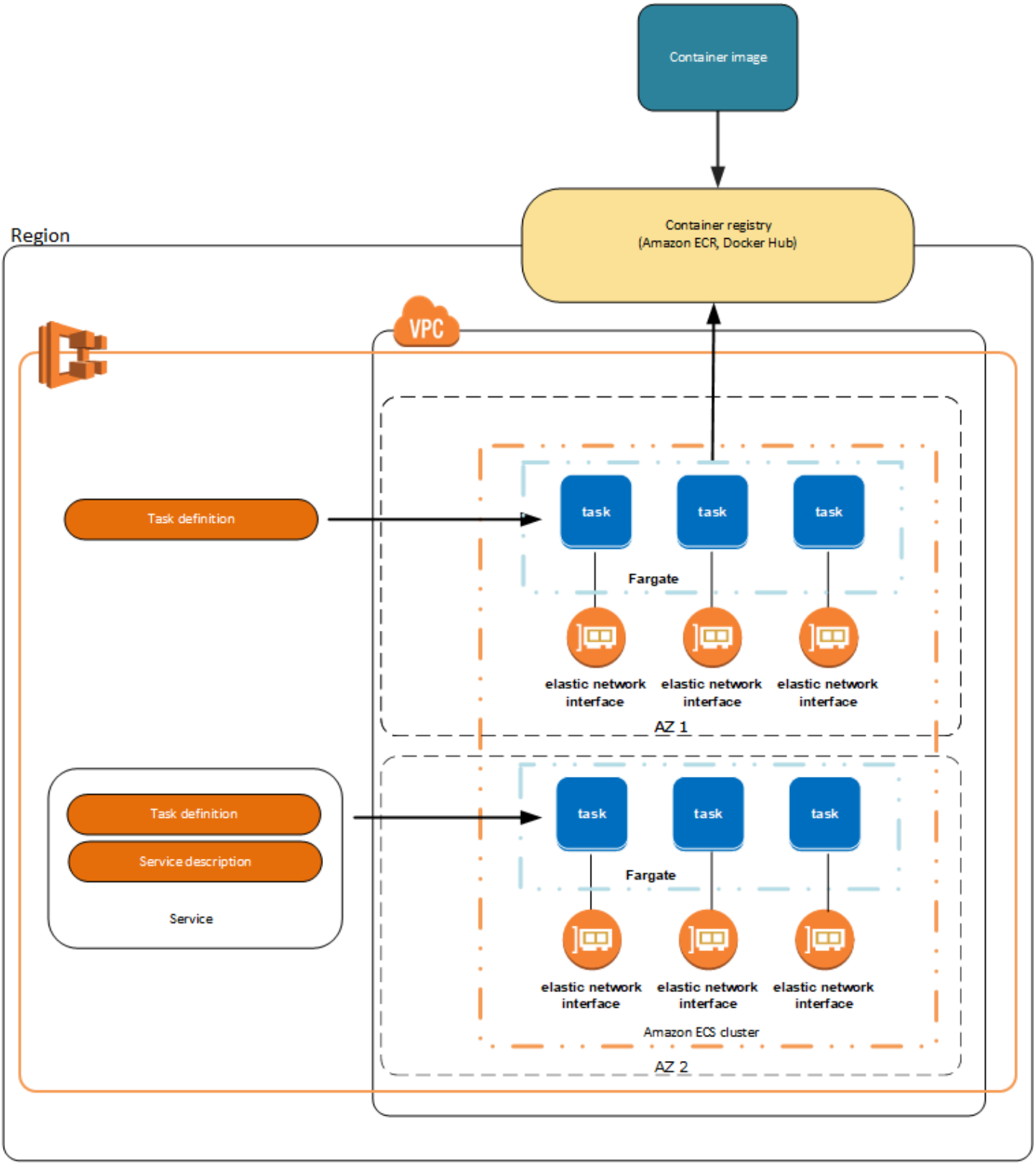
Fargate è adatto per i seguenti carichi di lavoro:

- Carichi di lavoro di grandi dimensioni che richiedono un sovraccarico operativo ridotto
- Piccoli carichi di lavoro con sequenza occasionale

- Carichi di lavoro ridotti
- Carichi di lavoro in batch

Per informazioni sulle regioni che supportano Fargate, consulta the section called “AWS Regioni di Fargate”.

Il diagramma seguente illustra l'architettura generale.



Per ulteriori informazioni su Amazon ECS con Fargate, consulta AWS Fargateper Amazon ECS.

EC2

Il tipo di avvio EC2 è adatto per carichi di lavoro di grandi dimensioni che devono essere ottimizzati per il prezzo.

Quando devi decidere come modellare i servizi e le definizioni di attività mediante il tipo di avvio EC2, è consigliabile considerare quali attività dovranno essere eseguite contemporaneamente e come dimensionare ciascun componente.

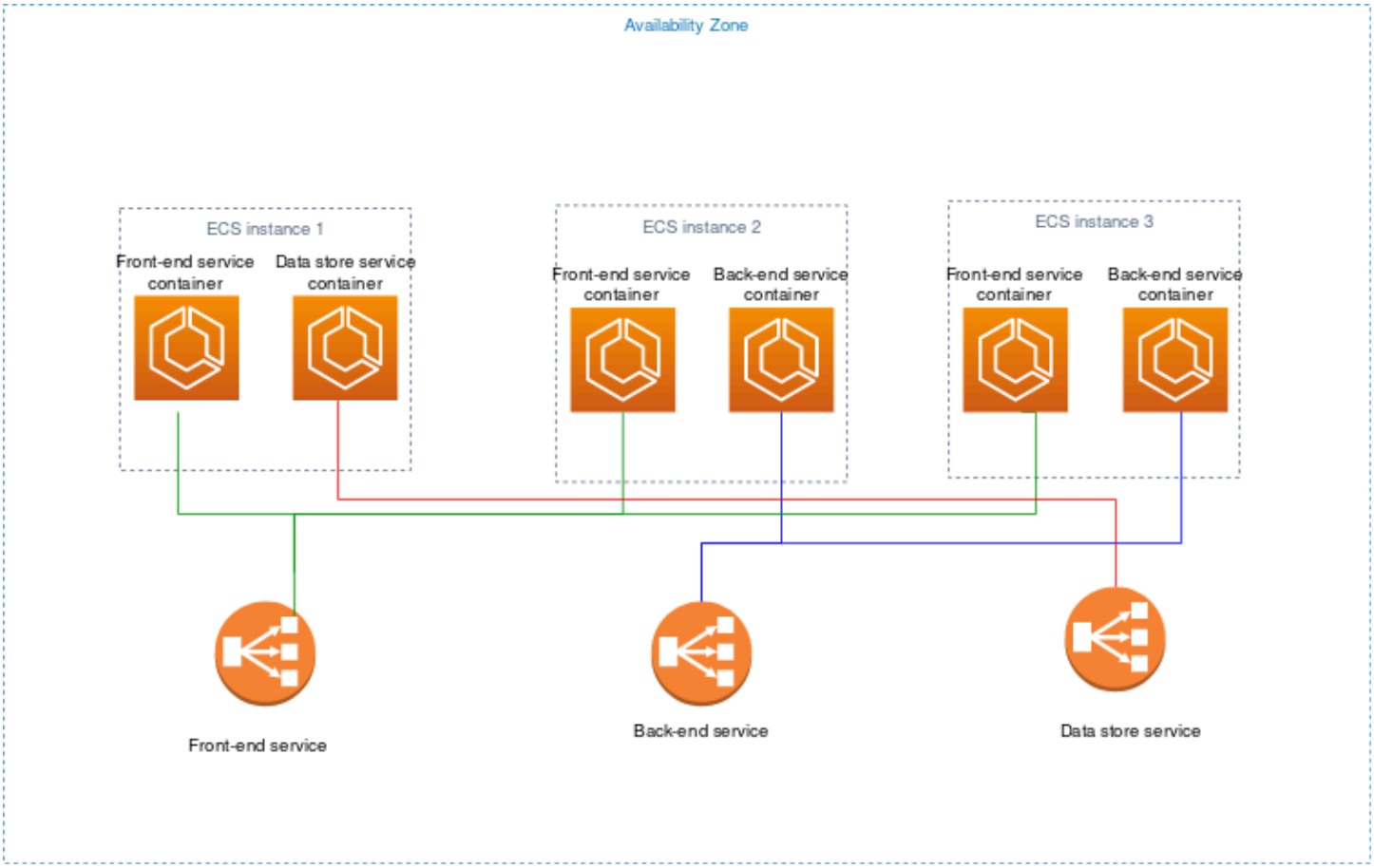
Ad esempio, immagina un'applicazione che comprende i seguenti componenti:

- Un servizio front-end che visualizza le informazioni su una pagina Web
- Un servizio back-end che fornisce le API per il servizio front-end
- Un datastore

Per questo esempio, crea definizioni di attività che raggruppino i container utilizzati per scopi comuni. Dividi i diversi componenti in definizioni di attività multiple e separate. Il seguente esempio di cluster presenta tre istanze di container su cui vengono eseguiti tre container per i servizi front-end, due container per i servizi back-end e un container per i servizi di datastore.

Puoi raggruppare i container correlati in una definizione di attività, ad esempio i container collegati che devono essere eseguiti contemporaneamente. Ad esempio, aggiungi un container per lo streaming dei log al servizio front-end, includendolo nella stessa definizione di attività.

Una volta configurate, dalle definizioni di attività puoi creare dei servizi per mantenere la disponibilità dei tuoi processi desiderati. Per ulteriori informazioni, consulta Creazione di un servizio Amazon ECS utilizzando la console. Nei servizi, puoi associare i container mediante i load balancer di Elastic Load Balancing. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS. Quando le tue esigenze relative alle applicazioni cambiano, puoi aggiornare i tuoi servizi per incrementare o ridurre il numero di attività desiderate. In alternativa, puoi aggiornare i servizi per implementare versioni più recenti dei container nelle attività. Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console.



Esterno

Il tipo di avvio esterno viene utilizzato per eseguire le applicazioni containerizzate sul server on-premise o sulla macchina virtuale (VM) che si registra nel cluster Amazon ECS e si gestisce in remoto. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio esterno.

Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones

Amazon ECS supporta carichi di lavoro che utilizzano Local Zones, Wavelength Zones e quando è richiesta una bassa latenza o AWS Outposts l'elaborazione locale dei dati.

- Puoi utilizzare Local Zones come estensione di e 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 AWS Outposts elaborazione locale dei dati nelle Domande frequenti. AWS Wavelength

Sottoreti condivise

È possibile utilizzare la Condivisione VPC per condividere sottoreti con altri account AWS all'interno della stessa AWS Organizations.

Puoi utilizzare VPC condivisi per il tipo di lancio di EC2 tenendo conto delle seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarla per le risorse Amazon ECS.
- Non puoi utilizzare il gruppo di sicurezza predefinito VPC per le istanze del contenitore perché appartiene al proprietario. Inoltre, i partecipanti non possono avviare istanze utilizzando gruppi di sicurezza di proprietà di altri partecipanti o del proprietario.
- In una sottorete condivisa, il partecipante e il proprietario controllano separatamente i gruppi di sicurezza all'interno di ciascun account. Il proprietario della sottorete può visualizzare i gruppi di sicurezza che sono stati creati dai partecipanti, ma non può eseguire operazioni sugli stessi. Se il proprietario della sottorete desidera rimuovere o modificare questi gruppi di sicurezza, è il partecipante che ha creato il gruppo di sicurezza che deve eseguire l'operazione.
- Il proprietario del VPC condiviso non può visualizzare, aggiornare o eliminare un cluster creato da un partecipante nella sottorete condivisa. Ciò si aggiunge alle risorse VPC alle quali ogni account ha un accesso diverso. Per ulteriori informazioni, consulta la sezione Responsabilità e autorizzazioni per proprietari e partecipanti nella Guida per l'utente di Amazon VPC.

Puoi utilizzare VPC condivisi per il tipo di lancio Fargate tenendo conto delle seguenti considerazioni:

- Il proprietario della sottorete VPC deve condividere una sottorete con un account partecipante prima che tale account possa utilizzarla per le risorse Amazon ECS.
- Non puoi creare un servizio o eseguire un'attività utilizzando il gruppo di sicurezza predefinito per il VPC perché appartiene al proprietario. Inoltre, i partecipanti non possono creare un servizio o eseguire un'attività utilizzando gruppi di sicurezza di proprietà di altri partecipanti o del proprietario.
- In una sottorete condivisa, il partecipante e il proprietario controllano separatamente i gruppi di sicurezza all'interno di ciascun account. Il proprietario della sottorete può visualizzare i gruppi di sicurezza che sono stati creati dai partecipanti, ma non può eseguire operazioni sugli stessi. Se il proprietario della sottorete desidera rimuovere o modificare questi gruppi di sicurezza, è il partecipante che ha creato il gruppo di sicurezza che deve eseguire l'operazione.
- Il proprietario del VPC condiviso non può visualizzare, aggiornare o eliminare un cluster creato da un partecipante nella sottorete condivisa. Ciò si aggiunge alle risorse VPC alle quali ogni account ha un accesso diverso. Per ulteriori informazioni, consulta la sezione Responsabilità e autorizzazioni per proprietari e partecipanti nella Guida per l'utente di Amazon VPC.

Per ulteriori informazioni sulla condivisione delle sottoreti VPC, consulta la pagina Condivisione del VPC con altri account nella Guida per l'utente di Amazon VPC.

Zone locali

Una zona locale è un'estensione di una Regione AWS zona geografica molto vicina agli utenti. Le zone locali hanno le loro connessioni a Internet e supportano AWS Direct Connect. Le risorse create in una zona locale possono servire gli utenti locali con comunicazioni a latenza molto bassa. Per ulteriori informazioni, consulta AWS Zone locali.

Una zona locale è rappresentata da un codice della regione seguito da un identificatore che indica la posizione, ad esempio `us-west-2-lax-1a`.

Per utilizzare una zona locale, è necessario prima attivare la zona. Una volta eseguita l'attivazione, dovrai creare un Amazon VPC e una sottorete nella zona locale.

Puoi avviare le istanze Amazon EC2, i server di file Amazon FSx e gli 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 istanze Amazon EC2 per eseguire applicazioni che richiedono una latenza estremamente bassa e una connessione nella regione. Servizi AWS

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, potrai avviare le istanze Amazon EC2 nella zona da utilizzare per i cluster e le attività Amazon ECS.

Per ulteriori informazioni, consulta Nozioni di base su AWS Wavelength nella Guida per gli sviluppatori di AWS Wavelength .

Le Wavelength Zone non sono disponibili in tutte. Regioni AWS Per informazioni sulle regioni che supportano le zone Wavelength, consulta Zone Wavelength disponibili nella Guida per gli sviluppatori di AWS Wavelength .

Amazon Elastic Container Service su AWS Outposts

AWS Outposts abilita AWS servizi, infrastrutture e modelli operativi nativi nelle strutture locali. Negli AWS Outposts ambienti, AWS è possibile utilizzare le stesse API, gli stessi strumenti e la stessa infrastruttura utilizzati in. Cloud AWS

Amazon ECS on AWS Outposts è ideale per carichi di lavoro a bassa latenza che devono essere eseguiti in prossimità di dati e applicazioni locali.

Per ulteriori informazioni su AWS Outposts, consulta la Guida per l'utente.AWS Outposts

Considerazioni

Di seguito sono riportate alcune considerazioni sull'utilizzo di Amazon ECS su: AWS Outposts

- Amazon Elastic Container Registry e Network Load Balancer vengono eseguiti nella AWS regione, non su. AWS Identity and Access Management AWS Outposts In tal modo si aumenta la latenza tra questi servizi e i container.
- AWS Fargate non è disponibile su. AWS Outposts

Di seguito sono riportate le considerazioni relative alla connettività di rete per AWS Outposts:

- Se la connettività di rete tra l'utente AWS Outposts e la relativa AWS regione viene interrotta, i cluster continueranno a funzionare. Tuttavia, non potrai creare nuovi cluster o intraprendere nuove azioni nei cluster esistenti fino a quando la connettività non viene ripristinata. In caso di errori di istanza, l'istanza non verrà sostituita automaticamente. L'agente CloudWatch Logs non sarà in grado di aggiornare i registri e i dati degli eventi.
- Ti consigliamo di fornire una connettività affidabile, ad alta disponibilità e a bassa latenza tra la tua regione AWS Outposts e quella corrispondente. AWS

Prerequisiti

Di seguito sono riportati i prerequisiti per l'utilizzo di Amazon ECS su: AWS Outposts

- Devi aver installato e configurato un Outpost nel data center locale.
- É necessaria una connessione di rete affidabile tra l'Outpost e la relativa Regione AWS .

Crea un cluster su AWS Outposts

Per creare un cluster Amazon ECS su un AWS Outposts con AWS CLI, specifica un gruppo di sicurezza e una sottorete da associare al tuo. AWS Outposts

Per creare una sottorete associata al tuo. AWS Outposts

```
aws ec2 create-subnet \
    --cidr-block 10.0.3.0/24 \
    --vpc-id vpc-xxxxxxxx \
    --outpost-arn arn:aws:outposts:us-west-2:123456789012:outpost/op-xxxxxxxxxxxxxxxxx \
    --availability-zone-id usw2-az1
```

Nell'esempio seguente viene creato un cluster Amazon ECS in un AWS Outposts.

1. Crea un ruolo e una politica 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 nel riferimento all'API IAM

```
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 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
```

5. Creare il cluster.

6. 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`.

7. 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 sull'istanza Linux all'avvio nella Amazon EC2 User Guide.

   ```
   aws ec2 run-instances --count 1 --image-id ami-xxxxxxxx --instance-type c5.large \
       --key-name aws-outpost-key --subnet-id subnet-xxxxxxxxxxxxxxxxx \
       --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.

8. Registra la tua definizione di processo. Usa il seguente comando e sostituisci `ecs-task.json` con il nome della definizione di processo.

   ```
   aws ecs register-task-definition --cli-input-json file://ecs-task.json
   ```

9. Eseguire l'attività o creare il servizio.

   Run the task

   ```
   aws ecs run-task --cluster mycluster --count 1 --task-definition outpost-app:1
   ```

Create the service

```
aws ecs create-service --cluster mycluster --service-name outpost-service \
    --task-definition outpost-app:1 --desired-count 1
```

Ottimizza la capacità e la disponibilità di Amazon ECS

La disponibilità delle applicazioni è fondamentale per fornire un'esperienza priva di errori e per ridurre al minimo la latenza delle applicazioni. La disponibilità dipende dalla disponibilità di risorse 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 (AZ). La scalabilità automatica gestisce il numero di attività o istanze in base a parametri definiti dall'utente, mentre le zone di disponibilità consentono di ospitare l'applicazione in posizioni isolate ma geograficamente vicine.

Come per quanto riguarda le dimensioni delle attività, la capacità e la disponibilità presentano alcuni compromessi da considerare. Idealmente, la capacità sarebbe perfettamente allineata alla domanda. Ci sarebbe sempre la capacità sufficiente per soddisfare le richieste ed elaborare i lavori per soddisfare gli obiettivi del livello di servizio (SLO), tra cui una bassa latenza e un tasso di errore. La capacità non sarebbe mai troppo elevata, con conseguenti costi eccessivi, né troppo bassa, con conseguenti alti tassi di latenza e di errore.

La scalabilità automatica è un processo latente. Innanzitutto, è necessario fornire metriche in tempo reale a. CloudWatch Quindi, devono essere aggregate per l'analisi, che può richiedere fino a diversi minuti a seconda della granularità della metrica. CloudWatch confronta le metriche con le soglie di allarme per identificare una carenza o un eccesso di risorse. Per evitare l'instabilità, configura gli allarmi in modo da richiedere che la soglia impostata venga superata per alcuni minuti prima che l'allarme si spenga. Inoltre, è necessario del tempo per fornire nuove attività e terminare quelle che non sono più necessarie.

A causa di questi potenziali ritardi nel sistema descritto, è importante mantenere un certo margine di manovra mediante un approvvigionamento eccessivo. In questo modo è possibile far fronte a picchi di domanda a breve termine. Ciò consente inoltre all'applicazione di soddisfare richieste aggiuntive senza raggiungere la saturazione. È buona norma impostare l'obiettivo di scalabilità tra il 60 e l'80% dell'utilizzo. Questo aiuta l'applicazione a gestire meglio i picchi di domanda extra mentre la capacità aggiuntiva è ancora in fase di provisioning.

Un altro motivo per cui consigliamo di effettuare un eccesso di provisioning è quello di poter rispondere rapidamente ai guasti della zona di disponibilità. 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. Cioè, esegui tre attività ogni due necessarie per il servizio ordinario.

Massimizzazione della velocità di scalabilità

La scalabilità automatica è un processo reattivo che richiede tempo per avere effetto. Tuttavia, esistono alcuni modi per ridurre al minimo il tempo necessario per la scalabilità orizzontale.

Ridurre al minimo le dimensioni dell'immagine. Le immagini più grandi richiedono più tempo per essere scaricate da un archivio di immagini e decomprimute. Pertanto, mantenere le dimensioni delle immagini più piccole riduce il tempo necessario per l'avvio di un contenitore. Per ridurre le dimensioni dell'immagine, puoi seguire questi consigli specifici:

- Se puoi creare un file binario statico o usare Golang, crea la tua immagine `FROM scratch` e includi solo la tua applicazione binaria nell'immagine risultante.
- Usa immagini di base ridotte al minimo da fornitori di distribuzioni upstream, come Amazon Linux o Ubuntu.
- Non includete artefatti di costruzione nell'immagine finale. L'utilizzo di build in più fasi può aiutarti in questo senso.
- `RUN`Palchi compatti laddove possibile. Ogni `RUN` fase crea un nuovo livello di immagine, che comporta un ulteriore percorso di andata e ritorno per scaricare il layer. Una singola `RUN` fase con più comandi uniti && ha meno livelli di una con più `RUN` fasi.
- Se desideri includere dati, come i dati di inferenza ML, nell'immagine finale, includi solo i dati necessari per avviare e iniziare a generare traffico. Se recuperi dati su richiesta da Amazon S3 o altro storage senza influire sul servizio, archivia invece i dati in quei luoghi.

Tieni le immagini vicine. Maggiore è la latenza di rete, maggiore è il tempo necessario per scaricare l'immagine. Ospita le tue immagini in un repository nella stessa 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 contenitori. L'hosting delle immagini nella stessa regione migliora l'affidabilità complessiva. Riduce il rischio di problemi di connettività di rete e problemi di disponibilità in una regione diversa. In alternativa, puoi anche implementare la replica interregionale di Amazon ECR per aiutarti in questo.

Riduci le soglie di controllo dello stato del sistema di bilanciamento del carico. I sistemi di bilanciamento del carico eseguono controlli di integrità prima di inviare traffico all'applicazione. La configurazione predefinita del controllo dello stato di salute per un gruppo target può richiedere 90 secondi o più. Durante questo periodo, il load balancer controlla lo stato di salute e riceve le richieste. La riduzione dell'intervallo dei controlli sanitari e del numero di soglie può far sì che l'applicazione accetti il traffico più rapidamente e riduca il carico su altre attività.

Considerate le prestazioni con avvio a freddo. Alcune applicazioni utilizzano runtime come Java per eseguire la compilazione 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 basate sul tracciamento degli obiettivi. 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 istanze Amazon EC2 per fornire capacità di cluster, prendi in considerazione i seguenti consigli:

Usa istanze Amazon EC2 più grandi e volumi Amazon EBS più veloci. Puoi migliorare la velocità di download e preparazione delle immagini utilizzando un'istanza Amazon EC2 più grande e un volume Amazon EBS più veloce. All'interno di una determinata famiglia di istanze Amazon EC2, la rete e il throughput massimo di Amazon EBS aumentano all'aumentare delle dimensioni dell'istanza (ad esempio, da a). `m5.xlarge` `m5.2xlarge` Inoltre, puoi personalizzare i volumi Amazon EBS per

aumentarne il throughput e gli IOPS. Ad esempio, se utilizzi volumi, utilizza `gp2` volumi più grandi che offrono un throughput di base maggiore. Se utilizzi `gp3` volumi, specifica la velocità effettiva e gli IOPS quando crei il volume.

Usa la modalità di rete bridge per le attività in esecuzione su istanze Amazon EC2. Le attività che utilizzano la modalità di `bridge` rete su Amazon EC2 vengono avviate più rapidamente delle attività che utilizzano la modalità di `awsvpc` rete. Quando viene utilizzata la modalità di `awsvpc` rete, Amazon ECS collega un'elastic network interface (ENI) all'istanza prima di avviare l'attività. Ciò introduce una latenza aggiuntiva. Tuttavia, ci sono diversi compromessi per l'utilizzo del bridge networking. Queste attività non dispongono di un proprio gruppo di sicurezza e vi sono alcune implicazioni per il bilanciamento del carico. Per ulteriori informazioni, consulta Load Balancer target group nella Elastic Load Balancing User Guide.

Gestione degli shock legati alla domanda

Alcune applicazioni subiscono improvvisi forti shock della domanda. Ciò accade per una serie di motivi: un evento giornalistico, una grande vendita, un evento mediatico o qualche altro evento che diventa virale e causa un aumento rapido e significativo del traffico in un lasso di tempo molto breve. Se non pianificato, ciò può far sì che la domanda superi rapidamente le risorse disponibili.

Il modo migliore per gestire gli shock legati alla domanda è prevederli e pianificare di conseguenza. Poiché la scalabilità automatica può richiedere del tempo, consigliamo di ridimensionare l'applicazione prima che cominci lo shock della domanda. Per ottenere risultati ottimali, consigliamo di adottare un piano aziendale che preveda una stretta collaborazione tra i team che utilizzano un calendario condiviso. Il team che pianifica l'evento dovrebbe lavorare in anticipo a stretto contatto con il team responsabile della candidatura. Questo dà al team abbastanza tempo per avere un piano di pianificazione chiaro. Possono pianificare la scalabilità orizzontale prima dell'evento e ampliarla dopo l'evento. Per ulteriori informazioni, consulta Dimensionamento pianificato nella Guida per l'utente di Dimensionamento automatico delle applicazioni.

Se disponi di un piano Enterprise Support, assicurati di collaborare anche con il tuo Technical Account Manager (TAM). Il TAM può verificare le quote di servizio e garantire che le quote necessarie vengano aumentate prima dell'inizio dell'evento. In questo modo, non si raggiungono accidentalmente le quote di servizio. Inoltre, possono aiutarvi con servizi di preriscaldamento, come i sistemi di bilanciamento del carico, per assicurarvi che l'evento si svolga senza intoppi.

La gestione degli shock imprevisti della domanda è un problema più difficile. Gli shock non programmati, se di ampiezza sufficientemente elevata, possono far sì che la domanda superi

rapidamente la capacità. Può anche superare la capacità di reazione della scalabilità automatica. Il modo migliore per prepararsi agli shock non programmati è quello di fornire risorse in eccesso. È necessario disporre di risorse sufficienti per gestire la massima richiesta di traffico prevista in qualsiasi momento.

Mantenere la capacità massima in previsione di shock non programmati della domanda può essere costoso. Per mitigare l'impatto sui costi, individuate un indicatore, una metrica o un evento anticipatore che preveda l'imminenza di un forte shock della domanda. Se la metrica o l'evento forniscono in modo affidabile un preavviso significativo, avvia il processo di scalabilità orizzontale immediatamente quando si verifica l'evento o quando la metrica supera la soglia specifica che hai impostato.

Se la tua applicazione è soggetta a improvvisi shock di domanda non programmati, prendi in considerazione l'aggiunta di una modalità ad alte prestazioni all'applicazione che sacrifichi le funzionalità non critiche ma mantenga funzionalità cruciali per il cliente. Ad esempio, supponiamo che l'applicazione possa passare dalla generazione di costose risposte personalizzate alla pubblicazione di una pagina di risposta statica. In questo scenario, è possibile aumentare in modo significativo la velocità effettiva senza scalare affatto l'applicazione.

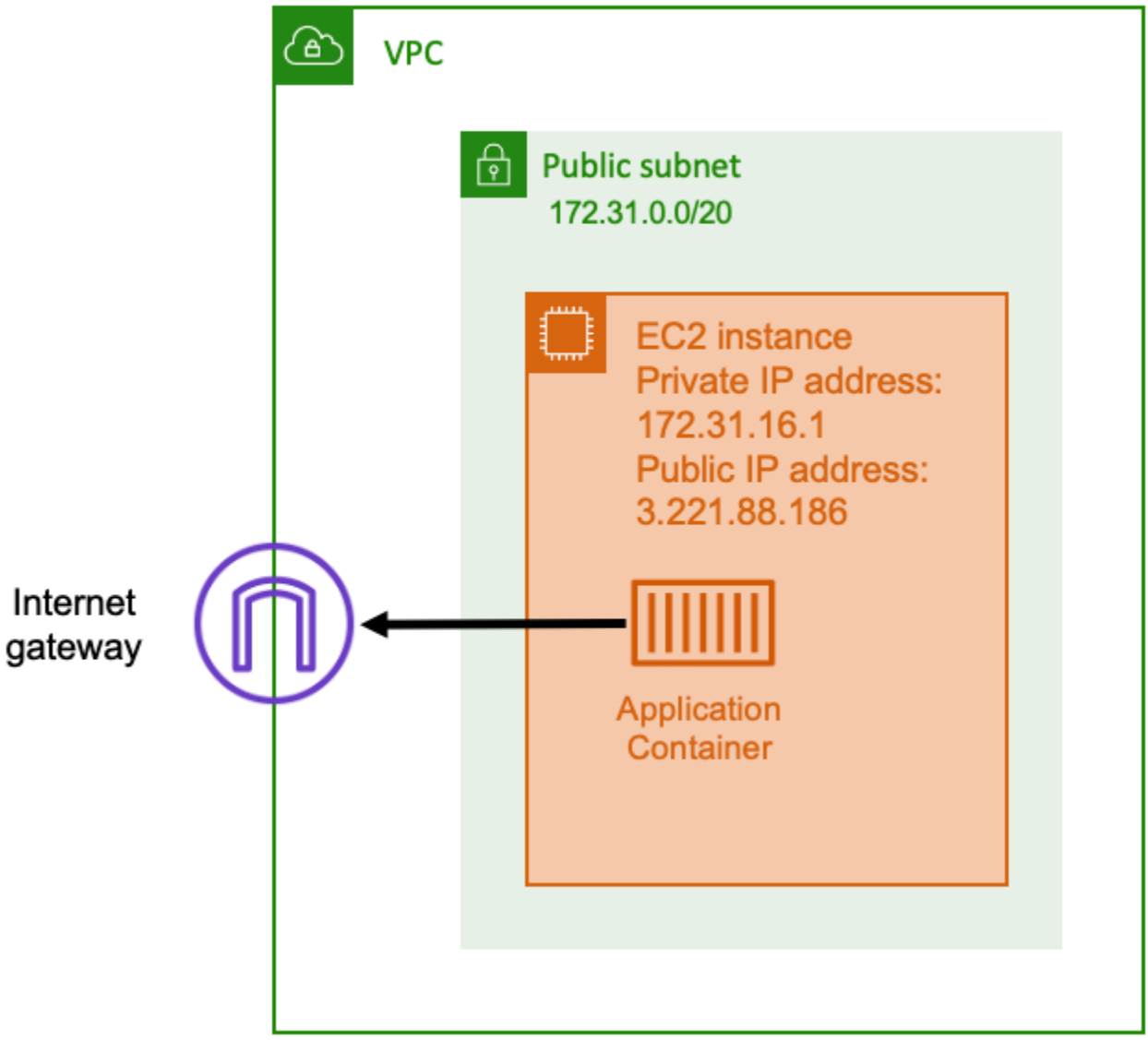
Infine, potete prendere in considerazione la possibilità di smontare i servizi monolitici per far fronte meglio agli shock della domanda. Se la tua applicazione è un servizio monolitico, costoso da eseguire e lento da scalare, potresti riuscire a estrarre o riscrivere parti essenziali per le prestazioni ed eseguirle come servizi separati. Questi nuovi servizi possono quindi essere scalati indipendentemente dai componenti meno critici. Avere la flessibilità necessaria per scalare le funzionalità essenziali in termini di prestazioni separatamente dalle altre parti dell'applicazione può ridurre il tempo necessario per aggiungere capacità e contribuire a ridurre i costi.

Connetti le applicazioni Amazon ECS a Internet

La maggior parte delle applicazioni containerizzate ha almeno alcuni componenti che richiedono l'accesso in uscita a Internet. Ad esempio, il backend di un'app mobile richiede l'accesso in uscita alle notifiche push.

Amazon Virtual Private Cloud offre due metodi principali per facilitare la comunicazione tra il tuo VPC e Internet.

Sottorete pubblica e gateway Internet



Quando si utilizza una sottorete pubblica con un percorso verso un gateway Internet, l'applicazione containerizzata può essere eseguita su un host all'interno di un VPC su una sottorete pubblica. All'host che gestisce il contenitore viene assegnato un indirizzo IP pubblico. Questo indirizzo IP pubblico è instradabile da Internet. Per ulteriori informazioni, consulta la sezione Gateway Internet nella Guida per l'utente di Amazon VPC.

Questa architettura di rete facilita la comunicazione diretta tra l'host che esegue l'applicazione e altri host su Internet. La comunicazione è bidirezionale. Ciò significa che non solo puoi stabilire una connessione in uscita con qualsiasi altro host su Internet, ma anche altri host su Internet potrebbero

tentare di connettersi al tuo host. Pertanto, è necessario prestare molta attenzione al gruppo di sicurezza e alle regole del firewall. Ciò garantisce che altri host su Internet non possano aprire connessioni che non desideri vengano aperte.

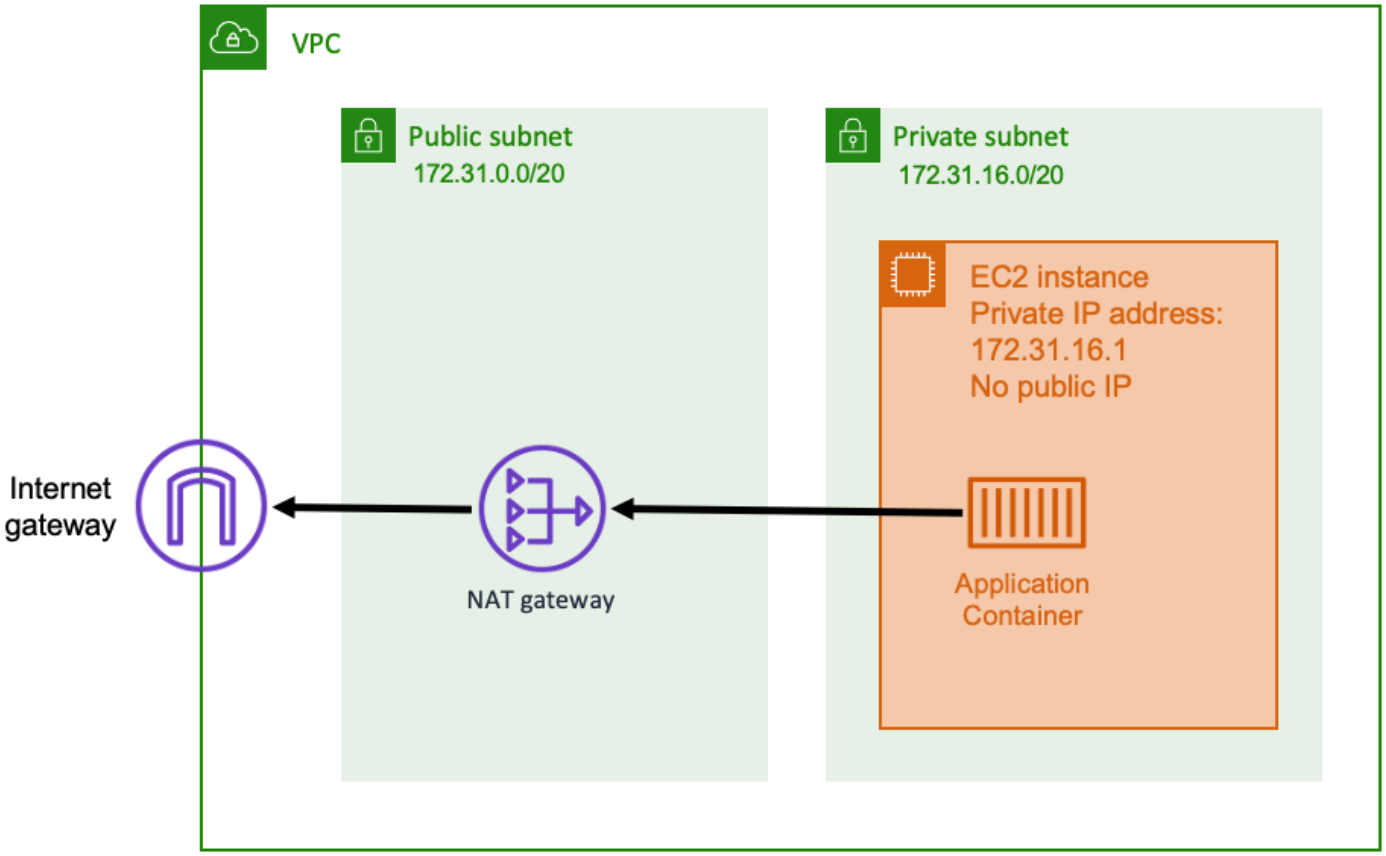
Ad esempio, se l'applicazione viene eseguita su Amazon EC2, assicurati che la porta 22 per l'accesso SSH non sia aperta. In caso contrario, l'istanza potrebbe ricevere continui tentativi di connessione SSH da parte di bot malintenzionati su Internet. Questi bot esplorano gli indirizzi IP pubblici. Dopo 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 istanze EC2 su una sottorete pubblica. Amazon ECS utilizza queste istanze EC2 come capacità del cluster e qualsiasi contenitore in esecuzione sulle istanze può utilizzare l'indirizzo IP pubblico sottostante dell'host per le reti in uscita. Questo vale sia per le modalità di rete che per quelle di rete. `host` `bridge` Tuttavia, la modalità di `awsvpc` rete non fornisce ai task ENI indirizzi IP pubblici. Pertanto, non possono utilizzare direttamente un gateway Internet.
- Fargate: quando crei il tuo servizio Amazon ECS, specifica le sottoreti pubbliche per la configurazione di rete del servizio e utilizza l'opzione Assegna indirizzo IP pubblico. Ogni attività Fargate è collegata in rete nella sottorete pubblica e dispone di un proprio indirizzo IP pubblico per la comunicazione diretta con Internet.

Sottorete privata e gateway NAT



Quando si utilizza una sottorete privata e un gateway NAT, è possibile eseguire l'applicazione containerizzata su un host che si trova in una sottorete privata. Pertanto, questo host dispone di un indirizzo IP privato che è instradabile all'interno del tuo VPC, ma non è instradabile da Internet. Ciò significa che altri host all'interno del VPC possono connettersi all'host utilizzando il suo indirizzo IP privato, ma gli altri host su Internet non possono effettuare comunicazioni in entrata verso l'host.

Con una sottorete privata, è possibile utilizzare un gateway NAT (Network Address Translation) per consentire a un host all'interno di una sottorete privata di connettersi a Internet. Gli host su Internet ricevono una connessione in entrata che sembra provenire dall'indirizzo IP pubblico del gateway NAT che si trova all'interno di una sottorete pubblica. Il gateway NAT funge da ponte tra Internet e il VPC privato. Questa configurazione è spesso preferita per motivi di sicurezza perché significa che il tuo VPC è protetto dall'accesso diretto degli aggressori su Internet. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.

Questo approccio di rete privata è adatto per scenari in cui si desidera proteggere i contenitori dall'accesso esterno diretto. Gli scenari applicabili includono sistemi di elaborazione dei pagamenti o

contenitori che archiviano dati e password degli utenti. Ti vengono addebitati solo i costi di creazione E di utilizzo di un gateway NAT nell'account. Si applicano anche le tariffe orarie di utilizzo e di elaborazione dei dati del gateway NAT. Ai fini della ridondanza, è necessario disporre di un gateway NAT in ogni zona di disponibilità. In questo modo, la perdita di disponibilità di una singola zona di disponibilità non compromette la connettività in uscita. Per questo motivo, se hai un carico di lavoro ridotto, potrebbe essere più conveniente utilizzare sottoreti private e gateway NAT.

Questo approccio di rete è supportato sia quando si utilizza Amazon ECS su Amazon EC2 sia quando lo si utilizza su. AWS Fargate

- Amazon EC2: puoi avviare istanze EC2 su una sottorete privata. I contenitori eseguiti su questi host EC2 utilizzano la rete degli host sottostanti e le richieste in uscita passano attraverso il gateway NAT.
- Fargate: quando crei il servizio Amazon ECS, specificate le sottoreti private per la configurazione di rete del servizio e non utilizzate l'opzione Assegna indirizzo IP pubblico. Ogni attività Fargate è ospitata in una sottorete privata. Il suo traffico in uscita viene instradato attraverso qualsiasi gateway NAT associato a quella sottorete privata.

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

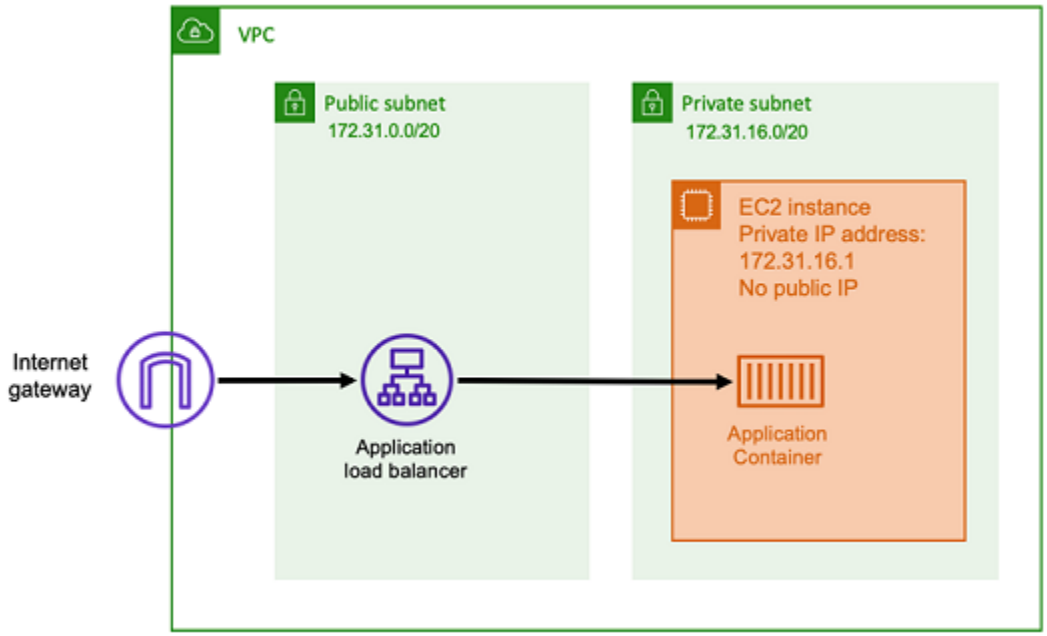
Se gestisci un servizio pubblico, devi accettare il traffico in entrata da Internet. Ad esempio, il sito Web pubblico deve accettare richieste HTTP in entrata dai browser. In tal caso, anche altri host su Internet devono avviare una connessione in entrata all'host dell'applicazione.

Un approccio a questo problema consiste nell'avviare i contenitori su host che si trovano in una sottorete pubblica con un indirizzo IP pubblico. Tuttavia, non lo consigliamo per applicazioni su larga scala. Per queste, un approccio migliore consiste nell'avere un livello di input scalabile che si colloca tra Internet e l'applicazione. Per questo approccio, è possibile utilizzare uno qualsiasi dei AWS servizi elencati in questa sezione come input.

Application Load Balancer

Un Application Load Balancer funziona a livello di applicazione. È il settimo livello del modello Open Systems Interconnection (OSI). Ciò rende un Application Load Balancer adatto ai servizi HTTP pubblici. Se disponi di un sito Web o di un'API REST HTTP, un Application Load Balancer è un

sistema di bilanciamento del carico adatto per questo carico di lavoro. Per ulteriori informazioni, consulta Cos'è un Application Load Balancer? nella Guida per l'utente di Application Load Balancers.



Con questa architettura, si crea un Application Load Balancer in una sottorete pubblica in modo che abbia un indirizzo IP pubblico e possa ricevere connessioni in entrata da Internet. Quando l'Application Load Balancer riceve una connessione in entrata, o più specificamente una richiesta HTTP, apre una connessione all'applicazione utilizzando il suo indirizzo IP privato. Quindi, inoltra la richiesta tramite la connessione interna.

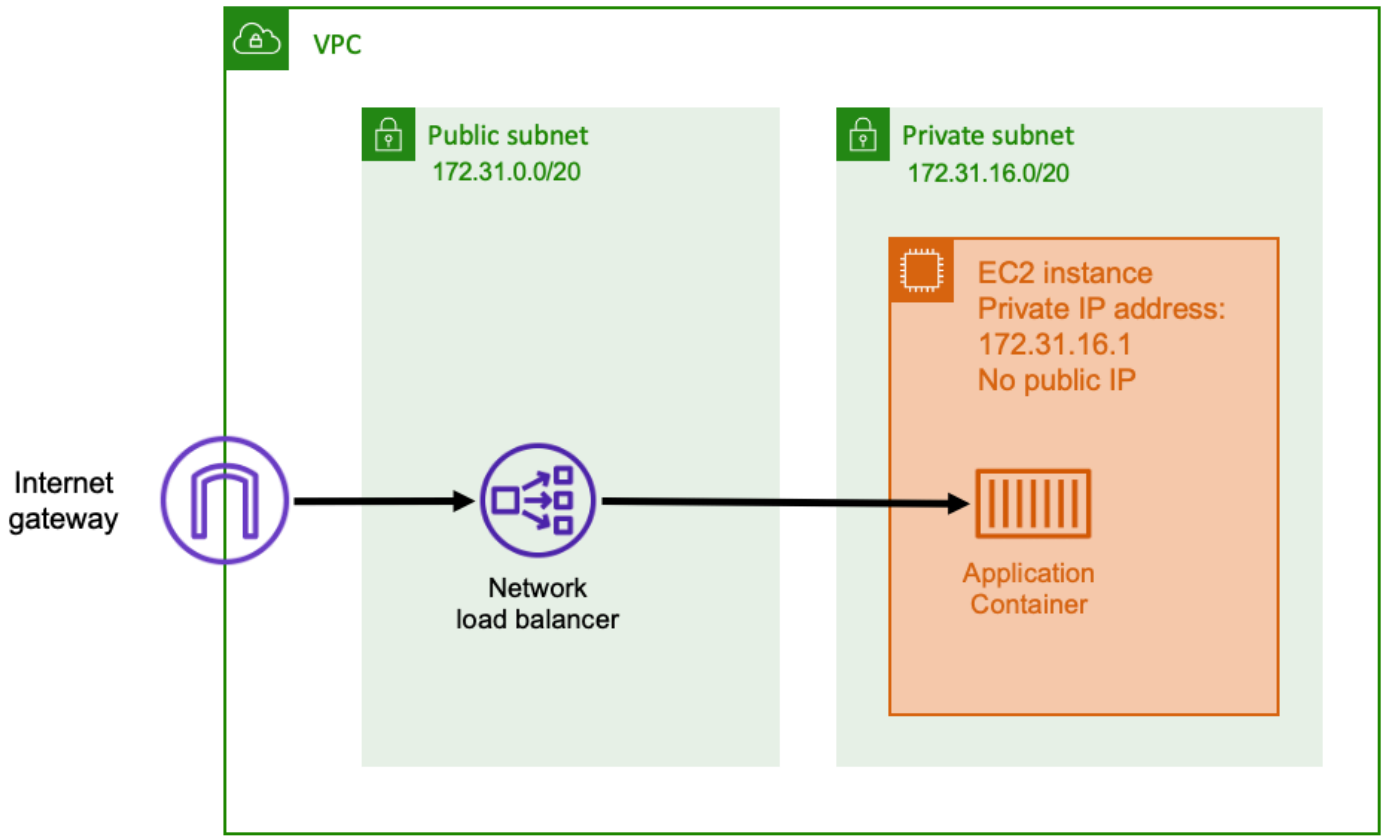
Un Application Load Balancer presenta i seguenti vantaggi.

- Terminazione SSL/TLS: un Application Load Balancer può supportare comunicazioni e certificati HTTPS sicuri per le comunicazioni con i client. Facoltativamente, può interrompere la connessione SSL a livello di load balancer in modo da non dover gestire i certificati nella propria applicazione.
- Routing avanzato: un Application Load Balancer può avere più nomi host DNS. Dispone inoltre di funzionalità di routing avanzate per inviare richieste HTTP in entrata a diverse destinazioni in base a metriche come il nome host o il percorso della richiesta. Ciò significa che è possibile utilizzare un singolo Application Load Balancer come input per molti servizi interni diversi o anche microservizi su percorsi diversi di un'API REST.
- Supporto gRPC e websocket: un Application Load Balancer può gestire più di un semplice HTTP. Può anche bilanciare il carico di servizi basati su gRPC e websocket, con supporto HTTP/2.
- Sicurezza: un Application Load Balancer aiuta a proteggere l'applicazione dal traffico dannoso. Include funzionalità come le mitigazioni della sincronizzazione HTTP ed è integrato con AWS

Web Application Firewall ()AWS WAF. AWS WAF può filtrare ulteriormente il traffico dannoso che potrebbe contenere schemi di attacco, come SQL injection o cross-site scripting.

Network Load Balancer

Un sistema Network Load Balancer funziona al quarto livello del modello Open Systems Interconnection (OSI). È adatto per protocolli o scenari non HTTP in cui è necessaria end-to-end la crittografia, ma non ha le stesse funzionalità specifiche HTTP di un Application Load Balancer. Pertanto, un Network Load Balancer è più adatto per le applicazioni che non utilizzano HTTP. Per ulteriori informazioni, consulta Cos'è un Network Load Balancer? nella Guida per l'utente di Network Load Balancers.



Quando un Network Load Balancer viene utilizzato come input, funziona in modo simile a un Application Load Balancer. Questo perché è creato in una sottorete pubblica e dispone di un indirizzo IP pubblico a cui è possibile accedere su Internet. Il Network Load Balancer apre quindi una connessione all'indirizzo IP privato dell'host che esegue il container e invia i pacchetti dal lato pubblico a quello privato.

Funzionalità di Network Load Balancer

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

- nd-to-end Crittografia E: poiché un Network Load Balancer opera al quarto livello del modello OSI, non legge il contenuto dei pacchetti. Ciò lo rende adatto per il bilanciamento del carico di comunicazioni che richiedono la crittografia. end-to-end
- Crittografia TLS: oltre alla end-to-end crittografia, Network Load Balancer può anche interrompere le connessioni TLS. In questo modo, le tue applicazioni di backend non devono implementare il proprio TLS.
- Supporto UDP: poiché un Network Load Balancer opera al quarto livello del modello OSI, è adatto per carichi di lavoro e protocolli non HTTP diversi dal TCP.

Chiusura delle connessioni

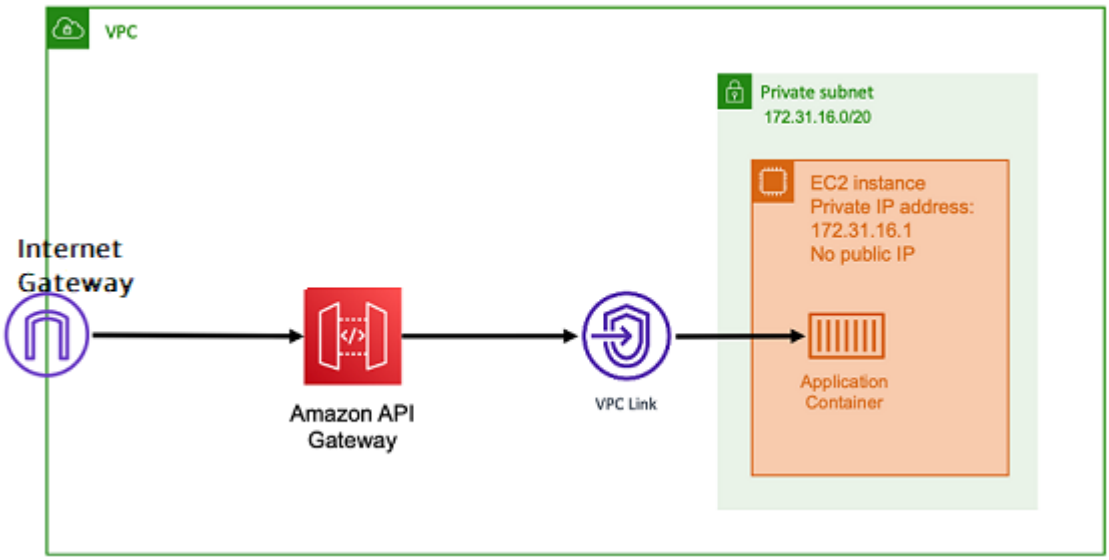
Poiché Network Load Balancer non rispetta il protocollo applicativo ai livelli superiori del modello OSI, non può inviare messaggi di chiusura ai client in tali protocolli. A differenza dell'Application Load Balancer, tali connessioni devono essere chiuse dall'applicazione oppure è possibile configurare Network Load Balancer per chiudere le connessioni di quarto livello quando un'attività viene interrotta o sostituita. Vedi l'impostazione di terminazione della connessione per i gruppi target di Network Load Balancer nella documentazione di Network Load Balancer.

Consentire al Network Load Balancer di chiudere le connessioni al quarto livello può far sì che i client visualizzino messaggi di errore indesiderati, se il client non li gestisce. Per ulteriori informazioni sulla configurazione del client consigliata, consulta la libreria per gli sviluppatori qui.

I metodi per chiudere le connessioni variano a seconda dell'applicazione, tuttavia un modo è garantire che il ritardo di annullamento della registrazione del target di Network Load Balancer sia più lungo del timeout della connessione client. Il client prima andava in timeout e si riconnetteva senza problemi tramite Network Load Balancer all'attività successiva, mentre la vecchia operazione esauriva lentamente tutti i client. Per ulteriori informazioni sul ritardo di annullamento della registrazione previsto per Network Load Balancer, consulta la documentazione di Network Load Balancer.

API HTTP di Amazon API Gateway

Amazon API Gateway è adatto per applicazioni HTTP con picchi improvvisi di volumi di richieste o volumi di richieste bassi. Per ulteriori informazioni, consulta Cos'è Amazon API Gateway? nella Guida per sviluppatori di API Gateway.



Il modello di prezzo per Application Load Balancer e Network Load Balancer include una tariffa oraria per mantenere i sistemi di bilanciamento del carico disponibili per accettare connessioni in entrata in ogni momento. Al contrario, API Gateway addebita separatamente ogni richiesta. Ciò ha l'effetto che, se non arriva alcuna richiesta, non ci sono costi. In caso di carichi di traffico elevati, un Application Load Balancer o un Network Load Balancer è in grado di gestire un volume maggiore di richieste a un prezzo per richiesta inferiore rispetto all'API Gateway. Tuttavia, se hai un numero complessivo di richieste limitato o hai periodi di traffico ridotto, il prezzo cumulativo per l'utilizzo dell'API Gateway dovrebbe essere più conveniente rispetto al pagamento di una tariffa oraria per mantenere un sistema di bilanciamento del carico sottoutilizzato. L'API Gateway può anche memorizzare nella cache le risposte API, il che potrebbe comportare una riduzione delle percentuali di richieste di backend.

Funzioni API Gateway che utilizzano un collegamento VPC che consente al servizio AWS gestito di connettersi agli host all'interno della sottorete privata del VPC, utilizzando il relativo indirizzo IP privato. Può rilevare questi indirizzi IP privati esaminando i record di rilevamento dei AWS Cloud Map servizi gestiti da Amazon ECS Service Discovery.

API Gateway supporta le seguenti funzionalità.

- Il funzionamento dell'API Gateway è simile a quello di un sistema di bilanciamento del carico, ma presenta funzionalità aggiuntive esclusive per la gestione delle API
- L'API Gateway offre funzionalità aggiuntive relative all'autorizzazione del client, ai livelli di utilizzo e alla modifica della richiesta/risposta. Per ulteriori informazioni, consulta le funzionalità di Amazon API Gateway.
- L'API Gateway può supportare endpoint gateway API edge, regionali e privati. Gli endpoint Edge sono disponibili tramite una distribuzione gestita CloudFront . Gli endpoint regionali e privati sono entrambi locali rispetto a una regione.
- Terminazione SSL/TLS
- Instradamento di diversi percorsi HTTP verso diversi microservizi di backend

Oltre alle funzionalità precedenti, API Gateway supporta anche l'utilizzo di autorizzatori Lambda personalizzati che puoi utilizzare per proteggere l'API dall'utilizzo non autorizzato. Per ulteriori informazioni, consulta Field Notes: API serverless basate su container con Amazon ECS e Amazon API Gateway.

Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account

È possibile accedere alle impostazioni dell'account Amazon ECS per attivare o disattivare determinate caratteristiche. Per ogni Regione AWS, puoi attivare o disattivare ogni impostazione dell'account a livello di account o per un utente o ruolo specifico.

È possibile attivare o disattivare determinate caratteristiche se una delle seguenti caratteristiche è rilevante per te:

- Un utente o ruolo può attivare o disattivare specifiche impostazioni del singolo account utente.
- Un utente o ruolo può configurare l'impostazione di attivazione o disattivazione predefinita per tutti gli utenti nell'account.
- L'utente root o un utente con privilegi di amministratore può attivare o disattivare qualsiasi ruolo o utente specifico sull'account. Se l'impostazione dell'account per l'utente root viene modificata, l'impostazione di default viene configurata per tutti gli utenti e i ruoli per i quali non è stata selezionata una singola impostazione dell'account.

> **Note**
>
> Gli utenti federati presuppongono l'impostazione dell'account dell'utente root e non possono avere impostazioni account esplicite impostate per loro separatamente.

Sono disponibili le seguenti impostazioni dell'account. È necessario attivare e disattivare separatamente ciascuna impostazione dell'account.

Amazon Resource Name (ARN) e ID

Nomi delle risorse: `serviceLongArnFormat`, `taskLongArnFormat` e `containerInstanceLongArnFormat`

Amazon ECS sta introducendo un nuovo formato per Amazon Resource Name (ARN) e ID risorsa per servizi, attività e istanze di container. 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. Per ulteriori informazioni, consulta Amazon Resource Name (ARN) e ID.

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.

AWSVPC trunking

Nome risorsa: `awsvpcTrunking`

Amazon ECS supporta l'avvio di istanze di container con densità dell'interfaccia di rete elastica (ENI) aumentata tramite i tipi di istanze Amazon EC2 supportati. Quando utilizzi questi tipi di istanze e fornisci il consenso esplicito all'impostazione dell'account `awsvpcTrunking`, ENI aggiuntive sono disponibili su nuove istanze di container avviate. Questa configurazione consente di posizionare più processi utilizzando la modalità di rete `awsvpc` su ogni istanza di container. Utilizzando questa caratteristica, un'istanza `c5.large` con `awsvpcTrunking` abilitata dispone di una quota ENI aumentata di dieci. 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. L'interfaccia di rete primaria e l'interfaccia di rete trunk non vengono conteggiate per la quota ENI. Pertanto, questa configurazione consente di avviare dieci processi sull'istanza di container anziché i due processi correnti. Per ulteriori informazioni, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.

Il valore predefinito è `disabled`.

Solo le risorse avviate dopo l'attivazione ricevono i limiti ENI aumentati. Tutte le risorse esistenti non sono interessate. Per eseguire la transizione di un'istanza di container alle quote ENI aumentate, l'istanza di container deve essere eliminata ne deve essere registrata una nuova nel cluster.

CloudWatch Informazioni sui container

Nome risorsa: `containerInsights`

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. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che consentono di isolare i problemi e risolverli in modo rapido. Puoi anche impostare CloudWatch allarmi sulle metriche raccolte da Container Insights. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.

Quando accetti le impostazioni `containerInsights`, tutti i nuovi cluster hanno Container Insights abilitato di default. Puoi disattivare questa impostazione per cluster specifici al momento della loro creazione. Puoi anche modificare questa impostazione utilizzando l'API. UpdateClusterSettings

Per cluster contenenti processi o servizi che utilizzano il tipo di avvio EC2, per utilizzare Container Insights le istanze di container devono eseguire la versione 1.29.0 o successive dell'agente Amazon ECS. Per ulteriori informazioni, consulta Gestione delle istanze di container Amazon ECS Linux.

Il valore predefinito è `disabled`.

IPv6 VPC dual-stack

Nome risorsa: `dualStackIPv6`

Amazon ECS supporta la specifica di processi con un indirizzo IPv6 in aggiunta all'indirizzo IPv4 privato principale.

Affinché i processi ricevano un indirizzo IPv6, il processo deve utilizzare la modalità di rete `awsvpc`, deve essere avviato in un VPC configurato per la modalità dual-stack e deve essere abilitata l'impostazione dell'account `dualStackIPv6`. Per ulteriori informazioni su altri requisiti, consulta Utilizzo di un VPC in modalità dual-stack il tipo di lancio EC2 e Utilizzo di un VPC in modalità dual-stack il tipo di lancio Fargate.

> **Important**
>
> L'impostazione dell'account `dualStackIPv6` può essere modificata solo utilizzando l'API Amazon ECS o la AWS CLI. Per ulteriori informazioni, consulta Modifica delle impostazioni dell'account Amazon ECS.

Se disponi di un processo in esecuzione che utilizza la modalità di rete `awsvpc` in una sottorete abilitata per IPv6 tra le date del 1° ottobre 2020 e del 2 novembre 2020, l'impostazione account `dualStackIPv6` di default nella regione in cui il processo 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`.

Conformità FIPS-140 di Fargate

Nome risorsa: `fargateFIPSMode`

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 valore predefinito è `disabled`.

È necessario attivare la conformità FIPS-140. 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.

Autorizzazione della risorsa tag

Nome risorsa: `tagResourceAuthorization`

Alcune operazioni API di Amazon ECS consentono di specificare tag quando crei le risorse.

Amazon ECS sta introducendo l'autorizzazione all'assegnazione di tag per la creazione di risorse. Gli utenti devono disporre delle autorizzazioni per le azioni che creano una risorsa, ad esempio. `ecsCreateCluster` Se i tag sono specificati nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva sull'azione per verificare se gli utenti o i ruoli dispongono delle `ecs:TagResource` autorizzazioni per creare tag. 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".

Periodo di attesa per il ritiro delle attività Fargate

Nome risorsa: `fargateTaskRetirementWaitPeriod`

AWS è responsabile dell'applicazione di patch e della manutenzione dell'infrastruttura sottostante per AWS Fargate. Quando si AWS determina che è necessario un aggiornamento della sicurezza o dell'infrastruttura per un'attività Amazon ECS ospitata su Fargate, le attività devono essere interrotte e avviate nuove attività per sostituirle. Puoi configurare il periodo di attesa prima che le attività vengano ritirate per l'applicazione di patch. È possibile ritirare l'attività immediatamente, attendere 7 giorni di calendario o attendere 14 giorni di calendario.

Questa impostazione si applica a livello di account.

Attivazione del monitoraggio del runtime

Nome risorsa: `guardDutyActivate`

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 Proteggere i carichi di lavoro Amazon ECS con Runtime Monitoring.

Argomenti

Amazon Resource Name (ARN) e ID

Quando vengono create le risorse Amazon ECS, a ogni risorsa viene assegnato un Amazon Resource Name (ARN) e un identificatore di risorsa (ID). Se stai utilizzando uno strumento a riga di comando o l'API Amazon ECS in combinazione con Amazon ECS, l'uso degli ARN e degli ID è necessario per determinati comandi. 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.

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 l'attivazione, tutte le nuove risorse create utilizzeranno il nuovo formato.

> **Note**
>
> Un ID risorsa non può essere modificato dopo la sua creazione. Pertanto, il consenso o il rifiuto esplicito del nuovo formato non influisce sugli ID risorsa esistenti.

Le seguenti sezioni descrivono in che modo i formati dell'ARN e dell'ID risorsa stanno cambiando. Per ulteriori informazioni sulla transizione ai nuovi formati, consulta Domande frequenti su Amazon Elastic Container Service.

Formato Amazon Resource Name (ARN)

Alcune risorse hanno un nome descrittivo, ad esempio un servizio denominato `production`. In altri casi, è necessario specificare una risorsa utilizzando il formato dell'Amazon Resource Name (ARN). Il nuovo formato dell'ARN per processi, servizi e istanze di container Amazon ECS include il nome del cluster. Per informazioni sull'attivazione con il formato dell'ARN, consulta Modifica delle impostazioni dell'account Amazon ECS.

La tabella seguente mostra sia il formato attuale (precedente) sia il nuovo formato per ogni tipo di risorsa.

Tipo di risorsa	ARN
Istanza del container	`arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:container-instance/` *`container-instance-id`* correnti Nuovo: `arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:container-instance/` *`cluster-name`* `/`*`container-instance-id`*
Servizio Amazon ECS	`arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:service/` *`service-name`* correnti Nuovo: `arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:service/` *`cluster-name`* `/`*`service-name`*
Processo di Amazon ECS	`arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:task/`*`task-id`* correnti Nuovo: `arn:aws:ecs:` *`region`*`:`*`aws_account_id`* `:task/`*`cluster-name`* `/`*`task-id`*

Lunghezza dell'ID risorsa

Un ID risorsa è formato da una combinazione univoca di lettere e numeri. I nuovi formati dell'ID risorsa includono ID più brevi per processi e istanze di container Amazon ECS. Il formato dell'ID risorsa precedente è lungo 36 caratteri. I nuovi ID vengono forniti 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.

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.

AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)

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”.

Esegui `put-account-setting-default` con l'opzione `fargateFIPSMode` impostata su `enabled`. Per ulteriori informazioni, consulta put-account-setting-default nella Documentazione di riferimento delle API di Amazon Elastic Container Service.

- È possibile utilizzare il seguente comando per attivare la conformità FIPS-140.

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

Output di esempio

```
{
    "setting": {
        "name": "fargateFIPSMode",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:root",
         "type": user
    }
}
```

Puoi eseguire il comando `list-account-settings` per visualizzare lo stato di conformità FIPS-140 corrente. Utilizza l'opzione `effective-settings` per visualizzare le impostazioni a livello di account.

```
aws ecs list-account-settings --effective-settings
```

Autorizzazione all'assegnazione di tag

Amazon ECS sta introducendo l'autorizzazione all'assegnazione di tag per la creazione di risorse. Gli utenti devono disporre delle autorizzazioni di etichettatura per le azioni che creano la risorsa, ad esempio. `ecsCreateCluster` Quando create una risorsa e specificate i tag per quella risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vi siano le autorizzazioni per creare tag. Pertanto, devi concedere le autorizzazioni esplicite per utilizzare l'operazione `ecs:TagResource`. Per ulteriori informazioni, consulta the section called “Assegnazione di tag alle risorse al momento della creazione”.

Per attivare l'autorizzazione all'assegnazione di tag, esegui il comando `put-account-setting-default` con l'opzione `tagResourceAuthorization` impostata su `enable`. Per ulteriori informazioni, consulta put-account-setting-default nella Documentazione di riferimento delle API di Amazon Elastic Container Service. Puoi eseguire il comando `list-account-settings` per visualizzare lo stato attuale dell'autorizzazione all'assegnazione di tag.

- È possibile utilizzare il comando seguente per abilitare l'autorizzazione all'etichettatura.

```
aws ecs put-account-setting-default --name tagResourceAuthorization --value on --
region region
```

Output di esempio

```
{
    "setting": {
        "name": "tagResourceAuthorization",
        "value": "on",
        "principalArn": "arn:aws:iam::123456789012:root",
        "type": user
    }
}
```

Dopo aver abilitato l'autorizzazione all'etichettatura, è necessario configurare le autorizzazioni appropriate per consentire agli utenti di taggare le risorse al momento della creazione. Per ulteriori informazioni, consulta the section called “Assegnazione di tag alle risorse al momento della creazione”.

Puoi eseguire il comando `list-account-settings` per visualizzare lo stato attuale dell'autorizzazione all'assegnazione di tag. Utilizza l'opzione `effective-settings` per visualizzare le impostazioni a livello di account.

```
aws ecs list-account-settings --effective-settings
```

Cronologia dell'autorizzazione all'assegnazione di tag

Puoi confermare se l'autorizzazione all'assegnazione di tag è attiva eseguendo il comando `list-account-settings` per visualizzare il valore di `tagResourceAuthorization`. Quando il valore è on, indica che l'autorizzazione all'assegnazione di tag è in uso. Per ulteriori informazioni, consulta list-account-setting nella Documentazione di riferimento delle API di Amazon Elastic Container Service.

Di seguito sono riportate le date importanti correlate all'autorizzazione all'assegnazione di tag.

- 18 aprile 2023: introduzione dell'autorizzazione all'assegnazione di tag. Tutti gli account nuovi ed esistenti devono aderire per utilizzare la funzionalità. Puoi scegliere di iniziare a utilizzare l'autorizzazione per l'etichettatura. Effettuando l'adesione, devi concedere le autorizzazioni appropriate.
- 9 febbraio 2024 - 6 marzo 2024: per tutti i nuovi account e per gli account esistenti non interessati è attivata l'autorizzazione all'etichettatura per impostazione predefinita. Puoi abilitare o disabilitare l'impostazione dell'`tagResourceAuthorization`account per verificare la tua politica IAM.

  AWS ha notificato gli account interessati.

  Per disabilitare la funzionalità, eseguila `put-account-setting-default` con l'`tagResourceAuthorization`opzione impostata su. `off`
- 7 marzo 2024: se hai abilitato l'autorizzazione all'etichettatura, non puoi più disabilitare l'impostazione dell'account.

  Ti consigliamo di completare i test delle policy IAM prima di questa data.
- 29 marzo 2024: tutti gli account utilizzano l'autorizzazione all'etichettatura. L'impostazione a livello di account non sarà più disponibile nella console Amazon ECS o. AWS CLI

AWS Fargate tempo di attesa per il ritiro dell'attività

AWS invia notifiche quando le attività di Fargate sono in esecuzione su una revisione della versione della piattaforma contrassegnata come ritirata. Per ulteriori informazioni, consulta AWS Domande frequenti sulla manutenzione delle attività di Fargate su Amazon ECS.

Puoi configurare l'ora in cui Fargate avvia il ritiro dell'attività. Per i carichi di lavoro che richiedono l'applicazione immediata degli aggiornamenti, scegli l'impostazione immediata (`0`). Quando hai bisogno di un maggiore controllo, ad esempio, quando un'attività può essere interrotta solo durante una determinata finestra, configura l'opzione 7 (7) o 14 giorni (14).

Consigliamo di scegliere un periodo di attesa più breve per ricevere prima le revisioni delle versioni più recenti della piattaforma.

Configura il periodo di attesa eseguendo `put-account-setting-default` o `put-account-setting` come utente root o utente amministrativo. Utilizza l'opzione `fargateTaskRetirementWaitPeriod` per il `name` e l'opzione `value` impostata su uno dei seguenti valori:

- `0`- AWS invia la notifica e inizia immediatamente a ritirare le attività interessate.
- `7`- AWS invia la notifica e attende 7 giorni di calendario prima di iniziare a ritirare le attività interessate.
- `14`: AWS invia la notifica e attende 14 giorni di calendario prima di iniziare a ritirare le attività interessate.

L'impostazione di default è 7 giorni.

Per ulteriori informazioni, consulta put-account-setting-default e put-account-setting nella Documentazione di riferimento delle API di Amazon Elastic Container Service.

È possibile eseguire il comando seguente per impostare il periodo di attesa su 14 giorni.

```
aws ecs put-account-setting-default --name fargateTaskRetirementWaitPeriod --value 14
```

Output di esempio

```
{
```

```
    "setting": {
        "name": "fargateTaskRetirementWaitPeriod",
        "value": "14",
        "principalArn": "arn:aws:iam::123456789012:root",
        "type: user"
    }
}
```

Puoi eseguire il comando `list-account-settings` per visualizzare il tempo di attesa corrente per il ritiro delle attività di Fargate. Utilizza l'opzione `effective-settings`.

```
aws ecs list-account-settings --effective-settings
```

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 istanze di container Fargate ed EC2 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 Proteggere i carichi di lavoro Amazon ECS con Runtime Monitoring.

Puoi eseguire `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"
    }
}
```

Visualizzazione delle impostazioni dell'account Amazon ECS tramite la console

Puoi usare il AWS Management Console per visualizzare le impostazioni del tuo account.

> **Important**
>
> Le impostazioni dell'account `dualStackIPv6`, `fargateFIPSMode` e `fargateTaskRetirementWaitPeriod` possono essere visualizzate o modificate solo utilizzando la AWS CLI.

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
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

Puoi utilizzarli AWS Management Console per modificare le impostazioni del tuo account.

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 Proteggere i carichi di lavoro Amazon ECS con Runtime Monitoring.

> **Important**
>
> Le impostazioni dell'account `dualStackIPv6`, `fargateFIPSMode` e `fargateTaskRetirementWaitPeriod` possono essere visualizzate o modificate solo utilizzando la AWS CLI.

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella barra di navigazione nella parte superiore, seleziona la Regione per la quale visualizzare le impostazioni dell'account.

3. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).
4. Scegli Aggiorna.
5. Per aumentare o diminuire il numero di attività che puoi eseguire nella modalità di rete awsvpc per ogni istanza EC2, in AWSVPC Trunking, seleziona Trunking. AWSVPC
6. Per utilizzare o interrompere l'utilizzo predefinito di CloudWatch Container Insights per i cluster, in Container Insights, seleziona o deseleziona CloudWatch Container Insights. CloudWatch
7. Per abilitare o disabilitare l'autorizzazione all'etichettatura delle risorse, in Autorizzazione di etichettatura delle risorse, seleziona o deseleziona l'autorizzazione all'etichettatura delle risorse.
8. Seleziona Salvataggio delle modifiche.
9. Nella schermata di conferma, scegliere Conferma per salvare la selezione.

Ripristino alle impostazioni predefinite dell'account Amazon ECS

Puoi utilizzare il AWS Management Console per ripristinare le impostazioni predefinite del tuo account Amazon ECS.

L'opzione Revert to account default (Ripristina valori predefiniti account) è disponibile solo quando le impostazioni dell'account non sono più le impostazioni predefinite.

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella barra di navigazione nella parte superiore, seleziona la Regione per la quale visualizzare le impostazioni dell'account.
3. Nel riquadro di navigazione, scegli Account Settings (Impostazioni account).
4. Scegli Aggiorna.
5. Scegli Revert to account default (Ripristina valori predefiniti account).
6. Nella schermata di conferma, scegli Conferma per salvare la selezione.

Gestione delle impostazioni dell'account Amazon ECS utilizzando AWS CLI

Puoi gestire le impostazioni del tuo account utilizzando l'API Amazon ECS AWS CLI o gli SDK. Le `dualStackIPv6` impostazioni `fargateFIPSMode` e le impostazioni dell'`fargateTaskRetirementWaitPeriod`account possono essere visualizzate o modificate solo utilizzando tali strumenti.

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 sovrascriva 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-ECS () AccountSettingAWS 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.

- Write-ECS () AccountSettingAWS 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.

- Write-ECS () AccountSettingAWS Tools for Windows PowerShell

```
Write-ECSAccountSetting -Name serviceLongArnFormat -Value enabled -PrincipalArn
 arn:aws:iam::aws_account_id:user/principalName -Region us-east-1 -Force
```

Ruoli IAM per Amazon ECS

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità utilizzate. Utilizza la tabella seguente per determinare quali ruoli IAM sono necessari per Amazon ECS.

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri AWS servizi per tuo conto.	La tua attività è ospitata su AWS Fargateo su istanze esterne e: • estrae un'immagine del contenitore da un repositor	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		y privato Amazon ECR. • estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività. • invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. `awslogs` La tua attività è ospitata su una delle AWS Fargate istanze Amazon EC2 e: • utilizza l'autenticazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di Secrets Manager o i parametri AWS	

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		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 istanze EC2 o alle istanze esterne di registrarsi nel cluster.	La tua attività è ospitata su istanze Amazon EC2 o su un'istanza esterna.	Ruolo IAM delle istanze di container Amazon ECS
Ruolo di Amazon ECS Anywhere	Questo ruolo consente alle istanze esterne di accedere AWS alle API.	La tua attività è ospitata su istanze esterne.	Ruolo IAM di Amazon ECS Anywhere
Ruolo di Amazon ECS CodeDeploy	Questo ruolo consente di CodeDeploy apportare aggiornamenti ai 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 di EventBridge apportare aggiornamenti ai tuoi servizi.	Utilizzi le EventBridge regole e gli obiettivi per pianificare le tue attività.	Ruolo EventBridge IAM di Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo dell'infrastruttura Amazon ECS	Questo ruolo consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• Vuoi collegare i volumi Amazon EBS alle tue attività Amazon ECS di tipo Fargate o EC2. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività. • Desideri utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect.	Ruolo IAM dell'infrastruttura Amazon ECS

Definizioni dei processi di Amazon ECS

Una definizione di attività è uno schema dell'applicazione. Si tratta di un file di testo in formato JSON che descrive i parametri e uno o più container che formano l'applicazione.

Alcuni dei parametri che puoi specificare in una definizione di attività includono:

- Il tipo di avvio da utilizzare, che determina l'infrastruttura in cui vengono ospitate le attività
- Immagine Docker da utilizzare con ogni container nel processo
- La quantità di CPU e di memoria da utilizzare con ogni processo o container all'interno di un processo
- I requisiti di memoria e CPU
- Il sistema operativo del contenitore su cui viene eseguita l'attività
- La modalità di rete Docker da utilizzare per i container nel tuo processo
- La configurazione di registrazione da utilizzare per i tuoi processi
- La possibilità che l'attività continui a essere eseguita in caso di interruzione o esito negativo del container
- Il comando che il container deve eseguire all'avvio
- Tutti i volumi di dati utilizzati con il container nell'attività
- Il ruolo IAM utilizzato dalle attività

Per un elenco completo dei parametri per la definizione di attività, consulta Parametri di definizione delle attività di Amazon ECS.

Dopo aver creato una definizione di attività, è possibile eseguirla come attività o servizio.

- Si definisce attività la creazione dell'istanza relativa a una definizione di attività all'interno di un cluster. Dopo aver creato una definizione di attività per la tua applicazione all'interno di Amazon ECS, puoi specificare il numero di processi che saranno eseguiti sul tuo cluster.
- un servizio Amazon ECS esegue e mantiene simultaneamente il numero desiderato di attività in un cluster Amazon ECS. Il funzionamento è che, se uno dei processi non riesce o si interrompe per qualsiasi motivo, il pianificatore del servizio Amazon ECS lancia un'altra istanza in base alla definizione di attività. Lo fa per sostituirlo e quindi mantenere il numero desiderato di processi nel servizio.

Argomenti

Stati di definizione delle attività di Amazon ECS

Una definizione di attività cambia lo stato quando la crei, annulli la registrazione o la elimini. È possibile visualizzare lo stato di definizione dell'attività nella console o utilizzando. `DescribeTaskDefinition`

Di seguito sono riportati i possibili stati di una definizione di attività.

ACTIVE

Una definizione di attività diventa `ACTIVE` una volta che è stata registrata con Amazon ECS. È possibile utilizzare le definizioni di attività nello stato `ACTIVE` per eseguire attività o creare servizi.

INACTIVE

Una definizione di attività passa dallo stato `ACTIVE` allo stato `INACTIVE` quando si annulla la registrazione di una definizione di attività. È possibile recuperare un definizione di attività `INACTIVE` attraverso la chiamata `DescribeTaskDefinition`. Non è possibile eseguire nuove attività o creare nuovi servizi con una definizione di attività nello stato `INACTIVE`. I servizi e le attività esistenti non subiscono alcun impatto.

DELETE_IN_PROGRESS

Una definizione di attività passa dallo stato `INACTIVE` allo stato `DELETE_IN_PROGRESS` dopo la richiesta di eliminazione della definizione di attività. Dopo che la definizione dell'attività

ha raggiunto lo stato `DELETE_IN_PROGRESS`, Amazon ECS verifica periodicamente che la definizione dell'attività di destinazione non sia referenziata da alcuna attività o implementazione attiva e successivamente elimina in via definitiva la definizione dell'attività. Non è possibile eseguire nuove attività o creare nuovi servizi con una definizione di attività nello stato `DELETE_IN_PROGRESS`. Una definizione di attività può essere inviata per l'eliminazione in qualsiasi momento senza influire sulle attività e sui servizi esistenti.

Le definizioni di attività nello stato `DELETE_IN_PROGRESS` sono visibili nella console ed è possibile recuperarle effettuando una chiamata `DescribeTaskDefinition`.

Quando elimini tutte le revisioni delle definizioni di attività nello stato `INACTIVE`, il nome della definizione di attività non viene visualizzato nella console e non viene restituito nell'API. Se `DELETE_IN_PROGRESS` lo stato di revisione della definizione dell'attività è attivo, il nome della definizione dell'attività viene visualizzato nella console e restituito nell'API. Il nome della definizione di attività viene mantenuto da Amazon ECS e la revisione viene incrementata durante la prossima creazione di una definizione di attività con tale nome.

Se lo utilizzi AWS Config per gestire le definizioni delle attività, ti AWS Config addebita tutte le registrazioni delle definizioni delle attività. Viene addebitato solo l'annullamento della registrazione della definizione di attività `ACTIVE` più recente. L'eliminazione di una definizione di attività non prevede alcun costo. Per ulteriori informazioni sui prezzi, consulta Prezzi di AWS Config.

Risorse Amazon ECS in grado di bloccare un'eliminazione

Una richiesta di eliminazione della definizione di attività non viene completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione dell'attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività di Amazon ECS: la definizione di attività è necessaria affinché l'attività rimanga integra.
- Implementazioni e set di attività 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, puoi 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 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:

- Il tipo di avvio da utilizzare, che determina l'infrastruttura su cui sono ospitate le attività.

  Quando utilizzi il tipo di avvio EC2, scegli anche il tipo di istanza. Per alcuni tipi di istanza, come la GPU, devi impostare parametri aggiuntivi. Per ulteriori informazioni, consulta Casi d'uso per la definizione delle attività di Amazon ECS.
- L'immagine del contenitore, che contiene il codice dell'applicazione e tutte le dipendenze necessarie per l'esecuzione del codice dell'applicazione.
- La modalità di rete da utilizzare per i contenitori coinvolti nell'attività

  La modalità di rete determina il modo in cui l'attività comunica sulla rete.

  Per le attività eseguite su un'istanza EC2, esistono diverse opzioni, ma consigliamo di utilizzare la modalità di `awsvpc` rete. La modalità `awsvpc` di rete semplifica il networking in container, poiché hai un maggiore controllo sul modo in cui le applicazioni comunicano tra loro e con altri servizi all'interno dei tuoi VPC.

  Per le attività eseguite su Fargate, è possibile utilizzare solo la modalità di `awsvpc` rete.
- La configurazione di registrazione da utilizzare per le tue attività.
- Qualsiasi volume di dati utilizzato con i contenitori dell'attività.

Per un elenco completo dei parametri per la definizione di attività, consulta Parametri di definizione delle attività di Amazon ECS.

Utilizza le linee guida seguenti per creare le definizioni di attività:

- Utilizza ogni famiglia di definizioni delle attività per un solo scopo aziendale.

  Se raggruppi più tipi di container delle applicazioni nella stessa definizione di attività, non puoi dimensionare tali container in modo indipendente. Ad esempio, è improbabile che un sito Web e un'API richiedano l'impiego della scalabilità orizzontale alla stessa velocità. Con l'aumento del traffico, sarà necessario un numero diverso di container Web rispetto ai container API. Se questi due container vengono implementati nella stessa definizione di attività, ogni attività esegue lo stesso numero di container Web e container API.
- Associa ogni versione dell'applicazione a una revisione della definizione di attività all'interno di una famiglia di definizioni delle attività.

  All'interno di una famiglia di definizioni delle attività, considera ogni revisione della definizione di attività come uno snapshot point-in-time delle impostazioni per una particolare immagine di container. È simile al modo in cui il container può essere considerato uno snapshot di tutti gli elementi necessari per eseguire una particolare versione del codice dell'applicazione.

  Assicurati che esista una one-to-one mappatura tra una versione del codice dell'applicazione, un tag di immagine del contenitore e una revisione della definizione dell'attività. Un tipico processo di rilascio prevede un commit git che viene trasformato in un'immagine di container con il tag SHA del commit git. Quindi, il tag dell'immagine di container riceve la propria revisione della definizione di attività di Amazon ECS. Infine, il servizio Amazon ECS viene aggiornato per ordinare l'implementazione della nuova revisione della definizione di attività.
- Utilizza ruoli IAM diversi per ogni famiglia di definizione delle attività.

  Definisci ogni definizione delle attività con il proprio ruolo IAM. Questa raccomandazione dovrebbe essere seguita congiuntamente al consiglio di fornire a ogni componente aziendale la propria famiglia di definizione delle attività. Implementando entrambe queste best practice, puoi limitare l'accesso di ciascun servizio alle risorse del tuo AWS account. Ad esempio, puoi concedere al servizio di autenticazione l'accesso per connettersi al database delle password. Allo stesso tempo, puoi garantire l'accesso alle informazioni di pagamento con carta di credito soltanto al servizio.

Le migliori pratiche per le immagini dei container Amazon ECS

Un'immagine del container è un insieme di istruzioni su come creare il container. Un'immagine del container contiene il codice dell'applicazione e tutte le dipendenze necessarie per l'esecuzione di tale codice. Le dipendenze dell'applicazione includono i pacchetti di codice sorgente su cui si basa il

codice dell'applicazione, un runtime del linguaggio per i linguaggi interpretati e i pacchetti binari su cui si basa il codice collegato dinamicamente.

Utilizza le linee guida seguenti per progettare e creare le immagini dei container:

- Rendi complete le immagini del container, archiviando tutte le dipendenze dell'applicazione come file statici all'interno dell'immagine del container.

  Se modifichi qualcosa nell'immagine del contenitore, crea una nuova immagine del contenitore con le modifiche.
- Esegui un singolo processo di applicazione all'interno di un container.

  La durata del container è pari a quella del processo di applicazione. Amazon ECS sostituisce i processi bloccati e determina dove avviare il processo di sostituzione. Un'immagine completa rende l'implementazione complessiva più resiliente.
- Fai in modo che sia la tua applicazione a gestire. `SIGTERM`

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

  Devi identificare il tempo impiegato dall'applicazione per completare il suo lavoro e assicurarti che le applicazioni gestiscano il segnale. `SIGTERM` La gestione del segnale dell'applicazione deve impedire all'applicazione di eseguire nuovi lavori e completare il lavoro in corso, oppure salvare il lavoro incompiuto in un archivio esterno all'attività quando il completamento del lavoro richiede troppo tempo.
- Configura le applicazioni containerizzate per la scrittura di log su `stdout` e `stderr`.

  Il disaccoppiamento della gestione dei log dal codice dell'applicazione offre la flessibilità necessaria per adattare la gestione dei log a livello di infrastruttura. Un esempio di ciò è la modifica del sistema di registrazione. Invece di modificare i servizi e creare e distribuire una nuova immagine del contenitore, puoi modificare le impostazioni.
- Usa i tag per controllare le versioni delle immagini dei container.

  Le immagini dei container vengono archiviate in un registro dei container. Ogni immagine all'interno del registro è identificata da un tag. È presente un tag denominato `latest`. Questo tag funge da puntatore verso la versione più recente dell'immagine del container dell'applicazione, simile

al comando HEAD in un repository git. Ti consigliamo di utilizzare il tag `latest` solo a scopo di test. Come best practice, assegna alle immagini del container un tag univoco per ogni build. Ti consigliamo di assegnare i tag alle immagini tramite SHA git per il commit git usato per creare l'immagine.

Non è necessario creare un'immagine di container per ogni commit. Tuttavia, ti consigliamo di creare una nuova immagine del container ogni volta che rilasci un particolare commit di codice nell'ambiente di produzione. Ti consigliamo inoltre di assegnare all'immagine un tag che corrisponda al commit git del codice presente nell'immagine. Se hai assegnato all'immagine un commit git, puoi trovare più rapidamente la versione del codice in esecuzione sull'immagine.

Ti consigliamo inoltre di attivare i tag di immagine immutabili in Amazon Elastic Container Registry. Con questa impostazione, non puoi modificare l'immagine del container a cui punta un tag. Amazon ECR impone invece che una nuova immagine debba essere caricata su un nuovo tag. Per ulteriori informazioni, consulta Mutabilità del tag immagine nella Guida per l'utente di Amazon ECR.

Quando si progetta l'applicazione in modo che venga eseguita AWS Fargate, è necessario decidere se distribuire più contenitori nella stessa definizione di attività e distribuire contenitori separatamente in più definizioni di attività. Quando sono necessarie le seguenti condizioni, si consiglia di implementare più container nella stessa definizione di attività:

- I container condividono un ciclo di vita comune (ovvero vengono avviati e terminati contemporaneamente).
- I container devono essere eseguiti sullo stesso host sottostante (ovvero, un container che fa riferimento all'altro su una porta localhost).
- I 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 contenitori separatamente.

Le migliori pratiche per le dimensioni delle attività di Amazon ECS

Una delle scelte più importanti da fare quando si distribuiscono container su Amazon ECS è la dimensione dei container e delle attività. Le dimensioni dei contenitori e delle attività sono entrambe essenziali per la scalabilità e la pianificazione della capacità. In Amazon ECS, vengono utilizzati due

parametri di risorse per la capacità: CPU e memoria. La CPU viene misurata in unità di 1/1024 di una vCPU completa (dove 1024 unità equivalgono a 1 vCPU intera). La memoria viene misurata in megabyte. Nella definizione dell'attività, è possibile dichiarare riserve e limiti di risorse.

Quando si dichiara una prenotazione, si dichiara la quantità minima di risorse richiesta da un'attività. L'attività riceve almeno la quantità di risorse richieste. L'applicazione potrebbe essere in grado di utilizzare più CPU o memoria rispetto alla prenotazione dichiarata. Tuttavia, ciò è soggetto ai limiti che hai anche dichiarato. L'utilizzo di un importo superiore all'importo della prenotazione è noto come bursting. In Amazon ECS, le prenotazioni sono garantite. Ad esempio, se utilizzi istanze Amazon EC2 per fornire capacità, Amazon ECS non colloca un'attività su un'istanza in cui la prenotazione non può essere soddisfatta.

Un limite è la quantità massima di unità CPU o memoria che il contenitore o l'attività può utilizzare. Qualsiasi tentativo di utilizzare una quantità di CPU superiore a questo limite comporta un rallentamento. Qualsiasi tentativo di utilizzare più memoria comporta l'interruzione del contenitore.

La scelta di questi valori può essere difficile. Questo perché i valori più adatti all'applicazione dipendono in larga misura dai requisiti di risorse dell'applicazione. Il test di carico dell'applicazione è la chiave per una corretta pianificazione del fabbisogno di risorse e una migliore comprensione dei requisiti dell'applicazione.

Applicazioni stateless

Per le applicazioni stateless con scalabilità orizzontale, ad esempio un'applicazione con sistema di bilanciamento del carico, consigliamo innanzitutto di determinare la quantità di memoria che l'applicazione consuma per soddisfare le richieste. A tale scopo, è possibile utilizzare strumenti tradizionali come `ps` or o soluzioni di monitoraggio come `top` Container Insights. CloudWatch

Nel determinare una prenotazione della CPU, considerate come volete scalare l'applicazione per soddisfare i requisiti aziendali. È possibile utilizzare riserve di CPU più piccole, ad esempio 256 unità CPU (o 1/4 vCPU), per scalare orizzontalmente in modo da ridurre al minimo i costi. Tuttavia, potrebbero non scalare abbastanza velocemente per soddisfare i picchi significativi della domanda. È possibile utilizzare prenotazioni di CPU più grandi per scalare più rapidamente e quindi soddisfare più rapidamente i picchi di domanda. Tuttavia, le prenotazioni di CPU più grandi sono più costose.

Altre applicazioni

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

consigliabile scegliere la quantità di memoria e CPU in base a ciò che i test di carico indicano che è necessario servire il traffico per raggiungere l'obiettivo del livello di servizio. Amazon ECS garantisce che l'applicazione sia collocata su un host con una capacità adeguata.

Best practice di sicurezza di rete per Amazon ECS

La sicurezza di rete è un argomento ampio che comprende diversi argomenti secondari. Queste includono segmentazione e isolamento della rete encryption-in-transit, firewall, routing del traffico e osservabilità.

Crittografia in transito

La crittografia del traffico di rete impedisce agli utenti non autorizzati di intercettare e leggere i dati trasmessi attraverso una rete. Con Amazon ECS, la crittografia di rete può essere implementata in uno qualsiasi dei modi descritti di seguito.

- Con una mesh di servizi (TLS):

  Con AWS App Mesh, puoi configurare le connessioni TLS tra i proxy Envoy distribuiti con endpoint mesh. Due esempi sono i nodi virtuali e i gateway virtuali. I certificati TLS possono provenire da (ACM). AWS Certificate Manager In alternativa, possono provenire dalla tua personale autorità di certificazione privata.

  - Abilitazione di Transport Layer Security (TLS)
  - Abilita la crittografia del traffico tra i servizi AWS App Mesh utilizzando certificati ACM o certificati forniti dal cliente
  - Procedura guidata ACM TLS
  - Procedura guidata per file TLS
  - Envoy
- Uso delle istanze Nitro:

  Per impostazione predefinita, il traffico viene crittografato in automatico tra i seguenti tipi di istanza Nitro: C5n, G4, I3en, M5dn, M5n, P3dn, R5dn e R5n. Il traffico non viene crittografato se instradato attraverso un gateway di transito, un sistema di bilanciamento del carico o un intermediario simile.

  - Crittografia in transito
  - Annuncio di novità del 2019
  - Questo intervento di RE:InForce 2019
- Uso dell'estensione Server Name Indication (SNI) con Application Load Balancer:

Application Load Balancer (ALB) e Network Load Balancer (NLB) supportano l'estensione Server Name Indication (SNI). Utilizzando SNI, puoi posizionare più applicazioni sicure dietro un unico ascoltatore. A tale scopo, ciascuna dispone di un suo certificato TLS. Consigliamo di effettuare il provisioning dei certificati per il sistema di bilanciamento del carico utilizzando AWS Certificate Manager (ACM) e di aggiungerli all'elenco di certificati dell'ascoltatore. Il AWS load balancer utilizza un algoritmo di selezione intelligente dei certificati con SNI. Se l'hostname fornito da un client corrisponde a un singolo certificato nell'elenco di certificati, il sistema di bilanciamento del carico seleziona tale certificato. Se un hostname fornito da un client corrisponde a più certificati nell'elenco di certificati, il sistema di bilanciamento del carico seleziona un certificato che il client è in grado di supportare. Alcuni esempi sono un certificato autofirmato o un certificato generato attraverso ACM.

  - SNI con Application Load Balancer
  - SNI con Network Load Balancer

- nd-to-end Crittografia E 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 Private Certificate Authority

Reti di attività

I seguenti suggerimenti prendono in considerazione il funzionamento di Amazon ECS. Amazon ECS non utilizza una rete di sovrapposizione. Le attività sono invece configurate per funzionare in diverse modalità di rete. Ad esempio, le attività configurate per l'uso della modalità `bridge` acquisiscono un indirizzo IP non instradabile da una rete Docker in esecuzione su ciascun host. Le attività configurate per l'uso della modalità di rete `awsvpc` acquisiscono un indirizzo IP dalla sottorete dell'host. Le attività configurate con reti `host` utilizzano l'interfaccia di rete dell'host. `awsvpc` è la modalità di rete preferita. Questo perché è l'unica modalità che puoi utilizzare per assegnare gruppi di sicurezza alle attività. È anche l'unica modalità disponibile per AWS Fargate le attività su Amazon ECS.

Gruppi di sicurezza per le attività

Consigliamo di configurare le attività per l'uso della modalità di rete `awsvpc`. Dopo aver configurato l'attività per l'uso di tale modalità, l'agente Amazon ECS effettua in automatico il provisioning e il collegamento di un'interfaccia di rete elastica (ENI) all'attività. Quando viene fornito l'ENI, l'attività viene registrata in un AWS gruppo di sicurezza. Il gruppo di sicurezza agisce da firewall virtuale che puoi utilizzare per il controllo del traffico in entrata e in uscita.

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 policy consentono di affinare l'accesso alle API di un servizio. Ad esempio, puoi creare una policy di endpoint per Amazon ECR che consenta solo di inviare immagini a registri in determinati account AWS . Una policy come questa potrebbe essere utilizzata per impedire che i dati vengano esfiltrati attraverso le immagini di container, pur consentendo agli utenti di effettuare l'invio ai registri Amazon ECR autorizzati. Per ulteriori informazioni, consulta Uso delle policy di endpoint VPC.

La seguente politica consente a tutti AWS i principali del tuo account di eseguire tutte le azioni solo sui tuoi repository Amazon ECR:

```
{
  "Statement": [
    {
      "Sid": "LimitECRAccess",
      "Principal": "*",
      "Action": "*",
      "Effect": "Allow",
```

```
      "Resource": "arn:aws:ecr:region:account_id:repository/*"
    },
  ]
}
```

Puoi migliorare ulteriormente quest'opzione impostando una condizione che utilizzi la nuova proprietà `PrincipalOrgID`. Ciò impedisce l'invio e l'estrazione di immagini da parte di un principale IAM che non fa parte del tuo. AWS Organizations Per ulteriori informazioni, consulta aws: PrincipalOrg ID.

Consigliamo di applicare la medesima policy sia agli endpoint `com.amazonaws.region.ecr.dkr` che agli endpoint `com.amazonaws.region.ecr.api`.

Impostazioni dell'agente Container

Il file di configurazione dell'agente di container Amazon ECS include diverse variabili di ambiente relative alla sicurezza di rete. `ECS_AWSVPC_BLOCK_IMDS` e `ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST` vengono utilizzate per bloccare l'accesso di un'attività ai metadati di Amazon EC2. `HTTP_PROXY` viene utilizzata per configurare l'agente in modo che effettui l'instradamento attraverso un proxy HTTP per la connessione a Internet. Per ricevere istruzioni sulla configurazione dell'agente e del runtime Docker per l'instradamento attraverso un proxy, consulta Configurazione del proxy HTTP.

> **Important**
>
> Queste impostazioni non sono disponibili quando utilizzi AWS Fargate.

Consigli sulla sicurezza della rete

Consigliamo di completare la procedura seguente quando configuri Amazon VPC, i sistemi di bilanciamento del carico e la rete.

Usa la crittografia di rete, ove applicabile, con Amazon ECS

È consigliabile utilizzare la crittografia di rete laddove applicabile. Alcuni programmi di conformità, come PCI DSS, richiedono la crittografia dei dati in transito se i dati contengono dati del titolare della carta. Se il tuo carico di lavoro presenta requisiti simili, configura la crittografia di rete.

I browser moderni avvisano gli utenti quando si connettono a siti non sicuri. Se il servizio è gestito da un sistema di bilanciamento del carico rivolto al pubblico, utilizza TLS/SSL per crittografare il

traffico dal browser del client al sistema di bilanciamento del carico e, se necessario, crittografarlo nuovamente nel backend.

Usa la modalità di **`awsvpc`** rete e i gruppi di sicurezza per controllare il traffico tra attività e altre risorse in Amazon ECS

È consigliabile utilizzare la modalità di rete `awsvpc` e i gruppi di sicurezza quando devi controllare il traffico tra le attività e tra le attività e altre risorse di rete. Se il tuo servizio è 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 VPC separati quando il traffico di rete deve essere strettamente isolato

È consigliabile creare cluster in Amazon VPC separati quando il traffico di rete deve essere sottoposto a un isolamento rigoroso. 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 isolamento della rete rigoroso, crea cluster in Amazon VPC separati ed esponi selettivamente i servizi ad altri Amazon VPC utilizzando gli endpoint Amazon VPC. Per ulteriori informazioni, consulta Endpoint Amazon VPC.

Configura AWS PrivateLink gli endpoint quando richiesto per Amazon ECS

È necessario configurare gli AWS PrivateLink endpoint quando necessario. Se la tua politica di sicurezza ti impedisce di collegare un Internet Gateway (IGW) ai tuoi Amazon VPC, configura gli AWS PrivateLink endpoint per Amazon ECS e altri servizi come Amazon ECR e Amazon. AWS Secrets Manager CloudWatch

Usa Amazon VPC Flow Logs per analizzare il traffico da e verso le attività a esecuzione prolungata in Amazon ECS

È consigliabile utilizzare Amazon VPC Flow Logs per analizzare il traffico da e verso le attività a esecuzione prolungata. Le attività che utilizzano la modalità di rete `awsvpc` ottengono la propria ENI. In tal modo, puoi monitorare il traffico da e verso le singole attività utilizzando Amazon VPC Flow Logs. Un recente aggiornamento di Amazon VPC Flow Logs (v3) arricchisce i log con metadati sul

traffico tra cui l'ID di vpc, l'ID di sottorete e l'ID dell'istanza. Questi metadati possono essere utilizzati per restringere un'indagine. Per ulteriori informazioni, consulta Amazon VPC Flow Logs.

> **Note**
>
> A causa della natura temporanea dei container, i log di flusso potrebbero non essere sempre un modo efficace per analizzare i pattern di traffico tra diversi container o tra container e altre risorse di rete.

Opzioni di task networking di Amazon ECS per il tipo di lancio EC2

Il comportamento della rete delle attività Amazon ECS ospitate su istanze Amazon EC2 dipende dalla modalità di rete definita nella definizione di 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	Container Linux su EC2	Container Windows su EC2	Descrizione
`awsvpc`	Sì	Sì	All'attività viene assegnata la propria interfaccia di rete elastica (ENI) e un indirizzo IPv4 privato primario. Ciò conferisce al processo le stesse proprietà di rete delle istanze Amazon EC2.
`bridge`	Sì	No	L'attività utilizza la rete virtuale integrata di Docker su Linux che viene eseguita all'interno di ogni istanza Amazon EC2 che ospita l'attività. La rete virtuale integrata su Linux utilizza il driver di rete `bridge` Docker. Questa è la modalità di rete predefinita su Linux se non viene specificata una modalità di rete nella definizione di attività.
`host`	Sì	No	L'attività utilizza la rete dell'host che ignora la rete virtuale integrata di Docker e mappa le

Modalità di rete	Container Linux su EC2	Container Windows su EC2	Descrizione
			porte del container direttamente all'interfaccia di rete elastica (ENI) dell'istanza Amazon EC2 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 sarà possibile eseguire più attività con la stessa definizione di attività su una singola istanza Amazon EC2.
`none`	Sì	No	L'attività non dispone di connettività di rete esterna.
`default`	No	Sì	L'attività utilizza la rete virtuale integrata di Docker che viene eseguita all'interno di ogni istanza Amazon EC2 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 definizio ne di attività.

Per ulteriori informazioni sulla rete Docker su Linux, consulta Panoramica della rete nella documentazione di Docker.

Per ulteriori informazioni sulla rete Docker su Windows, consulta Rete di container Windows nella documentazione di Microsoft Containers on Windows.

Assegna un'interfaccia di rete per un'attività Amazon ECS

Le funzionalità di rete delle attività fornite dalla modalità di rete `awsvpc` forniscono alle attività Amazon ECS le stesse proprietà di rete delle istanze Amazon EC2. L'utilizzo della modalità `awsvpc` di 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 all'interno dei tuoi VPC. La modalità `awsvpc` di rete offre inoltre una maggiore sicurezza per i container, consentendovi di utilizzare gruppi di sicurezza e strumenti di monitoraggio della rete a un livello più granulare nell'ambito delle vostre attività. Puoi anche utilizzare altre funzionalità di rete di Amazon EC2 come VPC Flow Logs per monitorare il traffico da e verso le tue attività. Inoltre, i container che appartengono alla stessa attività possono comunicare tramite l'interfaccia `localhost`.

La task elastic network interface (ENI) è una funzionalità completamente gestita di Amazon ECS. Amazon ECS crea l'ENI e la collega all'istanza Amazon EC2 dell'host con il gruppo di sicurezza specificato. Il processo invia e riceve il traffico di rete nell'ENI nello stesso modo con cui le istanze Amazon EC2 gestiscono le interfacce di rete primarie. Ad ogni ENI di processo viene assegnato un indirizzo IPv4 privato di default. Se il VPC è abilitato per la modalità dual-stack e si utilizza una sottorete con un blocco CIDR IPv6, anche l'ENI di processo riceverà un indirizzo IPv6. Ogni processo può avere una sola ENI.

Questi ENI sono visibili nella console Amazon EC2 del tuo account. Il tuo account non può scollegare o modificare gli ENI. 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'DescribeTasksAPI. Quando l'attività viene interrotta o se il servizio viene ridotto, l'ENI dell'attività viene scollegata ed eliminata.

Se hai bisogno di una maggiore densità ENI, utilizza le impostazioni dell'`awsvpcTrunking`account. Amazon ECS crea e collega anche un'interfaccia di rete «trunk» per l'istanza del contenitore. La rete trunk è completamente gestita da Amazon ECS. L'ENI trunk viene eliminata quando termini o annulli la registrazione dell'istanza di container dal cluster Amazon ECS. Per ulteriori informazioni sull'impostazione dell'`awsvpcTrunking`account, consulta. Prerequisiti

È necessario specificare `awsvpc` nel `networkMode` parametro della definizione dell'attività. Per ulteriori informazioni, consulta Modalità di rete.

Quindi, quando esegui un'attività o crei un servizio, utilizza il `networkConfiguration` parametro che include una o più sottoreti per collocare le attività in uno o più gruppi di sicurezza da collegare a un ENI. Per ulteriori informazioni, consulta Configurazione della rete. Le attività vengono posizionate

nelle istanze Amazon EC2 compatibili nelle stesse zone di disponibilità delle sottoreti e i gruppi di sicurezza specificati vengono associati all'ENI assegnata all'attività.

Considerazioni per Linux

Tieni in considerazione le informazioni seguenti durante l'utilizzo del sistema operativo Linux.

- Se utilizzi un'istanza p5.48xlarge in `awsvpc` modalità, non puoi eseguire più di un'attività sull'istanza.
- Le attività e i servizi che utilizzano la modalità di `awsvpc` rete richiedono il ruolo collegato ai servizi Amazon ECS per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS. Puoi anche creare il ruolo collegato al servizio con il seguente comando: AWS CLI

  ```
  aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
  ```

- L'istanza Linux di Amazon EC2 richiede la versione `1.15.0` o successiva dell'agente del container per eseguire processi che utilizzano la modalità di rete `awsvpc`. 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 processo Amazon ECS che utilizza la modalità di rete `awsvpc` riceve la propria interfaccia di rete elastica (ENI), che è collegata all'istanza Amazon EC2 che la ospita. Esiste una quota predefinita per il numero di interfacce di rete che possono essere collegate a un'istanza Linux di Amazon EC2. L'interfaccia di rete principale viene calcolata come unica rispetto a quella quota. Ad esempio, per impostazione predefinita, a un'istanza `c5.large` possono essere collegate fino a tre ENI. L'interfaccia di rete principale per l'istanza viene calcolata come unica. Puoi collegare altre due ENI all'istanza. Poiché ogni attività che utilizza la modalità di rete `awsvpc` richiede un'ENI, in genere puoi eseguire solo due attività su questo tipo di istanza. Per ulteriori informazioni sui limiti ENI predefiniti per ogni tipo di istanza, consulta Indirizzi IP per interfaccia di rete per tipo di istanza nella Amazon EC2 User Guide.

- Amazon ECS supporta l'avvio di istanze Linux di Amazon EC2 che utilizzano tipi di istanze supportati con densità ENI aumentata. Quando decidi di attivare l'impostazione dell'account `awsvpcTrunking` e registri le istanze Linux di Amazon EC2 che utilizzano questi tipi di istanza nel cluster, queste istanze hanno limiti ENI più elevati. L'utilizzo di queste istanze con questa quota più elevata significa che puoi eseguire più attività su ciascuna istanza Linux di Amazon EC2. Per utilizzare la densità ENI aumentata con la funzionalità di trunking, le istanze Amazon EC2 devono utilizzare la versione `1.28.1` o successiva dell'agente del container. Se utilizzi l'AMI ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione `1.28.1-2` del pacchetto `ecs-init`. Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsvpcTrunking`, consulta Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account. Per ulteriori informazioni su trunking dell'ENI, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.
- Quando si ospitano attività che utilizzano la modalità di rete `awsvpc` sulle istanze Linux di Amazon EC2, le ENI di attività non ricevono 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 le ENI che collega alle istanze Linux di Amazon EC2. Se hai collegato manualmente le ENI alle tue istanze, Amazon ECS potrebbe tentare di aggiungere un'attività a un'istanza che non dispone di adattatori di rete sufficienti. Ciò può comportare il timeout dell'attività e il passaggio a uno stato di deprovisioning e quindi allo stato di arresto. Ti consigliamo di non collegare manualmente le ENI alle istanze.
- Le istanze Linux di Amazon EC2 devono essere registrate con la funzionalità `ecs.capability.task-eni` per essere considerate per il posizionamento di processi con la modalità di rete `awsvpc`. Le istanze che eseguono la versione `1.15.0-4` o successiva di `ecs-init` vengono registrate automaticamente con questo attributo.
- Le ENI che vengono create e collegate alle tue istanze Linux di Amazon EC2 non possono essere scollegate manualmente o modificate dall'account. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare le ENI per un'attività, interrompi l'attività.
- 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 AwsVpcConfigurazione nel riferimento 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 i processi che utilizzano la modalità di rete `awsvpc` sono associate a un'ENI e non a un'istanza Linux di Amazon EC2. 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 le AMI Windows Server 2016 ottimizzate per Amazon ECS e AMI Windows che supportano 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 le modalità di rete `awsvpc`.
- L'istanza Windows di Amazon EC2 richiede una versione `1.57.1` o successiva dell'agente contenitore per utilizzare i CloudWatch parametri per i contenitori Windows che utilizzano la `awsvpc` modalità di rete.
- Le attività e i servizi che utilizzano la modalità di `awsvpc` rete richiedono il ruolo collegato ai servizi Amazon ECS per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure

quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS. Puoi anche creare il ruolo collegato al servizio con il seguente comando. AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- L'istanza Windows di Amazon EC2 richiede la versione `1.54.0` o successiva dell'agente del container per eseguire processi che utilizzano la modalità di rete `awsvpc`. Quando avvii l'istanza, devi configurare le opzioni necessarie per la modalità di rete `awsvpc`. Per ulteriori informazioni, consulta the section called “Avvio delle istanze dei container”.
- Amazon ECS popola il nome host dell'attività con un nome host DNS (interno) fornito da Amazon quando entrambe le opzioni `enableDnsHostnames` e `enableDnsSupport` sono abilitate nel VPC. Se queste opzioni non sono abilitate, il nome host DNS dell'attività sarà un nome host casuale. Per ulteriori informazioni sulle impostazioni DNS per un VPC, consulta Utilizzo del DNS con il VPC nella Guida per l'utente di Amazon VPC.
- Ogni processo Amazon ECS che utilizza la modalità di rete `awsvpc` riceve la propria interfaccia di rete elastica (ENI), che è collegata all'istanza Windows di Amazon EC2 che la ospita. Esiste una quota predefinita per il numero di interfacce di rete che possono essere collegate a un'istanza Windows di Amazon EC2. L'interfaccia di rete principale viene calcolata come unica rispetto a questa quota. Ad esempio, per impostazione predefinita, un'istanza `c5.large` può avere fino a tre ENI collegate. L'interfaccia di rete principale per l'istanza viene calcolata come una di quelle. Puoi collegare altre due ENI all'istanza. Poiché ogni attività che utilizza la modalità di rete `awsvpc` richiede un'ENI, in genere puoi eseguire solo due attività su questo tipo di istanza. Per ulteriori informazioni sui limiti ENI predefiniti per ogni tipo di istanza, consulta Indirizzi IP per interfaccia di rete per tipo di istanza nella Amazon EC2 User Guide.
- Quando si ospitano attività che utilizzano la modalità di rete `awsvpc` sulle istanze Windows di Amazon EC2, le ENI di attività non ricevono 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 tiene conto solo delle ENI che collega all'istanza Windows di Amazon EC2. Se hai collegato manualmente le ENI alle tue istanze, Amazon ECS potrebbe tentare di aggiungere un'attività a un'istanza che non dispone di adattatori di rete sufficienti. Ciò può comportare il timeout

dell'attività e il passaggio a uno stato di deprovisioning e quindi allo stato di arresto. Ti consigliamo di non collegare manualmente le ENI alle istanze.

- Le istanze Windows di Amazon EC2 devono essere registrate con la funzionalità `ecs.capability.task-eni` per essere considerate per il posizionamento di processi con la modalità di rete `awsvpc`.
- Non puoi modificare o scollegare manualmente le ENI che vengono create e collegate alle tue istanze Windows di Amazon EC2. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare le ENI per un'attività, interrompi l'attività.
- 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 AwsVpcConfigurazione nel riferimento 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 i processi che utilizzano la modalità di rete `awsvpc` sono associati a un'ENI e non a un'istanza Windows di Amazon EC2. 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.
- Gli elementi seguenti non sono supportati quando si utilizza la modalità di rete `awsvpc` in una configurazione Windows EC2:
  - 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 tramite IPv4, IPv6 o entrambi. Gli indirizzi IPv4 e IPv6 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 sulla configurazione del VPC per la modalità dual-stack, consulta Migrazione a IPv6 nella Guida per l'utente di Amazon VPC.

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 indirizzo IPv6 possono accedere a Internet tramite un gateway Internet o un gateway Internet solo egress. I gateway NAT sono opzionali. Per ulteriori informazioni, consulta Gateway Internet e Gateway Internet egress-only nella Guida per l'utente di Amazon VPC.

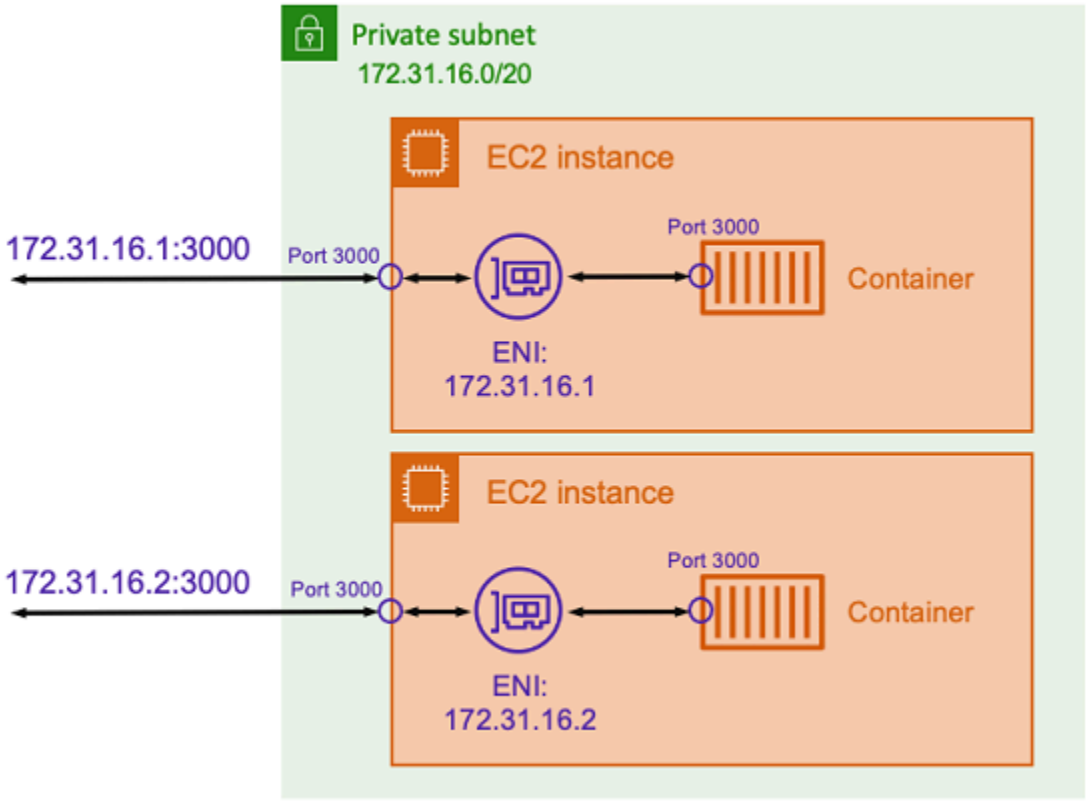
Ai processi Amazon ECS viene assegnato un indirizzo IPv6 se sono soddisfatte le seguenti condizioni:

- L'istanza Linux di Amazon EC2 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 VPC per la modalità dual-stack, consulta Migrazione a IPv6 e Modifica dell'attributo di assegnazione di indirizzi IPv6 pubblici per la sottorete nella Guida per l'utente di Amazon VPC.

Mappa le porte dei container Amazon ECS all'interfaccia di rete dell'istanza EC2

La modalità di rete `host` è supportata solo per le attività Amazon ECS ospitate sulle istanze Amazon EC2. 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 si utilizza la modalità di rete `host`, il container riceve il traffico sulla porta 3000 utilizzando l'indirizzo IP dell'istanza Amazon EC2 dell'host sottostante. Ti consigliamo di non utilizzare questa modalità.

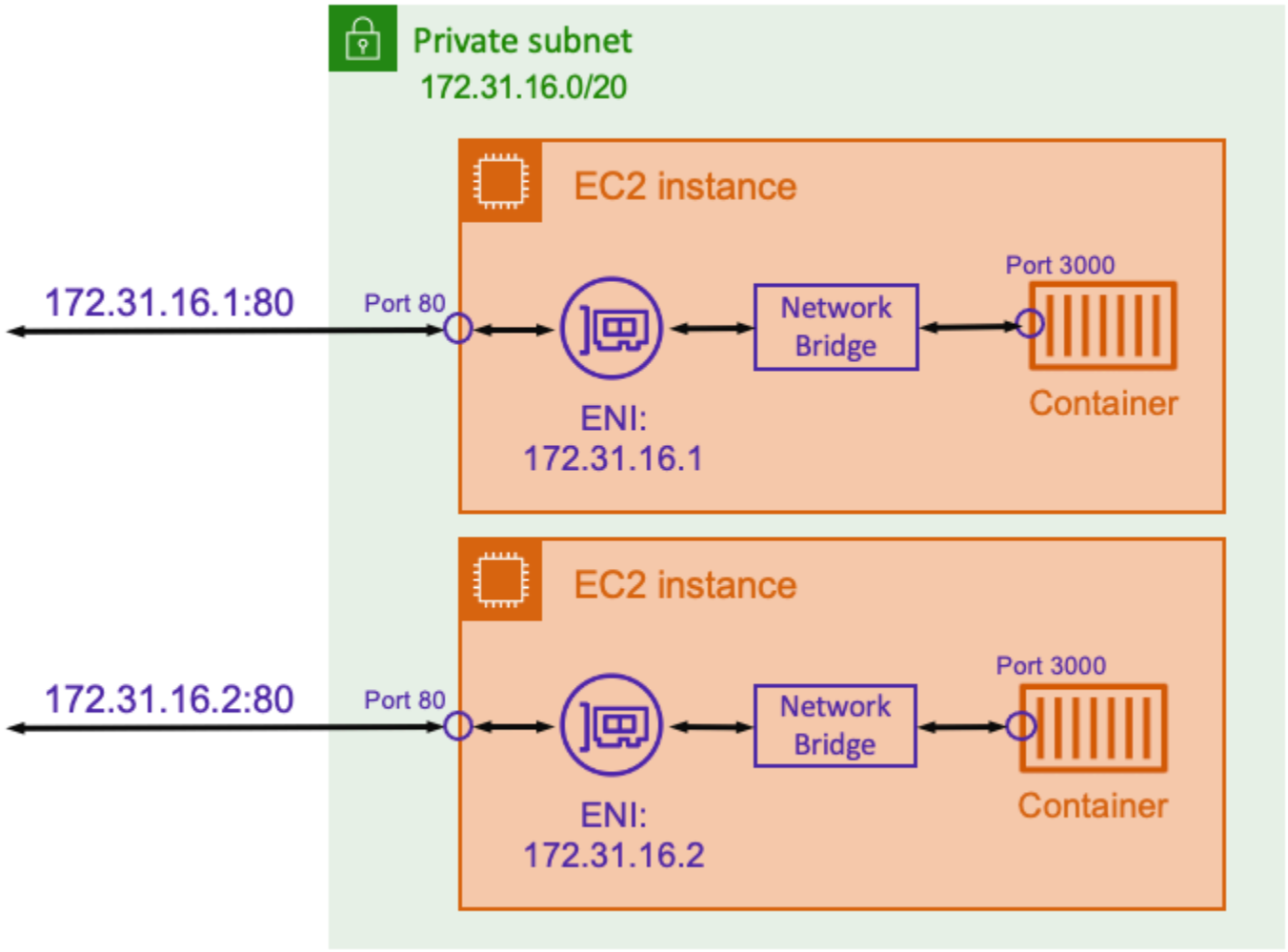
L'utilizzo di questa modalità di rete presenta notevoli svantaggi. È possibile creare solo una singola istanza di un'attività su ciascun host, dal momento che solo la prima attività può essere associata alla porta richiesta sull'istanza Amazon EC2. Inoltre, non è possibile rimappare una porta del container quando si utilizza la modalità di rete `host`. Ad esempio, se un'applicazione deve essere in ascolto su un determinato numero di porta, non è possibile rimappare direttamente il numero di porta. È invece necessario gestire eventuali conflitti di porte modificando la configurazione dell'applicazione.

L'utilizzo della modalità di rete `host` comporta anche implicazioni in termini di sicurezza. Questa modalità consente ai container di impersonare l'host e di connettersi ai servizi di rete loopback privati sull'host.

Usa la rete virtuale di Docker per le attività di Amazon ECS Linux

La modalità di rete `bridge` è supportata solo per le attività Amazon ECS ospitate sulle istanze Amazon EC2.

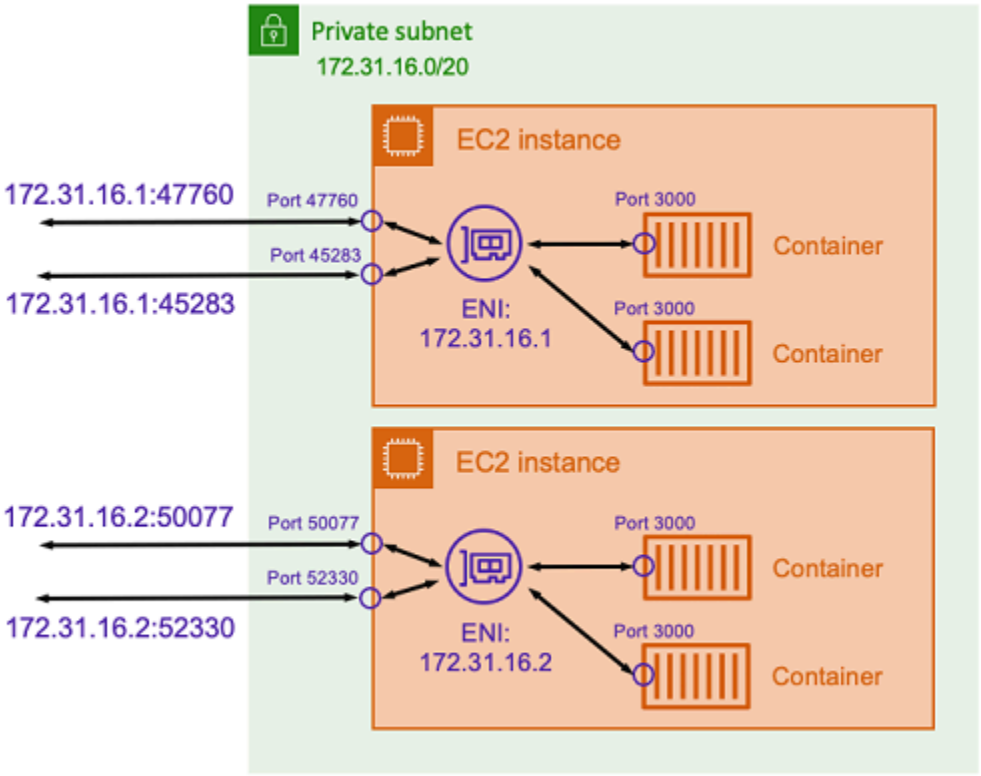
Con la modalità `bridge`, stai utilizzando un bridge di rete virtuale per creare un livello tra l'host e la rete del container. In questo modo, puoi creare mappature delle porte che rimappano una porta host su una porta container. Le mappature possono essere statiche o dinamiche.



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, invia il traffico alla porta `80` all'indirizzo IP dell'istanza Amazon EC2. Dal punto di vista dell'applicazione containerizzata, essa rileva il traffico in entrata sulla porta `3000`.

Se vuoi modificare solo la porta del traffico, la mappatura statica rappresenta la soluzione migliore. Tuttavia, presenta lo stesso svantaggio dell'utilizzo della modalità di rete `host`. È possibile creare solo una singola istanza di un'attività su ciascun host, dal momento che la mappatura statica delle porte consente di mappare solo un solo container sulla porta 80.

Per risolvere questo problema, prendi in considerazione l'utilizzo della modalità di rete `bridge` con una mappatura dinamica delle porte, come illustrato nel diagramma seguente.



Se non specifichi una porta host nella mappatura delle porte, Docker sceglie una porta casuale e inutilizzata dall'intervallo di porte provvisorie e la assegna come porta host pubblica per il container. Ad esempio, all'applicazione Node.js in ascolto sulla porta `3000` del container potrebbe essere assegnato un numero di porta elevato e casuale, ad esempio `47760`, sull'host Amazon EC2. In questo modo è possibile eseguire più copie del container sull'host. Inoltre, a ciascun container può essere assegnata una propria porta sull'host. Ogni copia del container riceve il traffico sulla porta `3000`. Tuttavia, i client che inviano traffico a questi container utilizzano le porte host assegnate casualmente.

Amazon ECS consente di tenere traccia delle porte assegnate casualmente per ogni attività. A tale scopo, aggiorna automaticamente i gruppi target del sistema di bilanciamento del carico e il

rilevamento dei AWS Cloud Map servizi in modo da disporre dell'elenco degli indirizzi IP e delle porte delle attività. Ciò semplifica l'utilizzo dei servizi che operano in modalità `bridge` con porte dinamiche.

Tuttavia, uno svantaggio dell'utilizzo della modalità di rete `bridge` è la difficoltà di bloccare le comunicazioni tra servizi. Poiché i servizi possono essere assegnati a qualsiasi porta casuale e inutilizzata, è necessario aprire ampi intervalli di porte tra gli host. Tuttavia, non è semplice creare regole specifiche in modo che un determinato servizio possa comunicare solo con un altro servizio specifico. I servizi non dispongono di porte specifiche da utilizzare per le regole di rete dei gruppi di sicurezza.

Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate

Di default, a ogni processo di Amazon ECS su Fargate viene fornita una interfaccia di rete elastica (ENI) con un indirizzo IP privato primario. Quando utilizzi una sottorete pubblica, puoi eventualmente assegnare un indirizzo IP pubblico all'ENI dell'attività. Se il tuo VPC è configurato per la modalità dual-stack e utilizzi una sottorete con un blocco CIDR IPv6, 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 sull'utilizzo di VPC e sottoreti, consulta VPC e sottorete nella Guida per l'utente di Amazon VPC.

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 si utilizzano immagini del container ospitate in Amazon ECR, puoi configurare Amazon ECR per utilizzare un endpoint VPC dell'interfaccia e l'estrazione dell'immagine si verificherà sull'indirizzo IPv4 privato dell'attività. 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 alle API di Amazon ECS tramite indirizzi IP privati. AWS PrivateLink limita tutto il traffico di rete tra il tuo VPC e Amazon ECS alla rete Amazon. Non è richiesto un gateway Internet, un dispositivo NAT o un gateway privato virtuale. Per ulteriori informazioni, consulta la sezione AWS PrivateLink nella Guida alle best practice di Amazon ECS.

Per esempi di come utilizzare la `NetworkConfiguration` risorsa con AWS CloudFormation, consulta. the section called “Creazione di risorse Amazon ECS mediante stack separati”

Le ENI create sono completamente gestite da AWS Fargate. Inoltre, esiste una policy IAM associata che viene utilizzata per concedere autorizzazioni per Fargate. Per le attività che utilizzano la piattaforma Fargate versione `1.4.0` o successiva, l'attività riceve un'unica ENI (denominata ENI di attività) e tutto il traffico di rete scorre attraverso tale ENI all'interno del VPC. Questo traffico viene registrato nei log di flusso VPC. Per le attività che utilizzano la piattaforma Fargate versione `1.3.0` e precedenti, oltre all'ENI di attività, l'attività riceve anche un'ENI di proprietà di Fargate che viene utilizzata per un traffico di rete non visibile nei log di flusso VPC. Di seguito vengono descritti il comportamento del traffico di rete e la policy IAM richiesta per ogni versione della piattaforma.

Azione	Flusso di traffico con piattaforma Linux versione **`1.3.0`** e versioni precedenti	Flusso di traffico con piattaforma Linux versione **`1.4.0`**	Flusso di traffico con piattaforma Windows versione **`1.0.0`**	Autorizzazione IAM
Recupero delle credenziali di accesso di Amazon ECR	ENI di proprietà Fargate	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Pull immagine	ENI attività	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Invio dei log tramite un driver di log	ENI attività	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo

Azione	Flusso di traffico con piattaforma Linux versione **1.3.0** e versioni precedenti	Flusso di traffico con piattaforma Linux versione **1.4.0**	Flusso di traffico con piattafor ma Windows versione **1.0.0**	Autorizzazione IAM
Invio di log FireLens per Amazon ECS	ENI attività	ENI attività	ENI attività	Ruolo IAM del processo
Recupero di segreti da Secrets Manager o Systems Manager	ENI di proprietà Fargate	ENI attività	ENI attività	Ruolo IAM per l'esecuzione del processo
Traffico del file system Amazon EFS	Non disponibile	ENI attività	ENI attività	Ruolo IAM del processo
Traffico delle applicazioni	ENI attività	ENI attività	ENI attività	Ruolo IAM del processo

Considerazioni

Tieni in considerazione le informazioni seguenti quando usi la rete di attività.

- Il ruolo collegato ai servizi Amazon ECS è necessario per fornire ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS. Puoi anche creare il ruolo collegato al servizio utilizzando il seguente comando. AWS CLI

  ```
  aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
  ```

- Amazon ECS popola il nome host dell'attività con un nome host DNS fornito da Amazon quando entrambe le opzioni `enableDnsHostnames` e `enableDnsSupport` sono abilitate nel VPC. Se queste opzioni non sono abilitate, il nome host DNS dell'attività è impostato su un nome host

casuale. Per ulteriori informazioni sulle impostazioni DNS per un VPC, consulta Utilizzo del DNS con il VPC nella Guida per l'utente di Amazon VPC.

- È possibile specificare solo fino a 16 sottoreti e 5 gruppi di sicurezza per `awsVpcConfiguration`. Per ulteriori informazioni, consulta AwsVpcConfigurazione nel riferimento all'API di Amazon Elastic Container Service.
- Non puoi scollegare o modificare manualmente le ENI create e collegate da Fargate. Questo serve a impedire l'eliminazione accidentale di un'ENI che è associata a un'attività in esecuzione. Per rilasciare le ENI per un'attività, interrompi l'attività.
- 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.
- Le attività avviate in sottoreti con blocchi CIDR IPv6 ricevono un indirizzo IPv6 solo quando si utilizza la versione della piattaforma Fargate `1.4.0` o successiva per Linux o `1.0.0` per Windows.
- Per le attività che utilizzano la versione `1.4.0` o successiva per Linux o `1.0.0` per Windows della piattaforma, le ENI di attività supportano frame jumbo. Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimensione del payload più grande che si adatta all'interno di un singolo frame. Più grande è l'MTU, più il payload dell'applicazione può essere adattato all'interno di un singolo fotogramma, riducendo il sovraccarico per fotogramma e aumentando l'efficienza. Il supporto dei frame jumbo riduce il sovraccarico quando il percorso di rete tra l'attività e la destinazione supporta frame jumbo.
- I servizi con attività che utilizzano il tipo di avvio Fargate supportano solo Application Load Balancer e Network Load Balancer. Classic Load Balancer non è supportato. Quando crei gruppi target, devi scegliere `ip` come tipo di target anziché `instance`. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS.

Utilizzo di un VPC in modalità dual-stack

Quando utilizzi un VPC in modalità dual-stack, i processi possono comunicare via IPv4, IPv6 o entrambi. Gli indirizzi IPv4 e IPv6 sono indipendenti l'uno dall'altro; devi configurare il routing e la sicurezza del VPC in modo separato per IPv4 e IPv6. Per ulteriori informazioni sulla configurazione del VPC per la modalità dual-stack, consulta Migrazione a IPv6 nella Guida per l'utente di Amazon VPC.

Alle attività Amazon ECS su Fargate viene assegnato un indirizzo IPv6 se sono soddisfatte le seguenti condizioni:

- L'impostazione del tuo `dualStackIPv6` account Amazon ECS è attivata (`enabled`) per il responsabile IAM che avvia le tue attività nella regione in cui le stai avviando. Questa impostazione può essere modificata solo utilizzando l'API o. AWS CLI Hai la possibilità di attivare questa impostazione per un principale IAM specifico sul tuo account o per l'intero account impostando l'impostazione predefinita dell'account. Per ulteriori informazioni, consulta Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account.
- Il VPC e la sottorete sono abilitati per IPv6. Per ulteriori informazioni sulla configurazione del VPC per la modalità dual-stack, consulta Migrazione a IPv6 nella Guida per l'utente di Amazon VPC.
- La sottorete è abilitata per l'assegnazione automatica degli indirizzi IPv6. Per ulteriori informazioni su come configurare la sottorete, consulta Modifica dell'attributo di assegnazione di indirizzi IPv6 pubblici per la sottorete nella Guida per l'utente di Amazon VPC.
- L'attività o il servizio utilizza la versione della piattaforma Fargate `1.4.0` o successiva per Linux.

Se configuri il tuo VPC con un gateway Internet o un gateway Internet solo uscita, le attività Amazon ECS su Fargate a cui è stato assegnato un indirizzo IPv6 possono accedere a Internet. I gateway NAT non sono necessari. Per ulteriori informazioni, consulta Gateway Internet e Gateway Internet egress-only nella Guida per l'utente di Amazon VPC.

Opzioni di storage per le attività di Amazon ECS

Amazon ECS ti offre opzioni di archiviazione easy-to-use dei dati flessibili, convenienti e a seconda delle tue esigenze. Amazon ECS supporta le seguenti opzioni di volume di dati per i contenitori:

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Amazon Elastic Block Store (Amazon EBS)	Fargate, Amazon EC2	Linux	Può essere mantenuto se collegato a un'attività autonoma. Effimero se associato a	I volumi Amazon EBS forniscono storage a blocchi conveniente, durevole e ad alte prestazio ni per carichi di lavoro container

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
			un'attività gestita da un servizio.	izzati a uso intensivo di dati. I casi d'uso più comuni includono carichi di lavoro transazionali come database, desktop virtuali e volumi root e carichi di lavoro ad alta intensità di throughput come l'elabora zione dei log e i carichi di lavoro ETL. Per ulteriori informazi oni, consulta Usa i volumi Amazon EBS con Amazon ECS.

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Amazon Elastic File System (Amazon EFS)	Fargate, Amazon EC2	Linux	Persistente	I volumi Amazon EFS forniscono uno storage di file condiviso semplice, scalabile e persistente da utilizzare con le attività di Amazon ECS, che cresce e si riduce automaticamente man mano che aggiungi e rimuovi file. I volumi Amazon EFS supportano la concorrenza e sono utili per le applicazioni containerizzate che scalano orizzontalmente e richiedono funzionalità di storage come bassa latenza, throughput elevato e coerenza. read-after-write I casi d'uso più comuni

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				includono carichi di lavoro come analisi dei dati, elaborazione multimediale, gestione dei contenuti e server web. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Amazon FSx per Windows File Server	Amazon EC2	Windows	Persistente	I volumi FSx for Windows File Server forniscono file server Windows completamente gestiti che è possibile utilizzare 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 storage persistente per salvare gli output delle applicazioni. Amazon FSx for Windows File Server offre una cartella locale nel contenitore

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				che consente la lettura e la scrittura di più contenitori sullo stesso file system supportato da una condivisione SMB. Per ulteriori informazioni, consulta Usa FSx per volumi Windows File Server con Amazon ECS.

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Volumi Docker	Amazon EC2	Windows, Linux	Persistente	I volumi Docker sono una funzionalità del runtime del contenitore Docker che consente ai contenitori di conservare i dati in modo persistente montando una directory dal file system dell'host. I driver di volume Docker (noti anche come plugin) vengono utilizzati per integrare i volumi dei container con sistemi di archiviazione esterni. I volumi Docker possono essere gestiti da driver di terze parti o dal driver integrato `local`. I casi d'uso comuni dei volumi Docker includono la

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
				fornitura di volumi di dati persistenti o la condivisione di volumi in posizioni diverse su contenitori diversi sulla stessa istanza di contenitore. Per ulteriori informazioni, consulta Usa i volumi Docker con Amazon ECS.

Volume di dati	Tipi di avvio supportati	Sistemi operativi supportati	Persistenza dell'archiviazione	Casi d'uso
Montaggi vincolati	Fargate, Amazon EC2	Windows, Linux	Effimero	I montaggi Bind sono costituiti da un file o una directory sull'host , ad esempio un'istanza AWS Fargate Amazon EC2, montata su un contenitore. I casi d'uso più comuni per i bind mount includono la condivisione di un volume da un contenitore di origine con altri contenitori nello stesso compito o il montaggio di un volume host o di un volume vuoto in uno o più contenito ri. Per ulteriori informazioni, consulta Usa i bind mount con Amazon ECS.

Usa i volumi Amazon EBS con Amazon ECS

I volumi Amazon Elastic Block Store (Amazon EBS) forniscono storage a blocchi ad alta disponibilità, conveniente, durevole e ad alte prestazioni per carichi di lavoro a uso intensivo di dati. I volumi

Amazon EBS possono essere utilizzati con le attività di Amazon ECS per applicazioni con throughput elevato e transazioni intensive.

Durante l'avvio di un'attività autonoma, puoi fornire la configurazione che verrà utilizzata per collegare un volume EBS all'attività. Durante la creazione o l'aggiornamento del servizio, è possibile fornire la configurazione che verrà utilizzata per collegare un volume EBS per attività a ciascuna attività gestita dal servizio ECS.

Fornendo la configurazione del volume al momento dell'avvio anziché nella definizione dell'attività, si creano definizioni di attività che non sono limitate a un tipo di volume di dati specifico o a impostazioni specifiche del volume EBS. È quindi possibile riutilizzare le definizioni delle attività in diversi ambienti di runtime. Ad esempio, è possibile fornire un throughput maggiore durante l'implementazione per i carichi di lavoro di produzione rispetto agli ambienti di pre-produzione.

I volumi Amazon EBS collegati alle attività di Amazon ECS sono gestiti da Amazon ECS per tuo conto. I volumi possono essere crittografati con chiavi AWS Key Management Service (AWS KMS) per proteggere i dati. È possibile configurare nuovi volumi vuoti per gli allegati oppure utilizzare istantanee per caricare dati da volumi esistenti.

Per monitorare le prestazioni del volume, puoi anche utilizzare i CloudWatch parametri di Amazon. Per ulteriori informazioni sui parametri di Amazon ECS per i volumi Amazon EBS, consulta i parametri di Metriche di Amazon ECS CloudWatch Amazon ECS Container Insights.

Per ulteriori informazioni sui volumi Amazon EBS, consulta i volumi Amazon EBS nella Amazon EBS User Guide.

Regioni AWS e zone di disponibilità per i volumi Amazon EBS

I volumi Amazon EBS possono essere collegati alle attività di Amazon ECS nei seguenti modi: Regioni AWS

Nome Regione	Codice regione
US East (N. Virginia)	us-east-1
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti occidentali (California settentrionale)	us-west-1
US West (Oregon)	us-west-2

Nome Regione	Codice regione
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Asia Pacific (Mumbai)	ap-south-1
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 Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europa (Francoforte)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europa (Milano)	eu-south-1
Europe (Paris)	eu-west-3
Europa (Spagna)	eu-south-2
Europa (Stoccolma)	eu-north-1
Europa (Zurigo)	eu-central-2

Nome Regione	Codice regione
Israele (Tel Aviv)	il-central-1
Medio Oriente (Bahrein)	me-south-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Sud America (São Paulo)	sa-east-1

> **Important**
>
> Non è possibile configurare i volumi Amazon EBS per il collegamento alle attività Fargate Amazon ECS `euc1-az2` nelle `use1-az3` zone di disponibilità e.

Considerazioni

Quando utilizzi i volumi Amazon EBS, considera quanto segue:

- I volumi Amazon EBS sono supportati solo per le attività Linux ospitate su Fargate e le attività di avvio EC2 ospitate su istanze Linux basate Nitro su Amazon ECS con Amazon Machine Images (AMI) ottimizzate per Amazon ECS. Per ulteriori informazioni sui tipi di istanza, consulta Tipi di istanza nella Guida per l'utente di Amazon EC2. Per ulteriori informazioni sui tipi di lancio di Amazon ECS, consultaTipi di avvio di Amazon ECS.
- Per le attività ospitate su 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 Linux per Amazon ECS.
- Per le attività ospitate su istanze Amazon EC2 Linux, i volumi Amazon EBS sono supportati su AMI ottimizzate per ECS o versioni successive. `20231219` Per ulteriori informazioni, consulta Recupero dei metadati dell'AMI ottimizzata per Amazon ECS.
- 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 Guida per l'utente di Amazon EC2.
- Un ruolo IAM dell'infrastruttura Amazon ECS è necessario quando si crea un servizio o un'attività autonoma che consiste nella configurazione di un volume al momento della distribuzione. Puoi allegare la policy `AmazonECSInfrastructureRolePolicyForVolumes` IAM AWS gestita al

ruolo oppure puoi utilizzare la policy gestita come guida per creare e allegare la tua policy con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni, consulta Ruolo IAM dell'infrastruttura Amazon ECS.

- Puoi allegare al massimo un volume Amazon EBS a ciascuna attività Amazon ECS e deve trattarsi di un nuovo volume. Non puoi collegare un volume Amazon EBS esistente a un'attività. Tuttavia, puoi configurare un nuovo volume Amazon EBS al momento della distribuzione utilizzando lo snapshot di un volume esistente.
- Puoi configurare i volumi Amazon EBS durante la distribuzione solo per i servizi che utilizzano il tipo di distribuzione con aggiornamento continuo e la strategia di pianificazione delle repliche.
- Amazon ECS aggiunge automaticamente i tag riservati `AmazonECSCreated` e `AmazonECSManaged` al volume allegato. Se rimuovi questi tag dal volume, Amazon ECS non sarà in grado di gestire il volume per tuo conto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS. Per ulteriori informazioni sull'etichettatura delle risorse Amazon ECS, consulta Tagging your Amazon ECS resources.
- Il provisioning di volumi da uno snapshot di un volume Amazon EBS che contiene partizioni non è supportato.
- I volumi collegati alle attività gestite da un servizio non vengono conservati e vengono sempre eliminati al termine dell'attività.
- Non è possibile configurare i volumi Amazon EBS per il collegamento alle attività di Amazon ECS in esecuzione. AWS Outposts

Rimanda la configurazione del volume all'ora di avvio in una definizione di attività Amazon ECS

Per configurare un volume Amazon EBS da allegare alla tua attività, devi specificare la configurazione del punto di montaggio nella definizione dell'attività e assegnare un nome al volume. È inoltre necessario `configuredAtLaunch` impostare `true` perché i volumi Amazon EBS non possono essere configurati per l'allegato nella definizione dell'attività. Invece, i volumi Amazon EBS sono configurati per essere collegati durante la distribuzione.

La seguente definizione dell'attività mostra la sintassi degli `volumes` oggetti `mountPoints` e nella definizione dell'attività. Per ulteriori informazioni sui parametri di definizione delle attività, vedereParametri di definizione delle attività di Amazon ECS. Per utilizzare questo comando, sostituisci *`user input placeholders`* con le tue informazioni.

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 AWS Management Console, vedereCreazione di una definizione di attività Amazon ECS utilizzando la console.

```
{
    "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"
}
```

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato a `Volumes` nella sezione Create a container di Docker Remote API e l'opzione `--volume` a docker run.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il nome del volume da montare.

`containerPath`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

  Tipo: Booleano

  Campo obbligatorio: no

  Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

`name`

- Tipo: stringa

Campo obbligatorio: no

Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini (`-`) e caratteri di sottolineatura (). `_` A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume` `mountPoints`

`configuredAtLaunch`

Tipo: Booleano

Obbligatorio: Sì, se desideri utilizzare, collega un volume EBS direttamente a un'attività.

Speciifica 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 del volume nella definizione dell'attività. Questo parametro deve essere fornito e impostato `true` per configurare un volume Amazon EBS da allegare a un'attività.

Crittografa i dati archiviati nei volumi Amazon EBS per Amazon ECS

Puoi usare AWS Key Management Service (AWS KMS) per creare e gestire chiavi crittografiche che proteggono i tuoi dati. I volumi Amazon EBS vengono crittografati a riposo utilizzando AWS KMS keys. I seguenti tipi di dati sono crittografati:

- Dati archiviati inattivi sul volume
- I/O del disco
- Istantanee create dal volume
- Nuovi volumi creati da istantanee

Puoi configurare la crittografia Amazon EBS per impostazione predefinita in modo che tutti i nuovi volumi creati e collegati a un'attività vengano crittografati utilizzando la chiave KMS che configuri per il tuo account. Per ulteriori informazioni sulla crittografia e la crittografia di Amazon EBS per impostazione predefinita, consulta la crittografia di Amazon EBS nella Guida per l'utente di Amazon EC2.

I volumi Amazon EBS collegati alle attività possono essere crittografati utilizzando un alias `alias/aws/ebs` predefinito Chiave gestita da AWS o una chiave simmetrica gestita dal cliente. Chiavi gestite da AWS I valori predefiniti sono unici Account AWS per ciascun utente Regione AWS

e vengono creati automaticamente. Per creare una chiave simmetrica gestita dal cliente, segui i passaggi descritti in Creazione di chiavi KMS con crittografia simmetrica nella Guida per gli sviluppatori.AWS KMS

Politica delle chiavi KMS gestite dal cliente

Per crittografare un volume EBS collegato all'attività utilizzando la chiave gestita dal cliente, è necessario configurare la politica delle chiavi KMS per garantire che il ruolo IAM utilizzato per la configurazione del volume disponga delle autorizzazioni necessarie per utilizzare la chiave. La policy chiave deve includere le autorizzazioni e. `kms:CreateGrant` `kms:GenerateDataKey*` Le `kms:ReEncryptFrom` autorizzazioni `kms:ReEncryptTo` e sono necessarie per crittografare i volumi creati utilizzando le istantanee. Se desideri configurare e crittografare solo nuovi volumi vuoti da allegare, puoi escludere le autorizzazioni and. `kms:ReEncryptTo` `kms:ReEncryptFrom`

Il seguente frammento di codice JSON mostra le principali dichiarazioni politiche che puoi allegare alla tua politica chiave KMS. L'utilizzo di queste istruzioni consentirà a ECS di utilizzare la chiave per crittografare il volume EBS. Per utilizzare le dichiarazioni politiche di esempio, sostituiscile *`user input placeholders`* con le tue informazioni. Come sempre, configura solo le autorizzazioni di cui hai bisogno.

```
{
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
      "Action": "kms:DescribeKey",
      "Resource":"*"
    },
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
      "Action": [
      "kms:GenerateDataKey*",
      "kms:ReEncryptTo",
      "kms:ReEncryptFrom"
      ],
      "Resource":"*",
      "Condition": {
        "StringEquals": {
          "kms:CallerAccount": "aws_account_id",
          "kms:ViaService": "ec2.region.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
```

```
          "kms:EncryptionContextKeys": "aws:ebs:id"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": { "AWS": "arn:aws:iam::111122223333:role/ecsInfrastructureRole" },
      "Action": "kms:CreateGrant",
      "Resource":"*",
      "Condition": {
        "StringEquals": {
          "kms:CallerAccount": "aws_account_id",
          "kms:ViaService": "ec2.region.amazonaws.com"
        },
        "ForAnyValue:StringEquals": {
          "kms:EncryptionContextKeys": "aws:ebs:id"
        },
        "Bool": {
          "kms:GrantIsForAWSResource": true
        }
      }
    }
```

Per ulteriori informazioni sulle politiche e le autorizzazioni chiave, consulta Politiche chiave AWS KMS e AWS KMS autorizzazioni nella Guida per gli AWS KMS sviluppatori. Per la risoluzione dei problemi relativi agli allegati dei volumi EBS relativi alle autorizzazioni chiave, consulta. Risoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS

Specificare la configurazione del volume Amazon EBS nella distribuzione di Amazon ECS

Dopo aver registrato una definizione di attività con il `configuredAtLaunch` parametro impostato su`true`, puoi configurare un volume Amazon EBS durante la distribuzione quando esegui un'attività autonoma o quando crei o aggiorni un servizio.

Per configurare un volume, puoi utilizzare le API Amazon ECS o passare un file JSON come input per i seguenti comandi: AWS CLI

- `run-task`per eseguire un'attività ECS autonoma.
- `start-task`per eseguire un'attività ECS autonoma in un'istanza di contenitore specifica. Questo comando non è applicabile alle attività di tipo avvio di Fargate.
- `create-service`per creare un nuovo servizio ECS.

- `update-service`per aggiornare un servizio esistente.

> **Note**
>
> Affinché un contenitore della tua attività possa scrivere sul volume Amazon EBS montato, devi eseguire il contenitore come utente root.

Puoi anche configurare un volume Amazon EBS utilizzando. AWS Management Console Per ulteriori informazioni, consulta Esecuzione di un'applicazione come attività Amazon ECS, Creazione di un servizio Amazon ECS utilizzando la console e Aggiornamento di un servizio Amazon ECS tramite la console.

Il seguente frammento di codice JSON mostra tutti i parametri di un volume Amazon EBS che può essere configurato al momento della distribuzione. Per utilizzare questi parametri per la configurazione del volume, sostituiscili *`user input placeholders`* con le tue informazioni. Per ulteriori informazioni su questi parametri, consulta Configurazioni dei volumi.

```
"volumeConfigurations": [
        {
            "name": "ebs-volume",
            "managedEBSVolume": {
                "encrypted": true,
                "kmsKeyId": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                "volumeType": "gp3",
                "sizeInGiB": 10,
                "snapshotId": "snap-12345",
                "iops": 3000,
                "throughput": 125,
                "tagSpecifications": [
                    {
                        "resourceType": "volume",
                        "tags": [
                            {
                                "key": "key1",
                                "value": "value1"
                            }
                        ],
                        "propagateTags": "NONE"
                    }
```

```
                ],
                "roleArn": "arn:aws::iam:1111222333:role/ecsInfrastructureRole",
                 "terminationPolicy": {
                    "deleteOnTermination": true//can't be configured for service-
managed tasks, always true
                },
                "filesystemType": "ext4"
            }
        }
    ]
```

> **Important**
>
> Assicurati che `volumeName` quello specificato nella configurazione sia lo stesso `volumeName` specificato nella definizione dell'attività.

Per informazioni sulla verifica dello stato del volume allegato, vedereRisoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS . Per informazioni sul ruolo dell'infrastruttura Amazon ECS AWS Identity and Access Management (IAM) necessario per l'allegato del volume EBS, consulta. Ruolo IAM dell'infrastruttura Amazon ECS

Di seguito sono riportati alcuni esempi di snippet JSON che mostrano la configurazione dei volumi Amazon EBS. Questi esempi possono essere utilizzati salvando gli snippet in file JSON e passando i file come parametri (utilizzando il parametro) per i comandi. `--cli-input-json file://filename` AWS CLI Sostituire *`user input placeholders`* con le proprie informazioni.

Configurare un volume per un'attività autonoma

Il seguente frammento mostra la sintassi per configurare i volumi Amazon EBS per l'allegato a un'attività autonoma. Il seguente frammento di codice JSON mostra la sintassi per la configurazione di,, e impostazioni. `volumeType` `sizeInGiB` `encrypted` `kmsKeyId` La configurazione specificata nel file JSON viene utilizzata per creare e allegare un volume EBS all'attività autonoma.

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "volumeConfigurations": [
        {
            "name": "datadir",
```

```
            "managedEBSVolume": {
                "volumeType": "gp3",
                "sizeInGiB": 100,
                "roleArn":"arn:aws:iam:1111222333:role/ecsInfrastructureRole",
                "encrypted": true,
                "kmsKeyId":
 "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            }
        }
    ]
}
```

Configura un volume al momento della creazione del servizio

Il seguente frammento mostra la sintassi per configurare i volumi Amazon EBS per l'allegato alle attività gestite da un servizio. I volumi provengono dallo snapshot utilizzando. `snapshotId` La configurazione specificata nel file JSON viene utilizzata per creare e allegare un volume EBS a ciascuna attività gestita dal servizio.

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "serviceName": "mysvc",
   "desiredCount": 2,
   "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
              "roleArn":"arn:aws:iam:1111222333:role/ecsInfrastructureRole",
              "snapshotId": "snap-12345"
            }
        }
   ]
}
```

Configura un volume durante l'aggiornamento del servizio

Il seguente frammento di codice JSON mostra la sintassi per l'aggiornamento di un servizio che in precedenza non aveva volumi Amazon EBS configurati per il collegamento alle attività. È necessario fornire l'ARN di una revisione della definizione di attività impostata `configuredAtLaunch` su. `true` Il seguente frammento di codice JSON mostra la sintassi per configurare le impostazioni`volumeType`,,, `sizeInGiB` e e. `throughput` `iops` `filesystemType` Questa

configurazione viene utilizzata per creare e collegare un volume EBS a ciascuna attività gestita dal servizio.

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "serviceName": "mysvc",
   "desiredCount": 2,
  "volumeConfigurations": [
        {
            "name": "myEbsVolume",
            "managedEBSVolume": {
              "roleArn":"arn:aws:iam:1111222333:role/ecsInfrastructureRole",
               "volumeType": "gp3",
                "sizeInGiB": 100,
                 "iops": 3000,
                "throughput": 125,
                "filesystemType": "ext4"
            }
        }
   ]
}
```

Configura un servizio per non utilizzare più i volumi Amazon EBS

Il seguente frammento di codice JSON mostra la sintassi per aggiornare un servizio in modo che non utilizzi più i volumi Amazon EBS. È necessario fornire l'ARN di una definizione di attività con `configuredAtLaunch` set to `false` o una definizione di attività senza il `configuredAtLaunch` parametro. È inoltre necessario fornire un `volumeConfigurations` oggetto vuoto.

```
{
   "cluster": "mycluster",
   "taskDefinition": "mytaskdef",
   "serviceName": "mysvc",
   "desiredCount": 2,
   "volumeConfigurations": []
}
```

Politica di terminazione per i volumi Amazon EBS

Quando un'attività Amazon ECS termina, Amazon ECS utilizza il `deleteOnTermination` valore per determinare se il volume Amazon EBS associato all'attività terminata deve essere eliminato. Per

impostazione predefinita, i volumi EBS collegati alle attività vengono eliminati quando l'attività viene terminata. Per le attività autonome, puoi modificare questa impostazione per preservare invece il volume al termine dell'attività.

> **Note**
>
> I volumi collegati alle attività gestite da un servizio non vengono conservati e vengono sempre eliminati al termine dell'attività.

Contrassegna i volumi Amazon EBS

Puoi etichettare i volumi Amazon EBS utilizzando l'`tagSpecifications`oggetto. Utilizzando l'oggetto, puoi fornire tag personalizzati e impostare la propagazione dei tag dalla definizione dell'attività o dal servizio, a seconda che il volume sia collegato a un'attività autonoma o a un'attività in un servizio. Il numero massimo di tag che è possibile allegare a un volume è 50.

> **Important**
>
> Amazon ECS allega automaticamente i tag `AmazonECSManaged` riservati a un volume Amazon EBS. `AmazonECSCreated` Ciò significa che puoi controllare l'allegato di un massimo di 48 tag aggiuntivi a un volume. Questi tag aggiuntivi possono essere tag definiti dall'utente, gestiti da ECS o propagati.

Se desideri aggiungere tag gestiti da Amazon ECS al tuo volume, devi `enableECSManagedTags` impostarlo su `true` nella tua `UpdateService` chiamata `CreateService` `RunTask` o`StartTask`. Se attivi i tag gestiti da Amazon ECS, Amazon ECS taggherà automaticamente il volume con informazioni su cluster e servizi (`aws:ecs:`*`clusterName`*e). `aws:ecs:`*`serviceName`* Per ulteriori informazioni sull'etichettatura delle risorse Amazon ECS, consulta Tagging your Amazon ECS resources.

Il seguente frammento di codice JSON mostra la sintassi per etichettare ogni volume Amazon EBS collegato a ciascuna attività in un servizio con un tag definito dall'utente. Per utilizzare questo esempio per creare un servizio, sostituiscilo con le tue informazioni. *`user input placeholders`*

```
{
   "cluster": "mycluster",
```

```
    "taskDefinition": "mytaskdef",
    "serviceName": "mysvc",
    "desiredCount": 2,
    "enableECSManagedTags": true,
    "volumeConfigurations": [
         {
             "name": "datadir",
             "managedEBSVolume": {
                 "volumeType": "gp3",
                 "sizeInGiB": 100,
                  "tagSpecifications": [
                     {
                         "resourceType": "volume",
                         "tags": [
                             {
                                 "key": "key1",
                                 "value": "value1"
                             }
                         ],
                         "propagateTags": "NONE"
                     }
                 ]
                 "roleArn":"arn:aws:iam:1111222333:role/ecsInfrastructureRole",
                 "encrypted": true,
                 "kmsKeyId":
  "arn:aws:kms:region:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
             }
         }
    ]
}
```

> **Important**
>
> È necessario specificare un tipo di `volume` risorsa per etichettare i volumi Amazon EBS.

Prestazioni dei volumi Amazon EBS per le attività on-demand di Fargate

Il volume, gli IOPS e il throughput di base di Amazon EBS disponibili per un'attività on-demand di Fargate dipendono dal numero totale di unità CPU richieste per l'attività. Se richiedi 0,25, 0,5 o 1 unità CPU virtuale (vCPU) per l'attività Fargate, ti consigliamo di configurare un volume SSD per uso generico gp2 (gp3o) o un volume Hard Disk Drive (HDD) (o). `st1` `sc1` Se richiedi più di 1 vCPU

per l'attività Fargate, i seguenti limiti prestazionali di base si applicano a un volume Amazon EBS collegato all'attività. È possibile ottenere temporaneamente prestazioni EBS superiori ai seguenti limiti. Tuttavia, ti consigliamo di pianificare il tuo carico di lavoro in base a questi limiti.

Unità CPU richieste (in vCPU)	IOPS Amazon EBS di base (16 KiB di I/O)	Throughput di base di Amazon EBS (in, MiBps 128 KB di I/O)	Larghezza di banda di base (in Mbps)
2	3.000	75	360
4	5.000	120	1.150
8	10.000	250	2.300
16	15.000	500	4.500

> **Note**
>
> Quando configuri un volume Amazon EBS per il collegamento a un'attività Fargate, il limite di prestazioni di Amazon EBS per l'attività Fargate viene condiviso tra lo storage temporaneo dell'attività e il volume allegato.

Risoluzione dei problemi relativi ai volumi Amazon EBS allegati alle attività di Amazon ECS

Potrebbe essere necessario risolvere i problemi o verificare il collegamento dei volumi Amazon EBS alle attività di Amazon ECS.

Verifica lo stato degli allegati al volume

Puoi utilizzare il AWS Management Console per visualizzare lo stato dell'allegato di un volume Amazon EBS a un'attività Amazon ECS. Se l'attività inizia e l'allegato non riesce, vedrai anche un motivo dello stato che puoi utilizzare per risolvere i problemi. Il volume creato verrà eliminato e l'operazione verrà interrotta. Per ulteriori informazioni sui motivi dello stato, vedereMotivi 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.

2. Nella pagina Cluster, scegli il cluster in cui è in esecuzione l'attività. Viene visualizzata la pagina dei dettagli del cluster.
3. Nella pagina dei dettagli del cluster, scegli la scheda Attività.
4. Scegli l'attività per la quale desideri visualizzare lo stato degli allegati al volume. Potrebbe essere necessario utilizzare Filtra lo stato desiderato e scegliere Interrotto se l'attività che desideri esaminare è stata interrotta.
5. Nella pagina dei dettagli dell'attività, scegli la scheda Volumi. Potrai visualizzare lo stato degli allegati del volume Amazon EBS in Attachment status. Se il volume non riesce a collegarsi all'attività, puoi scegliere lo stato in Attachment status per visualizzare la causa dell'errore.

Puoi anche visualizzare lo stato degli allegati al volume di un'attività e il motivo dello stato associato utilizzando l'DescribeTasksAPI.

Guasti del servizio e delle attività

Potresti riscontrare errori di servizio o attività non specifici dei volumi Amazon EBS che possono influire sull'allegato del volume. Per ulteriori informazioni, consulta la pagina

- Messaggi di evento relativi ai servizi
- Codici di errore delle attività interrotte
- Motivi di errore dell'API

Motivi dello stato dell'allegato del volume Amazon EBS alle attività di Amazon ECS

Utilizza il seguente riferimento per risolvere i problemi che potresti riscontrare sotto forma di motivi di stato AWS Management Console durante la configurazione dei volumi Amazon EBS da allegare alle attività di Amazon ECS. Per ulteriori informazioni su come individuare questi motivi di status nella console, consulta. Verifica lo stato degli allegati al volume

ECS non è stato in grado di assumere il ruolo dell'infrastruttura ECS configurato 'arn:aws:iam:: 111122223333:role/ ecs '. InfrastructureRole
Verifica che il ruolo assegnato abbia un rapporto di fiducia adeguato con Amazon ECS

Questo motivo di status appare nei seguenti scenari.

- Fornisci un ruolo IAM senza la necessaria policy di fiducia allegata. Amazon ECS non può accedere al ruolo IAM dell'infrastruttura Amazon ECS fornito da te se il ruolo non dispone della politica di fiducia necessaria. L'attività può rimanere bloccata nello `DEPROVISIONING` stato.

Per ulteriori informazioni sulla politica di fiducia necessaria, vedereRuolo IAM dell'infrastruttura Amazon ECS.

- Il tuo utente IAM non è autorizzato a trasferire il ruolo dell'infrastruttura Amazon ECS ad Amazon ECS. L'attività può rimanere bloccata nello `DEPROVISIONING` stato. Per evitare questo problema, puoi allegare l'`PassRole`autorizzazione al tuo utente. Per ulteriori informazioni, consulta Ruolo IAM dell'infrastruttura Amazon ECS.
- Il tuo ruolo IAM non dispone delle autorizzazioni necessarie per gli allegati del volume Amazon EBS. L'attività può rimanere bloccata nello `DEPROVISIONING` stato. Per ulteriori informazioni sulle autorizzazioni specifiche necessarie per collegare i volumi Amazon EBS alle attività, consulta. Ruolo IAM dell'infrastruttura Amazon ECS

> **Note**
>
> Potresti anche visualizzare questo messaggio di errore a causa di un ritardo nella propagazione dei ruoli. Se riprovando a utilizzare il ruolo dopo aver atteso qualche minuto il problema non risolve il problema, è possibile che tu abbia configurato erroneamente la politica di attendibilità per il ruolo.

ECS non è riuscito a configurare il volume EBS. Incontrato IdempotentParameterMismatch «; «Il token client che hai fornito è associato a una risorsa che è già stata eliminata. Utilizza un token client diverso.»

I seguenti scenari AWS KMS chiave possono portare alla `IdempotentParameterMismatch` visualizzazione di un messaggio:

- Specificate un ARN, un ID o un alias della chiave KMS che non è valido. In questo scenario, l'attività potrebbe sembrare avviata correttamente, ma alla fine l'attività non riesce perché AWS autentica la chiave KMS in modo asincrono. Per ulteriori informazioni, consulta la crittografia di Amazon EBS nella Guida per l'utente di Amazon EC2.
- Fornisci una chiave gestita dal cliente priva delle autorizzazioni che consentono al ruolo IAM dell'infrastruttura Amazon ECS di utilizzare la chiave per la crittografia. Per evitare problemi di autorizzazione relativi alle policy chiave, consulta la policy AWS KMS chiave di esempio in Data encryption for Amazon EBS Volumes.

Puoi configurare Amazon EventBridge per inviare eventi di volume Amazon EBS e eventi di modifica dello stato delle attività di Amazon ECS a una destinazione, ad esempio gruppi Amazon CloudWatch . Puoi quindi utilizzare questi eventi per identificare lo specifico problema relativo

alla chiave gestita dal cliente che ha influito sull'allegato del volume. Per ulteriori informazioni, consulta la pagina

- Come posso creare un gruppo di CloudWatch log da utilizzare come destinazione per una EventBridge regola? su AWS Re:post.
- Eventi di modifica dello stato dell'attività.
- EventBridgeper Amazon EBS nella Amazon EBS User Guide.

ECS è scaduto durante la configurazione dell'allegato del volume EBS al tuo Task.

I seguenti scenari di formato del file system generano questo messaggio.

- Il formato del file system specificato durante la configurazione non è compatibile con il sistema operativo dell'attività.
- Si configura un volume Amazon EBS da creare da uno snapshot e il formato del file system dello snapshot non è compatibile con il sistema operativo dell'operazione. Per i volumi creati da uno snapshot, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dello snapshot.

Puoi utilizzare i log degli agenti container di Amazon ECS per risolvere questo messaggio per attività di tipo di avvio di Amazon EC2. Per ulteriori informazioni, consulta le posizioni dei file di registro di Amazon ECS e Amazon ECS log collector.

Usa i volumi Amazon EFS con Amazon ECS

Amazon Elastic File System (Amazon EFS) offre archiviazione file semplice e scalabile da utilizzare con i processi Amazon ECS. Con Amazon EFS, la capacità di storage è elastica. La capacità di storage aumenta e si riduce automaticamente quando si aggiungono e si rimuovono i file. Le tue applicazioni possono disporre dello storage di cui hanno bisogno nel momento in cui ne hanno bisogno.

Puoi utilizzare i file system Amazon EFS con Amazon ECS per esportare i dati del file system all'interno del tuo parco istanze di container. In questo modo, le tue attività hanno accesso allo stesso storage persistente, indipendentemente dall'istanza in cui si trovano. Inoltre, per utilizzare il file system, le tue definizioni di attività devono fare riferimento ai montaggi di volume inerenti l'istanza di container.

Per un tutorial, vedere Configurazione dei file system Amazon EFS per Amazon ECS tramite la console.

Considerazioni

Quando usi i volumi Amazon EFS, tieni presente quanto segue:

- Per le attività che utilizzano il tipo di avvio EC2, il supporto del file system Amazon EFS è stato aggiunto come anteprima pubblica con l'AMI ottimizzata per Amazon ECS versione `20191212` con l'agente del 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 AMI Linux ottimizzate per Amazon ECS.

  > **Note**
  >
  > Se crei la tua AMI, è necessario utilizzare l'agente del container 1.38.0 o versione successiva, `ecs-init` versione 1.38.0-1 o successiva ed emettere i seguenti comandi sull'istanza Amazon EC2 per abilitare il plug-in del volume Amazon ECS. I comandi dipendono dal fatto che si stia usando Amazon Linux 2 o Amazon Linux come immagine di base.
  >
  > Amazon Linux 2
  >
  > ```
  > yum install amazon-efs-utils
  > systemctl enable --now amazon-ecs-volume-plugin
  > ```
  >
  > Amazon Linux
  >
  > ```
  > yum install amazon-efs-utils
  > sudo shutdown -r now
  > ```

- Per le attività ospitate su Fargate, i file system Amazon EFS sono supportati sulla piattaforma versione 1.4.0 o successiva (Linux). Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux 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 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 del tipo di lancio di Amazon EC2, consulta. Endpoint di metadati delle attività Amazon ECS versione 4 Per ulteriori informazioni sull'utilizzo del tipo di lancio Fargate, vedere. Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate

- L'uso di volumi Amazon EFS o la specifica di una `EFSVolumeConfiguration` non sono supportati su istanze esterne.
- Consigliamo di impostare il parametro `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION` nel file di configurazione dell'agente su un valore inferiore a quello predefinito (circa 1 ora). Questa modifica aiuta a prevenire la scadenza delle credenziali di montaggio EFS e consente la pulizia dei supporti che non sono in uso. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Usa i punti di accesso Amazon EFS

I punti di accesso Amazon EFS sono punti di accesso specifici dell'applicazione in un file system EFS per la gestione dell'accesso dell'applicazione ai set di dati condivisi. Per ulteriori informazioni sui punti di accesso Amazon EFS e su come controllare l'accesso a tali punti, consulta Working with Amazon EFS Access Points (Utilizzo dei punti di accesso Amazon EFS) nella Guida per l'utente di Amazon Elastic File System.

I punti di accesso possono applicare un'identità utente, inclusi i gruppi dell'utente POSIX, per tutte le richieste al file system effettuate tramite il punto di accesso. I punti di accesso possono inoltre applicare una directory root diversa per il file system. In questo modo i client possono accedere solo ai dati nella directory specificata o nelle relative sottodirectory.

> **Note**
>
> Quando crei un punto di accesso EFS, è necessario specificare un percorso nel file system da utilizzare come directory root. Quando si fa riferimento al file system EFS con un ID punto di accesso nella definizione di attività Amazon ECS, la directory root deve essere omessa o impostata su `/` in modo da forzare il percorso impostato sul punto di accesso EFS.

Puoi utilizzare un ruolo IAM del processo Amazon ECS per imporre che applicazioni specifiche utilizzino un punto di accesso specifico. Combinando le policy IAM con i punti di accesso, puoi fornire accesso sicuro a set di dati specifici per le applicazioni. Per ulteriori informazioni su come utilizzare i ruoli IAM dell'attività, consulta Ruolo IAM dell'attività Amazon ECS.

Best practice per l'utilizzo dei volumi Amazon EFS con Amazon ECS

Prendi nota dei seguenti consigli sulle best practice quando usi Amazon EFS con Amazon ECS.

Sicurezza e controlli di accesso per i volumi Amazon EFS

Amazon EFS offre funzionalità di controllo degli accessi che puoi utilizzare per garantire che i dati archiviati in un file system Amazon EFS siano sicuri e accessibili solo dalle applicazioni che ne hanno bisogno. Puoi proteggere i dati abilitando la crittografia a riposo e in transito. Per ulteriori informazioni, consulta Crittografia dati in Amazon EFS nella Amazon Elastic File System User Guide.

Oltre alla crittografia dei dati, puoi anche utilizzare Amazon EFS per limitare l'accesso a un file system. Esistono tre modi per implementare il controllo degli accessi in EFS.

- Gruppi di sicurezza: con gli obiettivi di montaggio di Amazon EFS, puoi configurare un gruppo di sicurezza utilizzato per consentire e negare il traffico di rete. Puoi configurare il gruppo di sicurezza collegato ad Amazon EFS per consentire il traffico NFS (porta 2049) dal gruppo di sicurezza collegato alle tue istanze Amazon ECS o, quando utilizzi la modalità di `awsvpc` rete, il task Amazon ECS.
- IAM: puoi limitare l'accesso a un file system Amazon EFS utilizzando IAM. Una volta configurate, le attività di Amazon ECS richiedono un ruolo IAM per l'accesso al file system per montare un file system EFS. Per ulteriori informazioni, consulta Using IAM per controllare l'accesso ai dati del file system nella Amazon Elastic File System User Guide.

  Le policy IAM possono anche applicare condizioni predefinite, come richiedere a un client di utilizzare TLS durante la connessione a un file system Amazon EFS. Per ulteriori informazioni, consulta le chiavi di condizione di Amazon EFS per i client nella Amazon Elastic File System User Guide.
- Punti di accesso Amazon EFS: i punti di accesso Amazon EFS sono punti di ingresso specifici dell'applicazione in un file system Amazon EFS. Puoi utilizzare i punti di accesso per applicare l'identità di un utente, inclusi i gruppi POSIX dell'utente, per tutte le richieste del file system effettuate tramite il punto di accesso. I punti di accesso possono inoltre applicare una directory root diversa per il file system. In questo modo i client possono accedere solo ai dati nella directory specificata o nelle sue sottodirectory.

Prendi in considerazione l'implementazione di tutti e tre i controlli di accesso su un file system Amazon EFS per la massima sicurezza. Ad esempio, puoi configurare il gruppo di sicurezza collegato a un punto di montaggio Amazon EFS per consentire solo il traffico NFS in ingresso da un gruppo

di sicurezza associato all'istanza del contenitore o all'attività Amazon ECS. Inoltre, puoi configurare Amazon EFS per richiedere un ruolo IAM per accedere al file system, anche se la connessione proviene da un gruppo di sicurezza consentito. Infine, puoi utilizzare gli access point Amazon EFS per applicare le autorizzazioni utente POSIX e specificare le directory root per le applicazioni.

Il seguente frammento di definizione delle attività mostra come montare un file system Amazon EFS utilizzando un punto di accesso.

```
"volumes": [
    {
      "efsVolumeConfiguration": {
        "fileSystemId": "fs-1234",
        "authorizationConfig": {
          "acessPointId": "fsap-1234",
          "iam": "ENABLED"
        },
        "transitEncryption": "ENABLED",
        "rootDirectory": ""
      },
      "name": "my-filesystem"
    }
]
```

Prestazioni del volume Amazon EFS

Amazon EFS offre due modalità di prestazioni: General Purpose e Max I/O. General Purpose è adatto per applicazioni sensibili alla latenza come sistemi di gestione dei contenuti e strumenti CI/CD. Al contrario, i file system Max I/O sono adatti per carichi di lavoro come analisi dei dati, elaborazione multimediale e apprendimento automatico. Questi carichi di lavoro devono eseguire operazioni parallele da centinaia o addirittura migliaia di container e richiedono il throughput aggregato e gli IOPS più elevati possibili. Per ulteriori informazioni, consulta le modalità di prestazione di Amazon EFS nella Amazon Elastic File System User Guide.

Alcuni carichi di lavoro sensibili alla latenza richiedono sia i livelli di I/O più elevati forniti dalla modalità di prestazioni Max I/O sia la latenza più bassa fornita dalla modalità prestazionale General Purpose. Per questo tipo di carico di lavoro, consigliamo di creare più file system in modalità prestazionale per uso generico. In questo modo, è possibile distribuire il carico di lavoro delle applicazioni su tutti questi file system, purché il carico di lavoro e le applicazioni siano in grado di supportarlo.

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 Provisioned Throughput o dalla quantità di dati archiviati nella classe di storage EFS Standard o One Zone per i file system che utilizzano Bursting Throughput. Per ulteriori informazioni, consulta Understanding metered throughput nella Amazon Elastic File System User Guide.

La modalità di throughput predefinita per i file system Amazon EFS è la modalità bursting. Con la modalità bursting, la velocità effettiva disponibile per un file system aumenta o diminuisce man mano che il file system cresce. Poiché i carichi di lavoro basati su file in genere registrano picchi, richiedendo livelli di throughput elevati per periodi di tempo e livelli di throughput inferiori per il resto del tempo, Amazon EFS è progettato per consentire livelli di throughput elevati per periodi di tempo. Inoltre, poiché molti carichi di lavoro richiedono un elevato livello di lettura, le operazioni di lettura vengono misurate con un rapporto di 1:3 rispetto ad altre operazioni NFS (come la scrittura).

Tutti i file system Amazon EFS offrono prestazioni di base costanti di 50 MB/s per ogni TB di storage Amazon EFS Standard o Amazon EFS One Zone. Tutti i file system (indipendentemente dalle dimensioni) possono raggiungere i 100 MB/s. I file system con più di 1 TB di storage EFS Standard o EFS One Zone possono raggiungere i 100 MB/s per ogni TB. Poiché le operazioni di lettura vengono misurate con un rapporto di 1:3, è possibile arrivare a 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 si ridimensiona in modo lineare e automatico con lo storage nella classe di storage Amazon EFS Standard. Se hai bisogno di un throughput superiore a quello che puoi ottenere con la quantità di dati archiviati, puoi configurare Provisioned Throughput sulla quantità specifica richiesta dal tuo carico di lavoro.

La velocità effettiva del file system è condivisa tra tutte le istanze Amazon EC2 connesse a un file system. Ad esempio, un file system da 1 TB in grado di raggiungere 100 MB/s di throughput può generare 100 MB/s da una singola istanza di Amazon EC2 e ogni istanza può raggiungere 10 MB/s. Per ulteriori informazioni, consulta le prestazioni di Amazon EFS nella Amazon Elastic File System User Guide.

Ottimizzazione dei costi per i volumi Amazon EFS

Amazon EFS semplifica la scalabilità dello storage per te. I file system Amazon EFS crescono automaticamente man mano che aggiungi più dati. Soprattutto con la modalità Amazon EFS Bursting Throughput, la velocità effettiva su Amazon EFS si ridimensiona all'aumentare delle dimensioni del file system nella classe di storage standard. Per migliorare la velocità effettiva senza pagare costi

aggiuntivi per la velocità effettiva assegnata su un file system EFS, puoi condividere un file system Amazon EFS con più applicazioni. Utilizzando i punti di accesso Amazon EFS, puoi implementare l'isolamento dello storage in file system Amazon EFS condivisi. In questo modo, anche se le applicazioni condividono ancora lo stesso file system, non possono accedere ai dati a meno che tu non li autorizzi.

Man mano che i dati crescono, Amazon EFS ti aiuta a spostare automaticamente i file a cui si accede raramente in una classe di storage inferiore. La classe di storage Amazon EFS Standard-Infrequent Access (IA) riduce i costi di storage per i file a cui non si accede ogni giorno. Lo fa senza sacrificare l'elevata disponibilità, l'elevata durabilità, l'elasticità e l'accesso al file system POSIX forniti da Amazon EFS. Per ulteriori informazioni, consulta le classi di storage Amazon EFS nella Amazon Elastic File System User Guide.

Prendi in considerazione l'utilizzo delle policy del ciclo di vita di Amazon EFS per risparmiare automaticamente denaro spostando i file a cui si accede raramente nello storage Amazon EFS IA. Per ulteriori informazioni, consulta Gestione del ciclo di vita EFS nella Guida per l'utente di Amazon Elastic File System.

Quando crei un file system Amazon EFS, puoi scegliere se Amazon EFS replica i dati su più zone di disponibilità (Standard) o archiviarli in modo ridondante all'interno di un'unica zona di disponibilità. La classe di storage Amazon EFS One Zone può ridurre i costi di storage di un margine significativo rispetto alle classi di storage Amazon EFS Standard. Prendi in considerazione l'utilizzo della classe di storage Amazon EFS One Zone per carichi di lavoro che non richiedono resilienza Multi-AZ. Puoi ridurre ulteriormente il costo dello storage Amazon EFS One Zone spostando i file a cui si accede raramente su Amazon EFS One Zone-Infrequent Access. Per ulteriori informazioni, consulta la sezione Amazon EFS Infrequent Access.

Protezione dei dati dei volumi Amazon EFS

Amazon EFS archivia i dati in modo ridondante su più zone di disponibilità per file system che utilizzano classi di storage Standard. Se selezioni le classi di storage Amazon EFS One Zone, i tuoi dati vengono archiviati in modo ridondante all'interno di un'unica zona di disponibilità. Inoltre, Amazon EFS è progettato per fornire il 99,99999% (11 9) di durabilità in un determinato anno.

Come per qualsiasi ambiente, è consigliabile disporre di un backup e creare protezioni contro l'eliminazione accidentale. Per i dati di Amazon EFS, tale best practice include un backup funzionante e regolarmente testato utilizzando AWS Backup. I file system che utilizzano le classi di storage Amazon EFS One Zone sono configurati per eseguire automaticamente il backup dei file per impostazione predefinita al momento della creazione del file system, a meno che non si scelga di

disabilitare questa funzionalità. Per ulteriori informazioni, consulta la sezione Protezione dei dati per Amazon EFS nella Amazon Elastic File System User Guide.

Specificare un file system Amazon EFS in una definizione di attività Amazon ECS

Per utilizzare i volumi del file system Amazon EFS per i container, è necessario specificare le configurazioni di volume e punto di montaggio nella definizione di attività. Il seguente frammento JSON della definizione di attività illustra la sintassi degli oggetti `volumes` e `mountPoints` per un container.

```
{
    "containerDefinitions": [
        {
            "name": "container-using-efs",
            "image": "amazonlinux:2",
            "entryPoint": [
                "sh",
                "-c"
            ],
            "command": [
                "ls -la /mount/efs"
            ],
            "mountPoints": [
                {
                    "sourceVolume": "myEfsVolume",
                    "containerPath": "/mount/efs",
                    "readOnly": true
                }
            ]
        }
    ],
    "volumes": [
        {
            "name": "myEfsVolume",
            "efsVolumeConfiguration": {
                "fileSystemId": "fs-1234",
                "rootDirectory": "/path/to/my/data",
                "transitEncryption": "ENABLED",
                "transitEncryptionPort": integer,
                "authorizationConfig": {
                    "accessPointId": "fsap-1234",
                    "iam": "ENABLED"
                }
```

```
            }
        }
    ]
}
```

`efsVolumeConfiguration`

Tipo: oggetto

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

-Tipo: stringa

Campo obbligatorio: no

La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omesso, 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 omesso o impostato su `/` per applicare il percorso impostato sul punto di accesso EFS.

`transitEncryption`

-Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Specifica se abilitare o meno la crittografia per i dati Amazon EFS in transito tra l'host Amazon ECS e il server Amazon EFS. Se si utilizza l'autorizzazione IAM di Amazon EFS, è necessario abilitare la crittografia di transito. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di `DISABLED`. Per ulteriori informazioni, consulta Crittografia dei dati in transito nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: integer

Campo obbligatorio: no

La porta da utilizzare per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non si specifica una porta di crittografia di transito, verrà utilizzata la strategia di selezione della porta utilizzata dall'helper per il montaggio di Amazon EFS. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: oggetto

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

- Tipo: stringa

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory root specificato in `efsVolumeConfiguration` deve essere omesso o impostato su `/` per applicare il percorso impostato sul punto di accesso EFS. Se si utilizza un punto di accesso, la crittografia di transito deve essere abilitata in `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta Utilizzo dei punti di accesso Amazon EFS nella Guida per l'utente di Amazon Elastic File System.

`iam`

- Tipo: stringa

  Valori validi: `ENABLED` | `DISABLED`

  Campo obbligatorio: no

  Specifica se utilizzare 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.

Configurazione dei file system Amazon EFS per Amazon ECS tramite la console

Scopri come usare i file system Amazon Elastic File System (Amazon EFS) con Amazon ECS.

Fase 1: Creazione di un cluster Amazon ECS

Utilizza la procedura seguente per creare un cluster Amazon ECS.

Creazione di un nuovo cluster (console Amazon ECS)

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

1. Apri la console all'indirizzo 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 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 istanze Amazon EC2 al cluster, espandi Infrastruttura e seleziona Istanze Amazon EC2. 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.
     - In EC2 instance type (Tipo di istanza EC2), selezionare `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 Crea.

Fase 2: creazione di un gruppo di sicurezza per le istanze Amazon EC2 e il file system Amazon EFS

In questa fase, crea un gruppo di sicurezza per le istanze Amazon EC2 che consenta il traffico di rete in entrata sulla porta 80 e per il file system Amazon EFS che consenta l'accesso in entrata dalle istanze di container.

Crea un gruppo di sicurezza per le istanze Amazon EC2 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 Create a security group 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 alla pagina https://console.aws.amazon.com/efs/.
2. Scegliere Create file system (Crea file system).
3. Inserisci un nome per il file system, quindi scegli il VPC in cui sono ospitate le istanze di container. Di default, ciascuna sottorete nel VPC specificato riceve un target di montaggio che utilizza il gruppo di sicurezza predefinito per tale VPC. Quindi, scegli Personalizza.

   > **Note**
   >
   > Questo tutorial presuppone che il file system Amazon EFS, il cluster Amazon ECS, le istanze di container e le attività si trovino nello stesso VPC. Per ulteriori informazioni sul montaggio di un file system da un VPC diverso, consulta la procedura dettagliata: montare un file system da un VPC diverso nella Amazon EFS User Guide.

4. Nella pagina Impostazioni del file system, configura le impostazioni facoltative, quindi in Impostazioni delle prestazioni, scegli Ottimizzazione come modalità di velocità di trasmissione effettiva per il file system. Dopo aver configurato le impostazioni, seleziona Avanti.

   a. (Opzionale) Aggiungi tag al tuo file system. Ad esempio, puoi specificare un nome di file system univoco inserendolo nella colonna Value (Valore) accanto alla chiave Name (Nome).

   b. (Facoltativo) Abilita la gestione del ciclo di vita per risparmiare denaro in storage con accessi non frequenti. Per ulteriori informazioni, consulta Gestione del ciclo di vita EFS nella Amazon Elastic File System User Guide.

   c. (Facoltativo) Abilita la crittografia. Seleziona la casella di controllo per abilitare la crittografia del file system Amazon EFS inattivo.

5. Nella pagina Accesso di rete, in Target di montaggio, sostituisci la configurazione del gruppo di sicurezza esistente per ogni zona di disponibilità con il gruppo di sicurezza creato per il file system in Fase 2: creazione di un gruppo di sicurezza per le istanze Amazon EC2 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'istanza Amazon EC2 e aggiungi contenuto. 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.

Come creare un'istanza di Amazon EC2 e montare il file system Amazon EFS

1. Aprire la console Amazon EC2 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: creazione di un gruppo di sicurezza per le istanze Amazon EC2 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, connettiti all'istanza Amazon EC2 e aggiungi contenuti al file system Amazon EFS.

Come connettersi all'istanza Amazon EC2 e aggiungere contenuti al file system Amazon EFS

1. Esegui l'SSH sull'istanza Amazon EC2 creata. Per ulteriori informazioni, consulta Connect to Your Linux Instance 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 all'indirizzo 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, 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": "nginx"
        }
    ],
    "volumes": [
        {
            "name": "efs-html",
            "efsVolumeConfiguration": {
                "fileSystemId": "fs-1324abcd",
                "transitEncryption": "ENABLED"
            }
        }
    ],
    "family": "efs-tutorial",
    "executionRoleArn":"arn:aws::iam::111122223333:role/ecsTaskExecutionRole"
}
```

> **Note**
>
> Puoi aggiungere le seguenti autorizzazioni al tuo ruolo IAM di esecuzione delle attività di Amazon ECS per consentire all'agente Amazon ECS di individuare e montare un file system Amazon EFS su un'attività all'avvio.
>
> - `elasticfilesystem:ClientMount`
> - `elasticfilesystem:ClientWrite`
> - `elasticfilesystem:DescribeMountTargets`
> - `elasticfilesystem:DescribeFileSystems`

5. Scegli 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 all'indirizzo 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	a. Nella pagina Cluster, seleziona il cluster in cui creare il servizio. b. Dalla scheda Processi, scegli Esegui nuovo processo.
Tipo di avvio	a. Nella pagina Task (Attività), scegli la definizione di attività. b. Se è presente più di una revisione, selezionane una. c. Scegli Create (Crea), Run task (Esegui attività).

3. (Facoltativo) Scegli come viene distribuita l'attività pianificata nell'infrastruttura cluster. Espandi Compute configuration (Configurazione di calcolo), quindi effettua le seguenti operazioni:

Metodo di distribuzione	Fasi
Tipo di avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio).

Metodo di distribuzione	Fasi
	b. Per Tipo di avvio, seleziona 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 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.

Usa FSx per volumi Windows File Server con Amazon ECS

FSx for Windows File Server fornisce file server Windows completamente gestiti, supportati da un file system Windows. Quando utilizzi FSx for Windows File Server insieme a ECS, puoi eseguire il provisioning dei processi di Windows con archiviazione file statica, persistente, distribuita, condivisa. Per ulteriori informazioni, consulta Cos'è FSx for Windows File Server?.

> **Note**
>
> Le istanze EC2 che utilizzano l'AMI completa di Windows Server 2016 ottimizzata per Amazon ECS, non supportano volumi di attività ECS di FSx for Windows File Server.

Non è possibile utilizzare i volumi FSx for Windows File Server in una configurazione di contenitori Windows su Fargate. È invece possibile modificare i contenitori per montarli all'avvio.

Puoi utilizzare FSx for Windows File Server per implementare carichi di lavoro di Windows che richiedono l'accesso a uno storage esterno condiviso, uno storage regionale disponibile o uno storage a velocità di trasmissione effettiva elevata. Puoi montare uno o più volumi del file system FSx for Windows File Server in un contenitore Amazon ECS eseguito su un'istanza Amazon ECS Windows. Puoi condividere i volumi del file system FSx for Windows File Server tra più contenitori Amazon ECS all'interno di una singola attività Amazon ECS.

Per abilitare l'utilizzo di FSx for Windows File Server con ECS, includi l'ID file system FSx for 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'istanza EC2 Windows per ECS unita a un dominio valido. Può essere ospitato da un AWS Directory Service for Microsoft Active DirectoryActive Directory locale o da Active Directory ospitato autonomamente su Amazon EC2.
- Un parametro AWS Secrets Manager segreto o di Systems Manager che contiene le credenziali utilizzate per accedere al dominio Active Directory e collegare il file system FSx for 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 i file system FSx for Windows File Server per Amazon ECS.

Considerazioni

Durante l'utilizzo dei volumi FSx for Windows File Server è opportuno considerare le informazioni riportate di seguito:

- FSx for Windows File Server con Amazon ECS supporta solo istanze Amazon EC2 di Windows. Le istanze Amazon EC2 di Linux non sono supportate.
- FSx for Windows File Server con Amazon ECS non supporta AWS Fargate.
- FSx for Windows File Server con Amazon ECS con la modalità di rete `awsvpc` richiede la versione `1.54.0` o successiva dell'agente del container.

- Il numero massimo di lettere di unità che possono essere utilizzate per un processo Amazon ECS è 23. A ogni processo con un volume FSx for 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.
- I processi in genere vengono eseguiti solo nello stesso VPC del file system FSx for Windows File Server. Tuttavia, è possibile avere supporto tra vPC se esiste una connettività di rete stabilita tra il VPC del cluster Amazon ECS e il file system FSx for Windows File Server tramite peering VPC.
- Puoi controllare l'accesso a un file system FSx for Windows File Server a livello di rete configurando i gruppi di sicurezza VPC. Solo le attività ospitate su istanze EC2 unite al dominio Active Directory con gruppi di sicurezza Active Directory correttamente configurati possono accedere alla condivisione di file FSx for Windows File Server. Se i gruppi di sicurezza non sono configurati correttamente, Amazon ECS non riesce ad avviare l'attività con il seguente messaggio di errore:`unable to mount file system fs-id`.»
- FSx for Windows File Server è integrato AWS Identity and Access Management con (IAM) per controllare le azioni che gli utenti e i gruppi IAM possono eseguire su risorse specifiche di FSx for Windows File Server. Con l'autorizzazione client, i clienti possono definire ruoli IAM che consentono o negano l'accesso a file system FSx for Windows File Server specifici, facoltativamente richiedono l'accesso in sola lettura e, sempre facoltativamente, consentono o non consentono l'accesso root al file system dal client. Per ulteriori informazioni, consulta Sicurezza nella Guida per l'utente di Amazon FSx Windows.

Best practice per l'utilizzo di FSx for Windows File Server con Amazon ECS

Prendi nota dei seguenti consigli sulle best practice quando usi FSx for Windows File Server con Amazon ECS.

Sicurezza e controlli di accesso per FSx for Windows File Server

FSx for Windows File Server offre le seguenti funzionalità di controllo degli accessi che è possibile utilizzare per garantire che i dati archiviati in un file system FSx for Windows File Server siano sicuri e accessibili solo dalle applicazioni che ne hanno bisogno.

Crittografia dei dati per volumi FSx for Windows File Server

FSx for Windows File Server supporta due forme di crittografia per i file system. Sono la crittografia dei dati in transito e la crittografia a riposo. La crittografia dei dati in transito è supportata nelle condivisioni di file mappate su un'istanza di contenitore che supporta il protocollo SMB 3.0 o versioni successive. La crittografia dei dati inattivi viene abilitata automaticamente durante la creazione di un file system Amazon FSx. Amazon FSx crittografa automaticamente i dati in transito utilizzando la crittografia SMB quando accedi al file system senza la necessità di modificare le applicazioni. Per ulteriori informazioni, consulta la sezione Crittografia dei dati in Amazon FSx nella Guida per l'utente di Amazon FSx for Windows File Server.

Usa gli ACL di Windows per il controllo degli accessi a livello di cartella

L'istanza Windows Amazon EC2 accede alle condivisioni di file Amazon FSx utilizzando le credenziali di Active Directory. Utilizza elenchi di controllo degli accessi (ACL) standard di Windows per un controllo granulare degli accessi a livello di file e cartella. È possibile creare più credenziali, ognuna per una cartella specifica all'interno della condivisione associata a un'attività specifica.

Nell'esempio seguente, l'attività ha accesso alla cartella `App01` utilizzando una credenziale salvata in Secrets Manager. Il suo Amazon Resource Name (ARN) è. `1234`

```
"rootDirectory": "\\path\\to\\my\\data\App01",
"credentialsParameter": "arn-1234",
"domain": "corp.fullyqualified.com",
```

In un altro esempio, un'attività ha accesso alla cartella `App02` utilizzando una credenziale salvata in Secrets Manager. Il suo ARN è 6789.

```
"rootDirectory": "\\path\\to\\my\\data\App02",
"credentialsParameter": "arn-6789",
"domain": "corp.fullyqualified.com",
```

Specificare un file system FSx for Windows File Server in una definizione di attività Amazon ECS

Per utilizzare i volumi del file system FSx for Windows File Server per i container, è necessario specificare le configurazioni del volume e del 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": [
        {
            "entryPoint": [
                "powershell",
                "-Command"
            ],
            "portMappings": [],
            "command": ["New-Item -Path C:\\fsx-windows-dir\\index.html -ItemType file
 -Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>It Works!</h2> <p>You are using Amazon
 FSx for Windows File Server file system for persistent container storage.</p>' -
Force"],
            "cpu": 512,
            "memory": 256,
            "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
            "essential": false,
            "name": "container1",
            "mountPoints": [
                {
                    "sourceVolume": "fsx-windows-dir",
                    "containerPath": "C:\\fsx-windows-dir",
                    "readOnly": false
                }
            ]
        },
        {
            "entryPoint": [
                "powershell",
                "-Command"
            ],
            "portMappings": [
                {
                    "hostPort": 443,
                    "protocol": "tcp",
                    "containerPort": 80
                }
            ],
            "command": ["Remove-Item -Recurse C:\\inetpub\\wwwroot\\* -Force; Start-
Sleep -Seconds 120; Move-Item -Path C:\\fsx-windows-dir\\index.html -Destination C:\
\inetpub\\wwwroot\\index.html -Force; C:\\ServiceMonitor.exe w3svc"],
            "mountPoints": [
                {
```

```
                    "sourceVolume": "fsx-windows-dir",
                    "containerPath": "C:\\fsx-windows-dir",
                    "readOnly": false
                }
            ],
            "cpu": 512,
            "memory": 256,
            "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-
ltsc2019",
            "essential": true,
            "name": "container2"
        }
    ],
    "family": "fsx-windows",
    "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
    "volumes": [
        {
            "name": "fsx-windows-dir",
            "fsxWindowsFileServerVolumeConfiguration": {
                "fileSystemId": "fs-0eeb5730b2EXAMPLE",
                "authorizationConfig": {
                    "domain": "example.com",
                    "credentialsParameter": "arn:arn-1234"
                },
                "rootDirectory": "share"
            }
        }
    ]
}
```

`FSxWindowsFileServerVolumeConfiguration`

Tipo: oggetto

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizza il file system FSx for Windows File Server per l'archiviazione delle attività.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system FSx for Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

Campo obbligatorio: sì

La directory all'interno del file system FSx for Windows File Server da montare come directory root all'interno dell'host.

`authorizationConfig`

`credentialsParameter`

Tipo: stringa

Campo obbligatorio: sì

Le opzioni delle credenziali di autorizzazione:

- L'Amazon Resource Name (ARN) di un segreto di Secrets Manager.
- Amazon Resource Name (ARN) di un parametro di Systems Manager.

`domain`

Tipo: stringa

Campo obbligatorio: sì

Un nome di dominio completo ospitato da una directory AWS Directory Service for Microsoft Active Directory(AWS Managed Microsoft AD) o da un Active Directory EC2 ospitato autonomamente.

Metodi per l'archiviazione delle credenziali del volume FSx for Windows File Server

Esistono due metodi diversi per archiviare le credenziali da utilizzare con il parametro delle credenziali.

- AWS Secrets Manager segreto

  Questa credenziale può essere creata nella AWS Secrets Manager console utilizzando la categoria segreta Altro tipo. Si aggiunge una riga per ogni coppia chiave/valore, nome utente/amministratore e password/*`password`*.
- Parametro di Systems Manager

Queste credenziali possono essere create nella console dei parametri di Systems Manager immettendo del testo nel modulo illustrato nel frammento di codice di esempio seguente.

```
{
  "username": "admin",
  "password": "password"
}
```

`credentialsParameter` nel parametro `FSxWindowsFileServerVolumeConfiguration` della definizione di attività conterrà l'ARN del segreto o l'ARN del parametro di Systems Manager. Per ulteriori informazioni, consulta Cos'è AWS Secrets Manager nella Guida per l'utente di Secrets Manager e Archivio parametri di Systems Manager nella Guida per l'utente di Systems Manager.

Scopri come configurare i file system FSx for Windows File Server per Amazon ECS

Scopri come avviare un'istanza Windows ottimizzata per Amazon ECS che ospita un file system FSx for Windows File Server e contenitori in grado di accedere al file system. A tale scopo, è innanzitutto necessario creare un Microsoft Active Directory AWS Directory Service AWS gestito. Quindi, crei un file system e un cluster FSx for Windows File Server File Server con un'istanza Amazon EC2 e una definizione di attività. Configura la definizione di attività per i container per utilizzare il file system FSx for Windows File Server. Infine, esegui il test del file system.

Ogni volta che si avvia o elimina Active Directory o il file system FSx for Windows File Server occorrono da 20 a 45 minuti. 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 coppia di chiavi `PEM` per la connessione all'istanza Windows di EC2 tramite l'accesso RDP. Per ulteriori informazioni sulla creazione di coppie di chiavi, consulta Coppie di chiavi Amazon EC2 e istanze Windows nella Guida per l'utente di Amazon EC2 per le istanze Windows.
- Un VPC con almeno una sottorete pubblica e una sottorete privata e un gruppo di sicurezza. È possibile utilizzare il VPC di default. Non è necessario un gateway o un dispositivo NAT. AWS Directory Service non supporta Network Address Translation (NAT) con Active Directory. Affinché ciò funzioni, Active Directory, il file system FSx per Windows File Server, il cluster ECS e

l'istanza EC2 devono trovarsi all'interno del VPC. Per ulteriori informazioni relative ai VPC e alle Active Directory, consulta Configurazioni della procedura guidata per la console Amazon VPC e Prerequisiti di AWS Managed Microsoft AD.

- Le autorizzazioni IAM ecsInstanceRole e ecsTaskExecution Role sono associate al tuo account. Questi ruoli collegati ai servizi consentono ai servizi di effettuare chiamate API e accedere a container, segreti, directory e server di file per tuo conto.

Fase 1: Creazione dei ruoli di accesso IAM

Crea un cluster con la AWS Management Console.

1. Verifica se ne possiedi uno ecsInstanceRole e scopri come puoi crearne uno se non ne hai uno. Ruolo IAM delle istanze di container Amazon ECS
2. Consigliamo di personalizzare le policy dei ruoli per ottenere autorizzazioni minime in un ambiente di produzione effettivo. Per completare questo tutorial, verifica che la seguente politica AWS gestita sia allegata al tuo ecs. InstanceRole Collega la policy, se non è già collegata.

   - Ruolo EC2 di Amazon EC2 ContainerServicefor
   - Amazon SSM ManagedInstance Core
   - Accesso ad DirectoryService AmazonSSM

   Per AWS allegare politiche 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 Ruolo IAM di esecuzione di attività Amazon ECS disponi di un ecs TaskExecutionRole e scopri come puoi crearne uno se non ne hai uno.

   Consigliamo di personalizzare le policy dei ruoli per ottenere autorizzazioni minime in un ambiente di produzione effettivo. Allo scopo di seguire questo tutorial, verifica che le seguenti politiche AWS gestite siano allegate al tuo ruolo ecs. TaskExecution Collega le policy, se non

sono già collegate. Utilizza la procedura indicata nella sezione precedente per allegare le politiche AWS gestite.

- SecretsManagerReadWrite
- AmazonF SxRead OnlyAccess
- Accesso ad AmazonSSM ReadOnly
- Amazonecs TaskExecution RolePolicy

Fase 2: Creazione di Windows Active Directory (AD)

1. Segui i passaggi descritti in Create Your AWS Managed AD Directory nella AWS Directory Service Administration Guide. Utilizza il VPC designato per questo tutorial. Nella fase 3 di Creazione della tua AWS Managed AD Directory, salva il nome utente e la password da utilizzare nella fase successiva. Prendi nota anche del nome di dominio completo per le operazioni successive. Puoi 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 Getting Started with AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager.
   a. Apri la console Secrets Manager.
   b. Fai clic su Archivia un nuovo segreto.
   c. Seleziona Altro tipo di segreti.
   d. Per Chiave/valore segreto, nella prima riga, crea una chiave **`username`** con valore **`admin`**. Fai clic su + Aggiungi riga.
   e. Nella nuova riga, crea una chiave **`password`**. Per ulteriori informazioni, digitate la password inserita nel passaggio 3 di Create Your AWS Managed AD Directory.
   f. Fai clic sul pulsante Successivo.
   g. Specifica un nome e una descrizione per il segreto. Fai clic su Next (Successivo).
   h. Fai clic su Next (Successivo). Fai clic su Archivia.
   i. Dall'elenco nella pagina Segreti, seleziona il segreto appena creato.
   j. Salva l'ARN del nuovo segreto per utilizzarlo nelle fasi seguenti.
   k. Durante la creazione di Active Directory, è possibile procedere alla fase successiva.

Fase 3: Verifica e aggiornamento del gruppo di sicurezza

In questa fase è possibile verificare e aggiornare le regole per il gruppo di sicurezza in uso. Pertanto è possibile utilizzare il gruppo di sicurezza di default creato per il VPC.

Verifica e aggiorna il gruppo di sicurezza.

È necessario creare o modificare il gruppo di sicurezza per inviare i dati da e verso le porte, operazioni descritte in Gruppi di sicurezza di Amazon VPC nella Guida per l'utente di FSx for 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 di inviare traffico da e verso il file system FSx for Windows File Server, Active Directory e l'istanza ECS come richiesto. 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.

Type	Protocollo	Intervallo porte	Origine
Tutto il traffico	Tutti	Tutti	*sg-securitygroup*
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.

Type	Protocollo	Intervallo porte	Destinazione

Type	Protocollo	Intervallo porte	Destinazione
Tutto il traffico	Tutti	Tutti	0.0.0.0/0

1. Apri la Console EC2 e seleziona Gruppi di sicurezza dal menu sulla 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'istanza EC2 più avanti in questa esercitazione, modifica l'origine RDP della regola in ingresso con l'indirizzo IP pubblico dell'istanza EC2 come descritto in Connessione all'istanza Windows nella Guida per l'utente di Amazon EC2 per le istanze Windows.

Fase 4: Creazione di un file system FSx for Windows File Server

Dopo la verifica e l'aggiornamento del gruppo di sicurezza e dopo aver creato Active Directory nello stato attivo, crea il file system FSx for Windows File Server nello stesso VPC di Active Directory. Completa la procedura seguente per creare un file system FSx for Windows File Server per i processi di Windows.

Creare il tuo primo file system.

1. Apri la console Amazon FSx.
2. Nel pannello di controllo, scegli Crea file system per avviare la procedura guidata di creazione del file system.
3. Sulla pagina Seleziona tipo di file system, seleziona FSx for Windows File Server, quindi seleziona Successivo. Viene visualizzata la pagina Crea file system.
4. Nella sezione Dettagli file system, specifica un nome per il file system. La denominazione dei file system ne semplifica la ricerca e la gestione. Puoi utilizzare fino a 256 caratteri Unicode. I caratteri consentiti sono lettere, numeri, spazi e caratteri speciali segno più (+), segno meno (-), segno uguale (=), punto (.), trattino basso (_), due punti (:) e barra (/).

5. Per Tipo di implementazione, scegli Single-AZ per implementare un file system distribuito in una zona di disponibilità singola. Single-AZ 2 è l'ultima generazione di file system a zona di disponibilità singola e supporta l'archiviazione SSD e HDD.
6. Per Tipo di archiviazione, scegli HDD.
7. In Capacità di archiviazione, specifica la capacità di archiviazione minima.
8. Mantieni Capacità di velocità effettiva sul valore di default.
9. Nella sezione Rete e sicurezza, scegli lo stesso Amazon VPC che hai scelto per la tua AWS Directory Service directory.
10. Per Gruppi di sicurezza VPC, scegli il gruppo di sicurezza verificato in Fase 3: Verifica e aggiornamento del gruppo di sicurezza.
11. Per Autenticazione Windows, scegli AWS Managed Microsoft Active Directory, quindi seleziona la tua directory AWS Directory Service dall'elenco.
12. Per Crittografia, mantieni l'impostazione di default Chiave di crittografia su aws/fsx (default).
13. Mantieni le impostazioni di default per Preferenze di manutenzione.
14. Fai clic sul pulsante Successivo.
15. Rivedi la configurazione del file system riportata nella pagina Crea file system. Come riferimento, prendi nota delle impostazioni del file system che è possibile modificare dopo la sua creazione. Scegliere Create file system (Crea file system).
16. Prendi nota dell'ID file system. Sarà utile in una fase successiva.

    Puoi passare alla procedura successiva per creare un cluster e un'istanza EC2 durante la creazione del file system FSx for Windows File Server.

Fase 5: Creazione di un cluster Amazon ECS

Creazione di un cluster tramite la console Amazon ECS

1. Apri la console all'indirizzo 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 Configurazione del cluster, per Nome del cluster immetti windows-fsx-cluster.
6. Espandi l'infrastruttura, cancella AWS Fargate (senza server) e seleziona le istanze Amazon EC2.

- 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.
  - In Tipo di istanza EC2, scegli t2.medium o t2.micro.

7. Scegli Crea.

Fase 6: creazione di un'istanza Amazon EC2 ottimizzata per 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 il nome dell'AMI per la regione che ospita il VPC. Per ulteriori informazioni, consulta Recupero dei metadati dell'AMI ottimizzata per Amazon ECS.
2. Utilizza la console di Amazon EC2 per avviare l'istanza.
   a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
   b. Seleziona la Regione da utilizzare nella barra di navigazione.
   c. Da Pannello di controllo EC2, scegli Avvia istanza.
   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 specifici 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. Questo dominio si unisce all'AD quando viene avviata l'istanza EC2.

l. In Dettagli avanzati, per il profilo dell'istanza IAM, scegli ecs. InstanceRole

m. Configura la tua istanza di container Amazon ECS con i seguenti dati utente. In Advanced Details (Dettagli avanzati), incolla il seguente script nel campo User data (Dati utente), sostituendo *cluster_name* con il nome del tuo 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
4. Nel pannello di navigazione, seleziona Cluster, quindi scegli windows-fsx-cluster.
5. Scegli la scheda Infrastruttura e verifica che l'istanza sia stata registrata nel cluster windows-fsx-cluster.

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. Il processo avvia due container che hanno accesso al file system FSx. 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 all'indirizzo 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, sostituisci i valori per il ruolo di esecuzione dell'attività e i dettagli sul file system FSx, 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 il processo, verifica che lo stato del file system FSx for 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione, seleziona Cluster, quindi scegli windows-fsx-cluster.
3. Seleziona la scheda Attività, quindi Esegui nuova attività.
4. In Tipo di avvio, scegli EC2.
5. In Configurazione dell'implementazione, per Definizione dell'attività, scegli fsx-windows, quindi scegli Crea.
6. Quando lo stato dell'attività è IN ESECUZIONE, scegli l'ID attività.
7. In Container, quando lo stato di container1 è ARRESTATO, seleziona container2 per visualizzarne i dettagli.
8. In Dettagli del container per container2, seleziona Associazioni di rete e fai clic sull'indirizzo IP esterno associato al container. Si aprirà il browser e sarà visualizzato il seguente messaggio.

```
Amazon ECS Sample App
It Works!
You are using Amazon FSx for Windows File Server file system for persistent
 container storage.
```

> **Note**
>
> Possono essere necessari alcuni minuti per visualizzare il messaggio. Se dopo qualche minuto non viene visualizzato il messaggio, assicurati che non stai utilizzando una VPN e che il gruppo di sicurezza per l'istanza di container consenta il traffico di rete in entrata sulla porta 443.

Fase 9: Pulizia

> **Note**
>
> L'eliminazione del file system FSx for Windows File Server o dell'AD richiede da 20 a 45 minuti. Prima di avviare le operazioni di eliminazione dell'AD, è necessario attendere il completamento delle operazioni di eliminazione del file system FSx for Windows File Server.

Cancella il file system FSx per Windows File Server.

1. Apri la console Amazon FSx.
2. Scegli il pulsante di opzione a sinistra del file system FSx per Windows File Server appena creato.
3. Scegli Azioni.
4. Seleziona Elimina file system.

Elimina l'AD.

1. Apri la AWS Directory Service console.
2. Scegli il pulsante di opzione a sinistra dell'AD appena creato.
3. Scegli Azioni.
4. Seleziona Elimina directory.

Elimina il cluster.

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione, seleziona Cluster, quindi scegli fsx-windows-cluster.
3. Scegli Delete cluster (Elimina cluster).
4. Inserisci la frase, quindi scegli Elimina.

Termina l'istanza EC2.

1. Aprire la console di Amazon EC2.
2. Nel menu a sinistra, seleziona Istanze.
3. Seleziona la casella a sinistra dell'istanza EC2 creata.
4. Fai clic su Stato istanza, Termina istanza.

Elimina il segreto.

1. Apri la console Secrets Manager.
2. Seleziona il segreto creato per questa procedura.
3. Fai clic su Operazioni.

4. Seleziona Elimina segreto.

Usa i volumi Docker con Amazon ECS

Quando utilizzi i volumi Docker, puoi usare il driver `local` integrato o un driver di volumi di terze parti. I volumi Docker sono gestiti da Docker e viene creata una directory in `/var/lib/docker/volumes` sull'istanza di container che contiene i dati del volume.

Per usare i volumi Docker, specifica `dockerVolumeConfiguration` nella definizione di attività. Per ulteriori informazioni, consulta l'articolo relativo all'utilizzo dei volumi.

Alcuni casi di utilizzo comune per i volumi Docker sono i seguenti:

- Offrire volumi di dati persistenti per l'utilizzo con i container
- Condividere un volume di dati definito in diverse posizioni su differenti container nella stessa istanza di container
- Definire un volume di dati vuoto, non persistente e montarlo su più container all'interno della stessa attività
- Per fornire un volume di dati per l'attività gestita da un driver di terze parti

Considerazioni sull'utilizzo dei volumi Docker

Quando usi volumi Docker, tieni presenti le considerazioni seguenti:

- I volumi Docker sono supportati solo se si utilizza il tipo di avvio EC2 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
    ```

Specificare un volume Docker in una definizione di attività Amazon ECS

Prima che i container possano utilizzare i volumi di dati, è necessario specificare le configurazioni del punto di montaggio e del volume nella definizione di attività. Questa sezione descrive la configurazione del volume per un container. Per le attività che usano un volume Docker, specifica `dockerVolumeConfiguration`. Per le attività che usano un volume host di montaggio vincolato, specifica `host` e facoltativamente `sourcePath`.

Il seguente JSON della definizione di attività illustra la sintassi degli oggetti `volumes` e `mountPoints` per un container.

```
{
    "containerDefinitions": [
        {
            "mountPoints": [
                {
                    "sourceVolume": "string",
                    "containerPath": "/path/to/mount_volume",
                    "readOnly": boolean
                }
            ]
        }
    ],
    "volumes": [
        {
            "name": "string",
            "dockerVolumeConfiguration": {
                "scope": "string",
                "autoprovision": boolean,
                "driver": "string",
                "driverOpts": {
                    "key": "value"
                },
                "labels": {
                    "key": "value"
                }
            }
        }
    ]
}
```

`name`

- Tipo: stringa

  Campo obbligatorio: no

  Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e caratteri di sottolineatura (-). _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume` `mountPoints`

`dockerVolumeConfiguration`

Tipo: oggetto DockerVolumedi configurazione

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker. I volumi Docker sono supportati solo durante l'esecuzione di attività su istanze EC2. I contenitori Windows supportano solo l'uso del driver. `local` Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

- Tipo: stringa

  Valori validi: `task` | `shared`

  Campo obbligatorio: no

  L'ambito del volume Docker che determina il suo ciclo di vita. I volumi Docker che rientrano nell'ambito `task` vengono automaticamente assegnati all'avvio del processo e distrutti quando il processo viene arrestato. I volumi Docker che vengono definiti come `shared` vengono mantenuti dopo l'arresto del processo.

`autoprovision`

  Tipo: Booleano

  Valore predefinito: `false`

  Campo obbligatorio: no

  Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo viene utilizzato solo se `scope` è`shared`. Se `scope` è`task`, allora questo parametro deve essere omesso o impostato `false` su.

`driver`

- Tipo: stringa

  Campo obbligatorio: no

  Il driver del volume Docker da utilizzare. Il valore del driver deve corrispondere al nome del driver fornito da Docker perché questo nome viene utilizzato per il posizionamento delle attività. Se il driver è stato installato utilizzando la CLI del plug-in Docker, `docker plugin ls` utilizzalo per recuperare il nome del driver dall'istanza del contenitore. Se il driver è stato installato utilizzando un altro metodo, utilizza Docker plugin discovery per recuperare il nome del driver. Per ulteriori informazioni, consulta l'argomento relativo al rilevamento del plug-in Docker. Questo parametro fa riferimento a `Driver` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--driver` in `docker volume create`.

`driverOpts`

- Tipo: stringa

  Campo obbligatorio: no

  Una mappa delle opzioni specifiche del driver Docker da esaminare. Questo parametro fa riferimento a `DriverOpts` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--opt` in `docker volume create`.

`labels`

- Tipo: stringa

  Campo obbligatorio: no

  Metadati personalizzati da aggiungere al volume Docker. Questo parametro fa riferimento a `Labels` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--label` in `docker volume create`.

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato a `Volumes` nella sezione Create a container di Docker Remote API e l'opzione `--volume` a docker run.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il nome del volume da montare.

`containerPath`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

  Tipo: Booleano

  Campo obbligatorio: no

  Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

Esempi di volumi Docker

Per fornire spazio di archiviazione effimero per un contenitore utilizzando un volume Docker

In questo esempio, un container utilizza un volume di dati vuoto che viene smaltito al termine dell'attività. Ad esempio, potresti avere un container che deve accedere alla posizione di storage di alcuni file temporanei durante un'attività. Questa attività può essere eseguita utilizzando un volume Docker.

1. Nella sezione `volumes` della definizione di attività, definisci un volume di dati con i valori `name` e `DockerVolumeConfiguration`. In questo esempio, specifichiamo l'ambito come `task` in modo che il volume venga eliminato dopo l'arresto dell'attività e venga utilizzato il driver `local` incorporato.

```
"volumes": [
    {
        "name": "scratch",
        "dockerVolumeConfiguration" : {
            "scope": "task",
            "driver": "local",
            "labels": {
                "scratch": "space"
            }
        }
    }
]
```

2. Nella sezione `containerDefinitions`, definisci un container con valori `mountPoints` che faccia riferimento al nome del volume definito e al valore `containerPath` per montare il volume sul container.

```
"containerDefinitions": [
    {
        "name": "container-1",
        "mountPoints": [
            {
              "sourceVolume": "scratch",
              "containerPath": "/var/scratch"
            }
        ]
    }
]
```

Per fornire lo storage persistente per un 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"
               }
```

```
            ]
        }
    ]
```

Usa i bind mount con Amazon ECS

Con bind mount, un file o una directory su un host, ad esempio un'istanza Amazon EC2, viene montato in un contenitore. I montaggi vincolati sono supportati per le attività ospitate su istanze Fargate o Amazon EC2. I bind mount sono legati al ciclo di vita del contenitore che li utilizza. Una volta arrestati tutti i container che utilizzano un montaggio vincolato, ad esempio quando viene arrestata un'attività, i dati vengono rimossi. Per le attività ospitate su istanze Amazon EC2, i dati possono essere collegati al ciclo di vita dell'istanza Amazon EC2 host specificando un `host` valore opzionale nella definizione dell'attività. `sourcePath` Per ulteriori informazioni, consulta Utilizzo di montaggi vincolati 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 spazio di archiviazione temporaneo per i bind mount. È possibile aumentare la quantità totale di storage temporaneo fino a un massimo di 200 GiB specificando il parametro nella definizione dell'`ephemeralStorage`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 istanze Amazon EC2, quando non sono specificati i valori `host` e `sourcePath`, il daemon Docker gestisce il montaggio vincolato per tuo conto. Quando nessun container fa riferimento a questo montaggio vincolato, viene alla fine eliminato dal servizio di pulizia dell'attività dell'agente del container Amazon ECS. Per impostazione predefinita, ciò avviene tre ore dopo la chiusura del container. Tuttavia, puoi configurare questa durata con la variabile dell'agente `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION`. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS. Se è necessario che questi dati vengano conservati oltre il ciclo di vita del container, specifica un valore `sourcePath` per il montaggio vincolato.

Specificare un bind mount in una definizione di attività Amazon ECS

Per le attività di Amazon ECS ospitate su istanze Fargate o Amazon EC2, il seguente frammento JSON di definizione delle attività mostra la sintassi per la definizione e gli oggetti per `volumes` una definizione di attività. `mountPoints` `ephemeralStorage`

```
{
   "family": "",
   ...
```

```
   "containerDefinitions" : [
      {
         "mountPoints" : [
            {
               "containerPath" : "/path/to/mount_volume",
               "sourceVolume" : "string"
            }
          ],
          "name" : "string"
       }
    ],
    ...
    "volumes" : [
       {
          "name" : "string"
       }
    ],
    "ephemeralStorage": {
    "sizeInGiB": integer
    }
}
```

Per le attività Amazon ECS ospitate su istanze Amazon EC2, puoi utilizzare il parametro `host` opzionale e un `sourcePath` quando specifichi i dettagli del volume dell'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

  Campo obbligatorio: no

Nome del volume. Sono consentiti fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e caratteri di sottolineatura (). - _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume` `mountPoints`

`host`

Campo obbligatorio: no

Il parametro `host` viene utilizzato per legare il ciclo di vita del montaggio vincolato all'istanza host di Amazon EC2 anziché al processo, dove invece è archiviato. 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 istanze Amazon EC2.

`sourcePath`

- Tipo: stringa

Campo obbligatorio: no

Quando viene utilizzato il parametro `host`, specifica un `sourcePath` per dichiarare il percorso sull'istanza Amazon EC2 dell'host presentata al container. Se questo parametro è vuoto, il daemon Docker assegna automaticamente un percorso host. Se il parametro `host` contiene una posizione del file `sourcePath`, il volume di dati rimane nella posizione specificata sull'istanza Amazon EC2 dell'host finché non viene eliminato manualmente. Se il valore `sourcePath` non esiste nell'istanza Amazon EC2 dell'host, viene creato automaticamente dal daemon Docker. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato a `Volumes` nella sezione Create a container di Docker Remote API e l'opzione `--volume` a docker run.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

- Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il nome del volume da montare.

`containerPath`

- Tipo: stringa

Obbligatorio: sì, quando si utilizzano `mountPoints`

Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

`ephemeralStorage`

Tipo: oggetto

Campo obbligatorio: no

La quantità di archiviazione temporanea da allocare per il processo. Questo parametro viene utilizzato per espandere la quantità totale di storage temporaneo 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 la CLI di Copilot CloudFormation, l' AWS SDK o la CLI per specificare lo storage temporaneo per un bind mount.

Esempi di montaggio vincolato

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 di 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"
        }
      ]
    }
]
```

Come fornire un volume di dati vuoto per un container legato al ciclo di vita dell'istanza host di Amazon EC2

Per le attività ospitate su istanze Amazon EC2, puoi utilizzare i montaggi vincolati e avere i dati legati al ciclo di vita dell'istanza host di Amazon 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`. Qualsiasi file scritto con il valore `containerPath` viene scritto con il valore `sourcePath`sull'istanza host di Amazon EC2.

> **Important**
>
> Amazon ECS non sincronizza il tuo spazio di archiviazione tra le istanze Amazon EC2. I processi che utilizzano archiviazione persistente possono essere posizionati su qualsiasi istanza Amazon EC2 nel cluster che abbia capacità disponibile. Se le tue attività richiedono uno storage persistente dopo l'arresto e il riavvio, specifica sempre la stessa istanza Amazon EC2 al momento dell'avvio dell'attività con il comando start-task. AWS CLI Puoi utilizzare anche volumi Amazon EFS per l'archiviazione persistente. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.

1. Nella sezione `volumes` della definizione di attività, definisci un montaggio vincolato con i valori `name` e `sourcePath`. Nell'esempio seguente, l'istanza host di Amazon EC2 contiene dati in `/ecs/webdata` che desideri montare all'interno del container.

```
"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": "nginx",
    "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 istanze Amazon EC2, puoi definire uno o più volumi su un container e quindi utilizzare il parametro `volumesFrom` in un'altra definizione del container (all'interno della stessa attività) per montare tutti i volumi da `sourceContainer` sui relativi punti di montaggio definiti originariamente. Il parametro `volumesFrom` si applica ai volumi configurati nella definizione di attività e a quelli integrati nell'immagine con un Dockerfile.

1. (Opzionale) Per condividere un volume incorporato in un'immagine, usa l'istruzione `VOLUME` nel Dockerfile. Il seguente Dockerfile di esempio utilizza un'immagine `httpd` e quindi aggiunge un volume e lo monta su `dockerfile_volume` nella radice del documento Apache. È la cartella utilizzata dal server Web `httpd`.

   ```
   FROM httpd
   VOLUME ["/usr/local/apache2/htdocs/dockerfile_volume"]
   ```

   Puoi creare un'immagine con questo Dockerfile ed eseguirne il push a un repository, ad esempio Docker Hub, e utilizzarla nella definizione di attività. L'`my-repo/httpd_dockerfile_volume`immagine di esempio utilizzata nei passaggi seguenti è stata creata con il precedente Dockerfile.
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

Amazon ECS ti permette di controllare l'utilizzo dello spazio di memoria swap sulle istanze Amazon EC2 basate su 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. 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 out-of-memory errori quando il contenitore è sotto carico.

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 omesso, il container utilizza la configurazione di swap per l'istanza di container su cui è in esecuzione. È necessario impostare un valore `maxSwap` per il parametro `swappiness` da utilizzare.

`swappiness`

In questo modo è possibile ottimizzare il comportamento swappiness di memoria di un container. Un valore `swappiness` di `0` fa sì che swap non si verifichi se non richiesto. Un valore `swappiness` di `100` produrrà lo swap delle pagine in modo aggressivo. I valori accettati sono numeri interi compresi tra `0` e `100`. Se il parametro `swappiness` non è specificato, viene utilizzato un valore predefinito `60`. Se non viene specificato un valore per `maxSwap`, questo parametro verrà ignorato. Questo parametro è mappato all'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'istanza di Amazon EC2 che ospita le attività per consentire ai container di utilizzarlo. Per impostazione predefinita, le AMI ottimizzate per Amazon ECS 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 memoria per funzionare come spazio di swap in un'istanza Amazon EC2 utilizzando un file di swap? .
- I parametri di definizione del container dello spazio di swap sono supportati solo per le definizioni di attività che specificano il tipo di avvio EC2. Questi parametri non sono supportati per le definizioni di attività destinate esclusivamente all'utilizzo di Amazon ECS su Fargate.
- Questa caratteristica è supportata solo per i container Linux. Al momento i container Windows non sono supportati.
- Se i parametri `maxSwap` e `swappiness` di definizione del container vengono omessi da una definizione di attività, ogni container avrà un valore `swappiness` predefinito di `60`. Inoltre, l'utilizzo totale dello swap è limitato a due volte la memoria del contenitore.
- Se utilizzi le attività su Amazon Linux 2023, il parametro `swappiness` non è supportato.

Differenze nella definizione delle attività di Amazon ECS per il tipo di lancio di Fargate

Per utilizzare Fargate, è necessario configurare la definizione dell'attività in modo da utilizzare il tipo di avvio Fargate. Ci sono altre considerazioni sull'utilizzo di Fargate.

Parametri di definizione di attività

I processi che utilizzano il tipo di avvio Fargate non supportano tutti i parametri di definizione dei processi di Amazon ECS disponibili. Alcuni parametri non sono supportati, mentre altri si comportano diversamente con i processi Fargate.

I parametri di definizione dei processi seguenti non sono validi nei processi Fargate:

- `disableNetworking`
- `dnsSearchDomains`
- `dnsServers`
- `dockerSecurityOptions`
- `extraHosts`
- `gpu`
- `ipcMode`
- `links`
- `placementConstraints`
- `privileged`
- `maxSwap`
- `swappiness`

I seguenti parametri di definizione dei processi sono validi in Fargate, ma hanno delle limitazioni da considerare:

- `linuxParameters`: quando si indicano opzioni specifiche di Linux che vengono applicate al container, in `capabilities` l'unica capacità che puoi aggiungere è `CAP_SYS_PTRACE`. I parametri `devices`, `sharedMemorySize` e `tmpfs` non sono supportati. Per ulteriori informazioni, consulta Parametri Linux.
- `volumes`: I processi Fargate supportano solo volumi host di montaggi vincolati, quindi il parametro `dockerVolumeConfiguration` non è supportato. Per ulteriori informazioni, consulta Volumi.
- `cpu`: per container Windows su AWS Fargate, il valore non può essere inferiore a 1 vCPU.

Per assicurarti che la definizione di attività sia valida per l'utilizzo con Fargate, puoi specificare quanto riportato di seguito quando registri la definizione di attività:

- Nel campo Richiede compatibilità AWS Management Console, specificare. `FARGATE`
- Nel AWS CLI, specificare l'opzione. `--requires-compatibilities`
- Nell'API Amazon ECS, specifica il flag `requiresCompatibilities`.

Sistemi operativi e architetture

Quando si configura una definizione di 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 contenitori Linux utilizzano solo il kernel e la configurazione del kernel del sistema operativo host. Ad esempio, la configurazione del kernel include i controlli di sistema `sysctl`. Un'immagine di container Linux può essere creata a partire da un'immagine di base che contiene i file e i programmi di qualsiasi distribuzione Linux. Se l'architettura della CPU corrisponde, puoi eseguire i container da qualsiasi immagine di container Linux e su qualsiasi sistema operativo.

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

Quando si eseguono container Windows su AWS Fargate, è necessario disporre di un'architettura CPU X86_64.

Quando esegui container Linux su AWS Fargate, puoi utilizzare l'architettura CPU X86_64 o l'architettura ARM64 per le applicazioni basate su ARM. Per ulteriori informazioni, consulta the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”.

CPU e memoria del processo

Le definizioni di attività di Amazon ECS per AWS Fargate richiedono che la CPU e la memoria vengano specificate a livello di processo. Sebbene per i processi Fargate sia possibile specificare CPU e memoria anche a livello di container, si tratta di un'opzione facoltativa. Per la maggior parte dei casi d'uso è necessario specificare queste risorse solo a livello di processo. La tabella seguente illustra le combinazioni valide di CPU e memoria a livello di attività. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in MiB o GB. Ad esempio, è possibile specificare

un valore di memoria `3072` in MiB o `3 GB` in GB. È possibile specificare i valori della CPU nel file JSON come stringa in unità CPU o CPU virtuali (vCPU). Ad esempio, è possibile specificare un valore CPU come `1024` nelle unità CPU o `1 vCPU` nelle vCPU.

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) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
16384 (16vCPU) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

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'indirizzo IPv4 privato del processo viene utilizzato per il pull dell'immagine. Per ulteriori informazioni sugli endpoint di interfaccia Amazon ECR, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.

Di seguito è riportato un esempio della `networkConfiguration` sezione relativa a un servizio Fargate:

```
"networkConfiguration": {
   "awsvpcConfiguration": {
      "assignPublicIp": "ENABLED",
      "securityGroups": [ "sg-12345678" ],
      "subnets": [ "subnet-12345678" ]
   }
}
```

Limiti delle risorse dei processi

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 `1024` 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 un`nofile`limite che è stato raddoppiato:

```
"ulimits": [
    {
       "name": "nofile",
       "softLimit": 2048,
       "hardLimit": 8192
    }
]
```

Per ulteriori informazioni sugli altri limiti di risorse che possono essere modificati, consulta Limiti delle risorse.

Registrazione

Registrazione degli eventi

Amazon ECS registra le azioni eseguite. EventBridge Puoi utilizzare Amazon ECS Events per EventBridge ricevere notifiche quasi in tempo reale sullo stato corrente dei tuoi cluster, servizi e attività Amazon ECS. Inoltre, puoi automatizzare le operazioni per rispondere a questi eventi. Per ulteriori informazioni, consulta Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge.

Registrazione del ciclo di vita delle attività

Le attività eseguite su Fargate pubblicano i timestamp per monitorare l'attività attraverso gli stati del ciclo di vita. Puoi visualizzare i timestamp nei dettagli dell'attività in AWS Management Console e descrivendola negli SDK. AWS CLI 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 Le migliori pratiche 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 `splunk`driver di registro.

Archiviazione dei processi

Per i processi Amazon ECS ospitati su Fargate, sono supportati i seguenti tipi di archiviazione:

- I volumi Amazon EBS forniscono storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Per ulteriori informazioni, consulta Usa i volumi Amazon EBS con Amazon ECS.
- Volumi Amazon EFS per l'archiviazione persistente. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.
- Montaggi vincolati per l'archiviazione temporanea. Per ulteriori informazioni, consulta Usa i bind mount con Amazon ECS.

Caricamento lento delle immagini di container utilizzando Seekable OCI (SOCI)

Le attività di Amazon ECS su Fargate che utilizzano la versione della piattaforma Linux `1.4.0` possono utilizzare Seekable OCI (SOCI) per avviare le attività più velocemente. Con SOCI, i container trascorrono solo pochi secondi sul recupero dell'immagine prima di avviarsi, lasciando così il tempo necessario per la configurazione dell'ambiente e la creazione di istanze dell'applicazione

mentre l'immagine viene scaricata in background. Questo processo si chiama caricamento lento. Quando Fargate avvia un'attività Amazon ECS, rileva automaticamente se esiste un indice SOCI per un'immagine nell'operazione e avvia il container senza attendere che l'intera immagine venga scaricata.

Per i container che funzionano senza indici SOCI, le immagini di container vengono scaricate completamente prima dell'avvio di quest'ultimo. Questo comportamento avviene su tutte le altre versione della piattaforma di Fargate e sull'AMI ottimizzata per Amazon ECS sulle istanze Amazon EC2.

Indici Seekable OCI

Seekable OCI (SOCI) è una tecnologia open source sviluppata da Seekable OCI AWS che può avviare i container più velocemente caricando pigramente l'immagine del contenitore. SOCI funziona creando un indice (indice SOCI) dei file all'interno di un'immagine di container esistente. Questo indice aiuta ad avviare i container più velocemente, offrendo la possibilità di estrarre un singolo file da un'immagine di container prima di scaricare l'intera immagine. L'indice SOCI deve essere archiviato come artefatto nello stesso repository dell'immagine all'interno del registro dei container. Devi utilizzare solo indici SOCI provenienti da fonti attendibili, poiché l'indice è la fonte autorevole per il contenuto dell'immagine. Per ulteriori informazioni, consulta Introduzione a Seekable OCI per il caricamento lento delle immagini di container.

Considerazioni

Se desideri che Fargate utilizzi un indice SOCI per caricare lentamente le immagini di container in un'attività, considera le seguenti informazioni:

- Solo le attività eseguite sulla versione della piattaforma Linux `1.4.0` possono utilizzare gli indici SOCI. Le attività che eseguono i container Windows su Fargate non sono supportate.
- Sono supportate le attività eseguite sull'architettura della CPU X86_64 o ARM64. Le attività Linux con architettura ARM64 non supportano il provider di capacità Fargate Spot.
- Le immagini di container nella definizione delle attività devono avere indici SOCI nello stesso registro di container dell'immagine.
- Le immagini di container nella definizione delle attività devono essere archiviate in un registro di immagini compatibile. Di seguito sono elencati i registri compatibili:
  - Registri privati Amazon ECR.
- 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.

- 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 vuoi evitare che un'immagine di container venga caricata lentamente, elimina l'indice SOCI dal registro di container. Se l'immagine di container inclusa nell'operazione non soddisfa una delle considerazioni, tale immagine viene scaricata con il metodo predefinito.

Creazione di un indice Seekable OCI

Affinché un'immagine del contenitore venga caricata lentamente, è necessario un indice SOCI (un file di metadati) creato e archiviato nel repository di immagini del contenitore insieme all'immagine del contenitore. Per creare e inviare un indice SOCI puoi utilizzare lo strumento CLI soci-snapshotter open source. GitHub In alternativa, puoi implementare SOCI Index Builder. CloudFormation AWS Si tratta di una soluzione serverless che crea e invia automaticamente un indice SOCI quando un'immagine del contenitore viene inviata ad Amazon ECR. Per ulteriori informazioni sulla soluzione e sui passaggi di installazione, consulta CloudFormation AWS SOCI Index Builder su. GitHub CloudFormation AWS SOCI Index Builder è un modo per automatizzare l'avvio a usare SOCI, mentre lo strumento open source soci offre una maggiore flessibilità nella generazione di indici e la capacità di integrare la generazione di indici nelle pipeline di integrazione continua e distribuzione continua (CI/CD).

> **Note**
>
> Affinché l'indice SOCI possa essere creato per un'immagine, l'immagine deve esistere 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 se viene utilizzato SOCI. `soci`

Differenze nella definizione delle attività di Amazon ECS per le istanze EC2 che eseguono Windows

Le attività eseguite su istanze EC2 Windows non supportano tutti i parametri di definizione delle attività di Amazon ECS disponibili. Alcuni parametri non sono affatto supportati e altri si comportano in modo diverso.

I seguenti parametri di definizione delle attività non sono supportati per le definizioni delle attività Windows di Amazon EC2:

- `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 Windows EC2 richiedono una configurazione aggiuntiva, ma gran parte di questa configurazione è simile alla configurazione dei ruoli IAM per le attività su istanze di container Linux. Per ulteriori informazioni, consulta the section called “ Configurazione aggiuntiva dell'istanza Windows di Amazon EC2”.

Creazione di una definizione di attività Amazon ECS utilizzando la console

Puoi creare una definizione di attività utilizzando la console o modificando un file JSON.

Convalida JSON

L'editor JSON della console Amazon ECS verifica quanto segue nel file JSON:

- Il file è un file JSON valido.
- Il file non contiene chiavi estranee.
- Il file contiene il parametro. `familyName`
- C'è almeno una voce sotto`containerDefinitions`.

AWS CloudFormation pile

Il seguente comportamento si applica alle definizioni delle attività create nella nuova console Amazon ECS prima del 12 gennaio 2023.

Quando crei una definizione di attività, la console Amazon ECS crea automaticamente uno CloudFormation stack con un nome che inizia con. `ECS-Console-V2-TaskDefinition-` Se hai utilizzato AWS CLI o un AWS SDK per annullare la registrazione della definizione dell'attività, devi eliminare manualmente lo stack di definizione dell'attività. Per ulteriori informazioni, consulta Eliminazione di uno stack nella Guida per l'utente.AWS CloudFormation

Per le definizioni delle attività create dopo il 12 gennaio 2023 non viene creato automaticamente uno CloudFormation stack per esse.

Procedura

Amazon ECS console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Nel menu Crea nuova definizione di attività, scegli Crea nuova definizione di attività.
4. Per Task definition family (Famiglia della definizione di attività) specifica un nome univoco per la definizione di attività.
5. Per Tipo di avvio, scegli l'ambiente dell'applicazione. L'impostazione predefinita della console è AWS Fargate(ovvero senza server). Amazon ECS utilizza questo valore per eseguire la convalida per garantire che i parametri di definizione delle attività siano validi per il tipo di infrastruttura.
6. In Operating system/Architecture (Sistema operativo/Architettura), scegli il sistema operativo e l'architettura della CPU per il processo.

   Per eseguire l'attività su un'architettura ARM a 64 bit, scegli Linux/ARM64. Per ulteriori informazioni, consulta the section called “Piattaforma di runtime”.

   Per eseguire le attività AWS Fargate sui container Windows, scegli un sistema operativo Windows supportato. Per ulteriori informazioni, consulta the section called “Sistemi operativi e architetture”.
7. Per Task size (Dimensioni attività), specifica i valori di CPU e memoria da riservare per l'attività. Il valore della CPU è specificato in vCPU e la memoria è specificata in 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

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows
8192 (8 vCPU) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
16384 (16vCPU) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

Per i processi ospitati su Amazon EC2, i valori della CPU dei processi supportati sono compresi tra 128 unità CPU (0,125 vCPU) e 10240 unità CPU (10 vCPU). Per specificare il valore della memoria in GB, inserisci GB dopo il valore. Ad esempio, per impostare il valore di memoria su 3 GB, immettere 3 GB.

Note

I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows.

8. In Network mode (Modalità di rete), scegli la modalità di rete da utilizzare. L'impostazione di default è la modalità awsvpc. Per maggiori informazioni, consulta Networking attività Amazon ECS.

   Se scegli bridge, in Port mappings, per Host port, inserisci il numero di porta sull'istanza del container da riservare per il container.

9. (Facoltativo) Espandi la sezione Task roles per configurare i ruoli AWS Identity and Access Management (IAM) per l'attività:

   a. Per Task role (Ruolo attività), scegli il ruolo IAM da assegnare all'attività. Un ruolo IAM di task fornisce le autorizzazioni per i contenitori di un'attività per richiamare le operazioni AWS API.

   b. Per Ruolo di esecuzione attività, scegli il ruolo.

      Per informazioni sui casi in cui utilizzare il ruolo di esecuzione di attività, consulta the section called “Ruolo IAM di esecuzione di attività”. Se non ti serve il ruolo, scegli Nessuno.

10. Per ogni container da definire nella tua definizione di attività, completa la procedura seguente.

    a. In Name (Nome) immetti un nome per il container.

    b. Per Image URI (URI immagine), specifica l'immagine da utilizzare per avviare un container. Le immagini nel registro Amazon ECR Public Gallery possono essere specificate utilizzando solo il nome del registro Amazon ECR Public. Ad esempio, se specificato, `public.ecr.aws/ecs/amazon-ecs-agent:latest` viene utilizzato il contenitore Amazon Linux ospitato nella Amazon ECR Public Gallery. Per tutti gli altri repository, specifica il repository utilizzando i `repository-url/image:tag` formati o. `repository-url/image@digest`

    c. Se l'immagine si trova in un registro privato esterno ad Amazon ECR, in Registro privato, attiva Autenticazione del registro privato. Quindi, in ARN o nome di Gestione dei segreti, inserisci il nome della risorsa Amazon (ARN) del segreto.

    d. Per Essential Container, se nella definizione dell'attività sono definiti due o più contenitori, è possibile specificare se il contenitore deve essere considerato essenziale. Quando un contenitore è contrassegnato come Essenziale, se quel contenitore si ferma, l'attività viene interrotta. Ogni definizione di attività deve contenere almeno un container essenziale.

e. La mappatura delle porte consente ai container di accedere alle porte sull'host per inviare o ricevere traffico. In Port mappings (Mappature di porte), esegui una delle operazioni seguenti:

   - Quando utilizzi la modalità di rete awsvpc, in Container port (Porta del container) e Protocol (Protocollo), specifica la mappatura delle porte da utilizzare per il container.
   - Quando utilizzi la modalità di rete bridge, in Container port (Porta del container) e Protocol (Protocollo), specifica la mappatura delle porte da utilizzare per il container.

   Scegli Add more port mappings (Aggiungi altre mappature porta) per specificare mappature aggiuntive delle porte del container.

f. Per consentire al container l'accesso in sola lettura al file system root, per File system root di sola lettura, seleziona Sola lettura.

g. (Facoltativo) Per definire i limiti di CPU, GPU e memoria a livello di contenitore diversi dai valori a livello di attività, in Limiti di allocazione delle risorse, procedi come segue:

   - Per CPU, inserisci il numero di unità CPU che l'agente container Amazon ECS riserva per il contenitore.
   - Per GPU, inserisci il numero di unità di GPU per l'istanza di container.

     Un'istanza Amazon EC2 con supporto GPU dispone di 1 unità di GPU per ogni GPU. Per ulteriori informazioni, consulta the section called “Definizioni delle attività per i carichi di lavoro GPU”.
   - Per Limite rigido di memoria, inserisci la quantità di memoria, in GB, da presentare al contenitore. Se il container tenta di superare il limite rigido, si arresta.
   - Il daemon Docker 20.10.0 o successivo riserva un minimo di 6 mebibyte (MiB) di memoria per un contenitore, quindi non specificare meno di 6 MiB di memoria per i tuoi contenitori.

     Il daemon Docker 19.03.13-ce o precedente riserva un minimo di 4 MiB di memoria per un contenitore, quindi non specificare meno di 4 MiB di memoria per i contenitori.
   - Per Limite di memoria flessibile, inserisci il limite flessibile (in GB) di memoria da prenotare per il container.

     Quando la memoria di sistema è in conflitto, Docker tenta di 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 nei contenitori Windows.

h. (Facoltativo) Espandi la sezione Variabili di ambiente per specificare le variabili di ambiente da inserire nel container. Puoi specificare le variabili di ambiente singolarmente utilizzando coppie chiave-valore o in blocco specificando un file di variabile di ambiente ospitato in un bucket Amazon S3. Per informazioni su come formattare un file di variabili di ambiente, consulta. Passa una singola variabile di ambiente a un contenitore Amazon ECS

Quando specificate una variabile di ambiente per l'archiviazione segreta, in Key, immettete il nome segreto. Quindi ValueFrom, inserisci l'ARN completo del segreto di Systems Manager Parameter Store o del segreto di Secrets Manager

i. (Facoltativo) Seleziona l'opzione Use log collection (Usa raccolta di log) per specificare una configurazione di log. Per ogni driver di log disponibile, sono disponibili opzioni del driver di log da specificare. L'opzione predefinita invia i log dei contenitori ad Amazon CloudWatch Logs. Le altre opzioni del driver di registro vengono configurate utilizzando. AWS FireLens Per ulteriori informazioni, 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 log in Splunk: configura l'attività per inviare i log dei container al Splunk driver che invia i log a un servizio remoto. È necessario immettere l'URL del servizio WebSplunk. Il Splunk token è specificato come opzione segreta perché può essere trattato come dati sensibili.
- Esportazione dei log su Amazon Data Firehose: configura l'attività per inviare i log dei container a Firehose. Vengono fornite le opzioni predefinite del driver di registro, che inviano il log a un flusso di distribuzione Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.

- Esportazione dei log su Amazon Kinesis Data Streams: configura l'attività per inviare i log dei container a Kinesis Data Streams. Vengono fornite le opzioni predefinite del driver di registro, che inviano i log a un flusso Kinesis Data Streams. Per specificare un nome del flusso diverso, modifica i valori dell'opzione del driver.
- Esporta i log su Amazon OpenSearch Service: configura l'attività per inviare i log dei container a un dominio di OpenSearch servizio. Devono essere fornite le opzioni del driver di log.
- Esportazione dei log su Amazon S3: configura l'attività per inviare i log dei container a un bucket Amazon S3. Vengono fornite le opzioni di driver di registro predefinite, ma è necessario specificare un nome di bucket Amazon S3 valido.

j. (Facoltativo) Configura i parametri del container aggiuntivi.

Per configurare questa opzione	Esegui questa operazione	
Healthcheck Questi sono i comandi che determinano se un contenitore è integro. Per ulteriori informazioni, consulta Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container.	HealthCheckEspandi e configura i seguenti elementi: • Per Command (Comando), inserisci un elenco di comandi separati da virgole. È possibile avviare i comandi con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell (interprete di comandi) predefinita del container. Se non è specificato nessuno dei due, viene utilizzato `CMD`. • In Interval (Intervallo), inserisci il numero di secondi tra ogni controllo dell'integrità. I valori validi sono compresi tra 5 e 30. • For Timeout, specifica il periodo di tempo in secondi per cui	

Per configurare questa opzione	Esegui questa operazione	
	attendere che un controllo dell'integrità venga superato prima che sia consideto un errore. I valori validi sono compresi tra 2 e 60. • Per Start period (Periodo di inizio), inserisci l'intervallo di tempo (in secondi) da attendere per l'avvio di un container prima dell'esecuzione dei comandi di controllo dell'integrità. I valori validi sono compresi tra 0 e 300. • In Retries (Tentativi), inserisci il numero di volte in cui riprovare i comandi di controllo dell'integrità in caso di errore. I valori validi sono compresi tra 1 e 10.	

Per configurare questa opzione	Esegui questa operazione	
Timeout container Queste opzioni determina no quando avviare e arrestare un container.	Espandi Timeout container, quindi configura gli elementi seguenti: • Per configurare il tempo di attesa prima di rinunciare alla risoluzio ne delle dipendenze per un contenitore, per Start timeout, inserisci il numero di secondi. • Per configurare il tempo di attesa prima che il contenitore si fermi se non esce normalmen te da solo, per Stop timeout, inserisci il numero di secondi.	

Per configurare questa opzione	Esegui questa operazione
Impostazioni di rete del container Queste opzioni determina no 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 **`^[a-zA-Z0-9-.]{0,253}[a-zA-Z0-9]$`** . •

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 `/etc/hosts` nel container, scegli Aggiungi altro host, quindi in Nome host e indirizzo IP, inserisci i relativi valori.	

Per configurare questa opzione	Esegui questa operazione	
Dockerconfigurazione Questi sovrascrivono i valori in. Dockerfile	DockerEspandi la configurazione, quindi configura i seguenti elementi: • Per Comando, inserisci un comando eseguibile per un container. Questo parametro è `Cmd` mappato alla sezione Crea un contenitore dell'API Docker remota e all'`COMMAND`opzione a`docker run` . Questo parametro sostituisce l'`CMD`istruzione in a. Dockerfile • Per Punto di ingresso, inserisci l'DockerE NTRYPOINT che viene passato al contenitore. Questo parametro è `Entrypoint` mappato alla sezione Crea un contenitore dell'API Docker remota e all'`--entrypoint` opzione a. `docker run`	

Per configurare questa opzione	Esegui questa operazione	
	Questo parametro sostituisce l'`ENTRYPOINT` istruzione in a. Dockerfile • Per Directory di lavoro, inserisci la directory in cui il container eseguirà ogni istruzione di punto di ingresso e di comando fornita. Questo parametro è `WorkingDir` mappato alla sezione Crea un contenitore dell'API Docker remota e all'`--workdir` opzione a. `docker run` Questo parametro sostituisce l'`WORKDIR`istruzione in a. Dockerfile	

Per configurare questa opzione	Esegui questa operazione	
Ulimits Questi valori sovrascrivono l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è mappato a `Ulimits` nella sezione Create a container di Docker Remote API e l'opzione `--ulimit` a docker run.	Espandi Resource limits (ulimits), quindi scegli Aggiungi. ulimit In Nome limite, scegli il limite. Quindi, per Limite flessibile e Limite rigido, inserisci i relativi valori. Per aggiungerne altriulimits, scegli Aggiungi ulimit.	
Dockeretichette Questa opzione aggiunge metadati al container. Questo parametro è mappato a `Labels` nella sezione Create a container di Docker Remote API e l'opzione `--label` a docker run.	Espandi Dockerle etichette, scegli Aggiungi coppia chiave-valore, quindi inserisci la chiave e il valore. Per aggiungere altre Docker etichette, scegli Aggiungi coppia chiave-valore.	

Per configurare questa opzione	Esegui questa operazione	
Ordine di avvio del container Questa opzione definisce le dipendenze per l'avvio e l'arresto del container. Un container può contenere più dipendenze.	Espandi Ordine delle dipendenze di avvio, quindi configura gli elementi seguenti: a. Scegli Aggiungi dipendenza del container. b. Per Container, seleziona il container. c. Per Condizione, scegli la condizione di dipendenza di avvio. Per aggiungere un'ulteri ore dipendenza, scegli Aggiungi dipendenza del container.	

k. (Facoltativo) Scegli Add more containers (Aggiungi altri container) per aggiungere altri container alla definizione del processo.

11. (Facoltativo) La sezione Archiviazione viene utilizzata per espandere la quantità di spazio di archiviazione temporaneo per le attività ospitate su Fargate. È inoltre possibile utilizzare questa sezione per aggiungere una configurazione del volume di dati per l'attività.

    - Per espandere lo spazio di archiviazione temporaneo disponibile oltre il valore predefinito di 20 gibibytes (GiB) per le attività Fargate, in Amount (Quantità), immetti un valore massimo di 200 GiB.

12. (Facoltativo) Per aggiungere una configurazione del volume di dati per la definizione dell'attività, scegliete Aggiungi volume e quindi seguite questi passaggi.

a. Per Volume name (Nome volume), immetti un nome per il volume di dati. Il nome del volume di dati viene utilizzato quando si crea un punto di montaggio del container.

b. Per Configurazione del volume, seleziona se desideri configurare il volume durante la creazione della definizione dell'attività o durante la distribuzione.

> **Note**
>
> I volumi che possono essere configurati durante la creazione di una definizione di attività includono Bind mountDocker, Amazon EFS e Amazon FSx for Windows File Server. I volumi che possono essere configurati al momento della distribuzione durante l'esecuzione di un'attività o durante la creazione o l'aggiornamento di un servizio includono Amazon EBS.

c. Per Tipo di volume, seleziona un tipo di volume compatibile con il tipo di configurazione selezionato, quindi configura il tipo di volume.

Tipo di volume	Fasi	
Bind mount	a. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente) , scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. b. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	

Tipo di volume	Fasi	
EFS	a. Per File system ID (ID del file system), scegli l'ID del file system Amazon EFS. b. (Facoltativo) Per Root directory (Directory root), specifica la directory all'interno del file system Amazon EFS da montare come directory root all'interno dell'host. Se questo parametro viene omesso, viene utilizzat a la radice del volume Amazon EFS. Se prevedi di utilizzar 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 crittogra fare 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 per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non si specifica una porta di crittografia di transito, verrà utilizzata la strategia di selezione della porta utilizzata dall'helper per il montaggio di Amazon EFS. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System. • Per utilizzare il ruolo IAM dell'attività	

Tipo di volume	Fasi	
	Amazon ECS riportato in una definizione di attività durante il montaggio del file system Amazon EFS, seleziona IAM authorization (Autorizzazione IAM). e. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume.	

Tipo di volume	Fasi	
	f. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
Docker	a. Per Driver, inserisci la configurazione del Docker volume. I contenitori Windows supportano solo l'uso del driver locale. Per utilizzare i montaggi vincolati, specifica un host. b. In Scope (Ambito), scegli il ciclo di vita del volume. • Per fare in modo che il ciclo di vita duri dall'inizio e all'interruzione dell'attività, scegli Task (Attività). • Per mantenere il volume dopo l'interruzione dell'attività, scegli Shared (Condiviso). c. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. •	

Tipo di volume	Fasi	
	Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. d. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
FSx per File Server Windows	a. Per ID del file system, seleziona l'ID del file system di FSx for Windows File Server. b. Per la directory principal e, immettere la directory , immettere la directory all'interno del file system FSx for 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 AWS Secrets Manager, inserisci l'Amazon Resource Name (ARN) di un segreto di Secrets Manager. • Per utilizzarlo AWS Systems Manager, inserisci l'Amazon Resource Name (ARN) di un parametro Systems Manager. 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 EC2 ospitato autonomamente. e. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente), scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso	

Tipo di volume	Fasi	
	in sola lettura al volume. f. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

Tipo di volume	Fasi	
Amazon EBS	a. Scegli Add mount point (Aggiungi punto di montaggio), quindi configura quanto segue: • Per Container, scegli il container per il punto di montaggio. • Per Source volume (Volume sorgente) , scegli il volume di dati da montare sul container. • Per Container path (Percorso container) specifica il percorso nel container per il montaggio del volume. • Per Sola lettura, seleziona se il container ha accesso in sola lettura al volume. b. Per aggiungere altri punti di montaggio, seleziona Add mount point (Aggiungi punto di montaggio).	

13. 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.

14. (Facoltativo) Per configurare le impostazioni di tracciamento e raccolta delle metriche dell'applicazione utilizzando l' AWS Distro for OpenTelemetryintegrazione, 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 su Amazon CloudWatch, la definizione dell'attività richiede un ruolo IAM dell'attività con le autorizzazioni richieste. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon CloudWatch.

    b. Quando selezioni Amazon Managed Service for Prometheus (Prometheus libraries instrumentation) (Amazon Managed Service for Prometheus [strumentazione delle librerie Prometheus]), la CPU, la memoria, la rete e i parametri di archiviazione a livello di processo e i parametri delle applicazioni personalizzati vengono instradati ad Amazon Managed Service for Prometheus. Per l'endpoint di scrittura remota Workspace, inserisci l'URL dell'endpoint di scrittura remota per il tuo spazio di lavoro. Prometheus Per Scraping target, inserisci l'host e la porta che il AWS Distro for OpenTelemetry raccoglitore può utilizzare per raccogliere i dati delle metriche. Per ulteriori informazioni, consulta Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus.

> **Important**
>
> Quando si esportano i parametri delle applicazioni in Amazon Managed Service for Prometheus, la definizione di attività richiede un ruolo IAM del processo con le autorizzazioni richieste. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon Managed Service for Prometheus.

c. Quando selezioni Amazon Managed Service for Prometheus OpenTelemetry (strumentazione), i parametri di CPU, memoria, rete e storage a livello di attività e i parametri delle applicazioni personalizzate vengono indirizzati ad Amazon Managed Service for Prometheus. Per l'endpoint di scrittura remota Workspace, inserisci l'URL dell'endpoint di scrittura remota per il tuo spazio 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.

15. (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).

16. Scegli Crea per registrare la definizione di attività.

Amazon ECS console JSON editor

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.

2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Nel menu Crea nuova definizione di attività, scegli Crea nuova definizione di attività con JSON.
4. Nella casella dell'editor JSON, modifica il tuo file JSON,

   Il JSON deve superare i controlli di convalida specificati in the section called “Convalida JSON”.
5. Scegli Crea.

Aggiornamento di una definizione di attività Amazon ECS tramite la console

Una revisione della definizione di attività è una copia della definizione di attività corrente con i nuovi valori dei parametri che sostituiscono quelli esistenti. Tutti i parametri che non si modificano sono nella nuova revisione.

Per aggiornare una definizione di attività, devi creare una revisione della definizione di attività. Se la definizione di attività viene utilizzata all'interno di un servizio, devi aggiornare quest'ultimo per poter utilizzare la definizione di attività aggiornata.

Quando si crea una revisione, è possibile modificare le seguenti proprietà del container e le proprietà dell'ambiente.

- URI dell'immagine del container
- Mappature di porte
- Variabili di ambiente
- Dimensioni processo
- Dimensioni del container
- Ruolo del processo
- Ruolo di esecuzione di attività
- Volumi e punti di montaggio dei container
- Registro privato

Convalida JSON

L'editor JSON della console Amazon ECS verifica quanto segue nel file JSON:

- Il file è un file JSON valido
- Il file non contiene chiavi esterne
- Il file contiene il parametro `familyName`
- È presente almeno una voce in `containerDefinitions`

Procedura

Amazon ECS console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Dalla barra di navigazione, scegli la Regione in cui si trova la definizione di attività.
3. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
4. Scegli la definizione di attività.
5. Seleziona la revisione della definizione di attività, quindi scegli Crea nuova revisione, Crea nuova revisione.
6. Nella pagina Create new task definition revision (Crea nuova revisione della definizione di attività), apporta le modifiche desiderate. Ad esempio, per modificare le definizioni del container esistenti (ad esempio l'immagine del container, i limiti di memoria o le mappature delle porte), seleziona il container e apporta le modifiche desiderate.
7. Verifica le informazioni e scegli Aggiorna.
8. Se la tua definizione di attività viene utilizzata all'interno di un servizio, devi aggiornare quest'ultimo con la definizione di attività aggiornata. Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console.

Amazon ECS console JSON editor

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
3. Scegli Create new revision (Crea nuova revisione), Create new revision with JSON (Crea nuova revisione con JSON).

4. Nella casella dell'editor JSON, modifica il tuo file JSON,

   Il JSON deve superare i controlli di convalida specificati in the section called “Convalida JSON”.

5. Scegli Crea.

Annullamento della registrazione di una revisione della definizione di attività di Amazon ECS tramite la console

Quando non è più necessaria una revisione specifica della definizione di attività in Amazon ECS, è possibile annullare la registrazione della revisione della definizione dell'attività in modo che non venga più visualizzata nelle chiamate `ListTaskDefinition` API o nella console quando si desidera eseguire un'attività o aggiornare un servizio.

Quando revochi la registrazione della revisione di una definizione di attività, questa viene immediatamente contrassegnata come `INACTIVE`. Processi e servizi esistenti che fanno riferimento alla revisione di una definizione di attività `INACTIVE` continuano a essere eseguiti senza interruzioni. I servizi esistenti che fanno riferimento a una revisione di definizione di attività `INACTIVE` possono ancora essere ridotti o dimensionati orizzontalmente modificando il conteggio di servizi desiderato.

Non è possibile utilizzare una revisione di definizione di attività `INACTIVE` per eseguire nuovi processi o creare nuovi servizi. Non potrai aggiornare un servizio esistente per fare riferimento a una revisione della definizione di attività `INACTIVE` (anche se sono necessari circa 10 minuti dopo la revoca della registrazione affinché queste restrizioni vengano applicate).

> **Note**
>
> Quando annulli la registrazione di tutte le revisioni, la famiglia di definizioni delle attività viene spostata nell'elenco `INACTIVE`. L'aggiunta di una nuova revisione di una definizione dell'attività `INACTIVE` riporta la famiglia di definizioni delle attività nell'elenco `ACTIVE`. In questo momento, le revisioni delle definizioni di processi `INACTIVE` rimangono individuabili nel tuo account a tempo indeterminato. Tuttavia, questo comportamento è soggetto a modifiche in futuro. Pertanto, non fare affidamento a revisioni delle definizioni di processo `INACTIVE` che vanno oltre il ciclo di vita dei processi e dei servizi associati.

AWS CloudFormation pile

Il seguente comportamento si applica alle definizioni delle attività create nella nuova console Amazon ECS prima del 12 gennaio 2023.

Quando crei una definizione di attività, la console Amazon ECS crea automaticamente uno CloudFormation stack con un nome che inizia con. `ECS-Console-V2-TaskDefinition-` Se hai utilizzato AWS CLI o un AWS SDK per annullare la registrazione della definizione dell'attività, devi eliminare manualmente lo stack di definizione dell'attività. Per ulteriori informazioni, consulta Eliminazione di uno stack nella Guida per l'utente.AWS CloudFormation

Per le definizioni delle attività create dopo il 12 gennaio 2023 non viene creato automaticamente uno CloudFormation stack per esse.

Procedura

Per annullare la registrazione di una nuova definizione di attività (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Dalla barra di navigazione, scegli la regione in cui si trova la definizione di attività.
3. Nel pannello di navigazione, scegli Task Definitions (Definizioni di processo).
4. Nella pagina Task definitions (Definizioni di processi), scegli la famiglia di definizioni di attività contenente una o più revisioni per cui desideri annullare la registrazione.
5. Nella pagina Nome definizione di attività, seleziona le revisioni da eliminare, quindi seleziona Operazioni, Annulla registrazione.
6. Verifica le informazioni riportate nella finestra Deregister (Annulla registrazione), quindi scegli Deregister (Annulla registrazione) per terminare.

Eliminazione di una revisione della definizione di attività di Amazon ECS tramite la console

Quando non è più necessaria una revisione specifica della definizione dell'attività in Amazon ECS, puoi eliminare la revisione della definizione dell'attività.

Quando elimini una revisione di definizione di attività, questa passa immediatamente da `INACTIVE` a `DELETE_IN_PROGRESS`. L'esecuzione delle attività e dei servizi esistenti che fanno riferimento a una revisione di definizione di attività `DELETE_IN_PROGRESS` prosegue senza interruzioni.

Non è possibile utilizzare una revisione di definizione dell'attività `DELETE_IN_PROGRESS` per eseguire nuove attività o creare nuovi servizi. Inoltre, non è possibile aggiornare un servizio esistente per fare riferimento a una revisione di definizione di attività `DELETE_IN_PROGRESS`.

Quando elimini tutte le revisioni delle definizioni di attività nello stato `INACTIVE`, il nome della definizione di attività non viene visualizzato nella console e non viene restituito nell'API. Se `DELETE_IN_PROGRESS` lo stato di revisione della definizione dell'attività è in corso, il nome della definizione dell'attività viene visualizzato nella console e restituito nell'API. Il nome della definizione di attività viene mantenuto da Amazon ECS e la revisione viene incrementata durante la prossima creazione di una definizione di attività con tale nome.

Risorse Amazon ECS in grado di bloccare un'eliminazione

Una richiesta di eliminazione della definizione di attività non viene completata quando sono presenti risorse Amazon ECS che dipendono dalla revisione della definizione dell'attività. Le seguenti risorse potrebbero impedire l'eliminazione di una definizione di attività:

- Attività di Amazon ECS: la definizione di attività è necessaria affinché l'attività rimanga integra.
- Implementazioni e set di attività 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, puoi 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 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
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 della definizione dell'attività, seleziona le revisioni da eliminare, quindi scegli Azioni, Elimina.

   Se l'opzione Elimina non è disponibile, è necessario annullare la registrazione della definizione dell'attività.
6. Verifica le informazioni nella casella di conferma dell'eliminazione, quindi scegli Elimina per terminare.

Casi d'uso per la definizione delle attività di Amazon ECS

Scopri di più su come scrivere definizioni di attività per vari AWS servizi e funzionalità.

A seconda del carico di lavoro, è necessario impostare determinati parametri di definizione delle attività. Inoltre, per il tipo di avvio di EC2, devi scegliere istanze specifiche progettate per il carico di lavoro.

Argomenti

- Definizioni di attività Amazon ECS per carichi di lavoro GPU
- Definizioni delle attività di Amazon ECS per carichi di lavoro di trancodifica video
- Definizioni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron
- Definizioni delle attività di Amazon ECS per le istanze di deep learning
- Definizioni delle attività di Amazon ECS per carichi di lavoro ARM a 64 bit
- Invia i log di Amazon ECS a CloudWatch

- Inviare i log di Amazon ECS a un servizio o AWSAWS Partner
- Utilizzo di immagini non AWS containerizzate in Amazon ECS
- Passa una singola variabile di ambiente a un contenitore Amazon ECS
- Passa le variabili di ambiente a un contenitore Amazon ECS
- Trasferisci dati sensibili a un contenitore Amazon ECS

Definizioni di attività Amazon ECS per carichi di lavoro GPU

Amazon ECS supporta carichi di lavoro che utilizzano le GPU, quando crei cluster con istanze di container che supportano le GPU. Le istanze di container basate su GPU di Amazon EC2 che utilizzano i tipi di istanza p2, p3, p5, g3, g4 e g5 forniscono accesso alle GPU NVIDIA. Per ulteriori informazioni, consulta Linux Accelerated Computing Instances nella Amazon EC2 User Guide.

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 AMI Linux ottimizzate per Amazon ECS.

Puoi designare un certo numero di GPU nella definizione di attività per considerazioni di posizionamento del processo a livello di container. Amazon ECS esegue la programmazione verso le istanze di container disponibili che supportano le GPU e vincolano le GPU fisiche ai container corretti per garantire prestazioni ottimali.

Sono supportati i seguenti tipi di istanza Amazon EC2 basati su GPU. Per ulteriori informazioni, consulta Istanze Amazon EC2 P2, Istanze Amazon EC2 P3, Istanze Amazon EC2 P4d, IstanzeAmazon EC2 P5, Istanze Amazon EC2 G3, Istanze Amazon EC2 G4, Istanze Amazon EC2 G5e Istanze Amazon EC2 G6.

Tipo di istanza	GPU	Memoria GPU (GiB)	vCPU	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	GPU	Memoria GPU (GiB)	vCPU	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	GPU	Memoria GPU (GiB)	vCPU	Memoria (GiB)
g6.xlarge	1	24	4	16
g6,2 x grande	1	24	8	32
g 6,4 x grande	1	24	16	64
g 6,8 x grande	1	24	32	128
g6,16.x grande	1	24	64	256
g6.12 x grande	4	96	48	192
g6,24 x grande	4	96	48	192
g6,48 x grande	8	192	192	768
g6. metallo	8	192	192	768
gr 6,4 x grande	1	24	16	128
gr 6,8 x grande	1	24	32	256

Puoi recuperare l'ID Amazon Machine Image (AMI) per le AMI ottimizzate per Amazon ECS interrogando l'API Parameter Store. AWS Systems Manager Con questo parametro, non è necessario eseguire una ricerca manuale degli ID dell'AMI ottimizzata per Amazon ECS. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere GetParameter. Il principale IAM che utilizzi deve disporre dell'autorizzazione IAM `ssm:GetParameter` per recuperare i metadati dell'AMI ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended --region us-east-1
```

Considerazioni

> **Note**
>
> Il supporto per il tipo di famiglia di istanze g2 è obsoleto.

La famiglia di istanze p2 è supportata solo nelle versioni precedenti a `20230912` dell'AMI Amazon ECS ottimizzata per GPU. Per continuare a utilizzare le istanze p2, consulta Operazioni da eseguire per utilizzare un'istanza P2.
Gli aggiornamenti locali dei driver NVIDIA/CUDA su entrambi i tipi di famiglie di istanze causano potenziali errori nel carico di lavoro della GPU.

È consigliabile considerare le informazioni seguenti prima di iniziare a utilizzare le 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 di container.

  L'esempio seguente avvia un'attività su un'istanza di container `g4dn.xlarge` nel cluster predefinito.

  ```
  aws ecs run-task --cluster default --task-definition ecs-gpu-task-def \
       --placement-constraints type=memberOf,expression="attribute:ecs.instance-type ==
    g4dn.xlarge" --region us-east-2
  ```

- Per ogni container che ha un requisito di risorsa GPU specificato nella definizione di container, Amazon ECS imposta il runtime del container in modo che sia il runtime del container NVIDIA.
- Il runtime del container NVIDIA richiede per poter funzionare l'impostazione di alcune variabili di ambiente nel container. Per un elenco di queste variabili di ambiente, vedere Configurazioni specializzate con Docker. Amazon ECS imposta il valore della variabile di ambiente `NVIDIA_VISIBLE_DEVICES` come un elenco di ID dispositivo GPU che Amazon ECS assegna al

container. 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 AMI Linux ottimizzate per Amazon ECS. Non è supportata nel flusso di lavoro Crea cluster nella console Amazon ECS. Per utilizzare questi tipi di istanze, devi utilizzare la console o l'API di Amazon EC2 e registrare manualmente le istanze nel cluster. AWS CLI
- Il tipo di istanza p4d.24xlarge funziona solo con CUDA 11 o versioni successive.
- L'AMI ottimizzata per le GPU di Amazon ECS ha abilitato IPv6, che causa problemi quando si utilizza `yum`. Questo può essere risolto configurando `yum` in modo da utilizzare IPv4 con il seguente comando.

  ```
  echo "ip_resolve=4" >> /etc/yum.conf
  ```

- Quando si crea un'immagine di container che non utilizza le immagini di base NVIDIA/CUDA, è necessario impostare la variabile di runtime del container `NVIDIA_DRIVER_CAPABILITIES` su uno dei seguenti valori:
  - `utility,compute`
  - `all`

  Per informazioni su come impostare la variabile, consulta Controllo del runtime del container NVIDIA sul sito Web di NVIDIA.
- Le GPU non sono supportate nei container Windows.

Avvia un'istanza di contenitore GPU per Amazon ECS

Per utilizzare un'istanza GPU su Amazon ECS, devi creare un modello di avvio, un file di dati utente e avviare l'istanza.

È quindi possibile eseguire un'attività che utilizza una definizione di attività configurata per la GPU.

Utilizzo di un modello di avvio

È possibile creare un modello di lancio.

- Crea un modello di lancio che utilizzi l'ID AMI GPU ottimizzato per Amazon ECS per l'AMI. Per informazioni su come creare un modello di lancio, consulta Creare un nuovo modello di lancio utilizzando i parametri definiti nella Amazon EC2 User Guide.

  Usa l'ID AMI del passaggio precedente per l'immagine Amazon Machine. Per informazioni su come specificare l'ID AMI con il parametro Systems Manager, consulta Specificare un parametro Systems Manager in un modello di avvio nella Amazon EC2 User Guide.

  Aggiungi quanto segue ai dati utente nel modello di lancio. Sostituire *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. Sostituire *cluster-name* con il nome del cluster.

   ```
   #!/bin/bash
   echo ECS_CLUSTER=cluster-name >> /etc/ecs/ecs.config;
   echo ECS_ENABLE_GPU_SUPPORT=true >> /etc/ecs/ecs.config
   ```

2. Esegui il seguente comando per ottenere l'ID AMI della GPU. Ti servirà per la fase successiva.

   ```
   aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
   recommended --region us-east-1
   ```

3. Esegui il comando seguente per avviare l'istanza GPU. Ricordati di sostituire i seguenti parametri:

   - Sostituisci la *sottorete* 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 da utilizzare.
   - Sostituisci *regione* con il codice della regione.

```
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 *my-cluster* con il nome del cluster.
   - Sostituisci *regione* con il codice della Regione.

```
aws ecs list-container-instances --cluster cluster-name --region region
```

Specificazione delle GPU in una definizione di attività Amazon ECS

Per utilizzare le GPU su un'istanza di container e il runtime della GPU del Docker, assicurati di indicare il numero di GPU richiesti dal container nella definizione di attività. Non appena i container che supportano le GPU vengono posizionati, l'agente del container di Amazon ECS vincola il numero desiderato di GPU fisiche al container appropriato. Il numero di GPU riservate per tutti i container in un'attività non deve superare il numero di GPU disponibili nell'istanza di container in cui viene avviata l'attività. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

> **Important**
>
> Se i requisiti della GPU non sono specificati nella definizione di attività, il processo utilizzerà il runtime del Docker di default.

Di seguito viene visualizzato il formato JSON per i requisiti GPU in una definizione di attività:

```
{
```

```
  "containerDefinitions": [
     {
        ...
        "resourceRequirements" : [
            {
               "type" : "GPU",
               "value" : "2"
            }
        ],
     },
...
}
```

L'esempio seguente mostra la sintassi per un container Docker che specifica un requisito GPU. Questo container utilizza 2 GPU, esegue l'utility `nvidia-smi` e quindi viene terminato.

```
{
  "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"
}
```

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 Lavora con i dati utente delle istanze 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'AMI EC2 privata compatibile con P2.

   Sostituisci la regione con la regione dell'istanza.

   ```
   REGION=region make al2keplergpu
   ```

4. Utilizza l'AMI con i seguenti dati utente dell'istanza per connetterti al cluster Amazon ECS.

   Sostituire cluster-name con il nome del cluster.

   ```
   #!/bin/bash
   echo "ECS_CLUSTER=cluster-name" >> /etc/ecs/ecs.config
   ```

Definizioni delle attività di Amazon ECS per carichi di lavoro di trancodifica video

Per utilizzare carichi di lavoro di transcodifica video su Amazon ECS, registra le istanze VT1 di Amazon EC2. Dopo aver registrato queste istanze, puoi eseguire carichi di lavoro di transcodifica video live e pre-renderizzati come attività su Amazon ECS. Le istanze VT1 di Amazon EC2 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 a implementare VT1 su Amazon ECS, tieni presente quanto segue:

- I cluster possono contenere una combinazione di istanze VT1 e non 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. Puoi eseguire tante attività di transcodifica quante sono le schede 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 si assicura che le risorse vengano utilizzate in modo efficace e che le attività per i carichi di lavoro di transcodifica video si trovino nelle istanze VT1. Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

  Nell'esempio seguente viene eseguita un'attività su una istanza `vt1.3xlarge` sul cluster `default`.

```
aws ecs run-task \
     --cluster default \
     --task-definition vt1-3xlarge-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'AMI VT1

Sono disponibili due opzioni per l'esecuzione di un'AMI su Amazon EC2 per 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.

- Xilinx offre AMI su. 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 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	vCPU	RAM (GiB)	Schede acceleratore U30	Dispositivi SoC XCU30 indirizzabili	Percorsi dispositivi
vt1.3xlarge	12	24	1	2	`/dev/dri/renderD128 ,/dev/dri/renderD129`

Tipo di istanza	vCPU	RAM (GiB)	Schede acceleratore U30	Dispositivi SoC XCU30 indirizzabili	Percorsi dispositivi
vt1.6xlarge	24	48	2	4	`/dev/dri/renderD128 ,/dev/dri/renderD129 ,/dev/dri/renderD130 ,/dev/dri/renderD131`

Tipo di istanza	vCPU	RAM (GiB)	Schede acceleratore U30	Dispositivi SoC XCU30 indirizzabili	Percorsi dispositivi
vt1.24xlarge	96	182	8	16	`/dev/dri/renderD128 ,/dev/dri/renderD129 ,/dev/dri/renderD130 ,/dev/dri/renderD131 ,/dev/dri/renderD132 ,/dev/dri/renderD133 ,/dev/dri/renderD134 ,/dev/dri/renderD135 ,/dev/dri/renderD136 ,/dev/dri/renderD137 ,/dev/dri/renderD13`

Tipo di istanza	vCPU	RAM (GiB)	Schede acceleratore U30	Dispositivi SoC XCU30 indirizzabili	Percorsi dispositivi
					`8 ,/dev/dri/renderD139 ,/dev/dri/renderD140 ,/dev/dri/renderD141 ,/dev/dri/renderD142 ,/dev/dri/renderD143`

> **Important**
>
> Se la definizione di attività elenca dispositivi di cui l'istanza EC2 non dispone, l'attività non viene eseguita. Quando l'attività ha esito negativo, viene visualizzato il seguente messaggio di errore in `stoppedReason`: `CannotStartContainerError: Error response from daemon: error gathering device information while adding custom device "/dev/dri/renderD130": no such file or directory.`

Specificare la transcodifica video in una definizione di attività Amazon ECS

Nell'esempio seguente viene fornita la sintassi utilizzata per una definizione di attività di un container 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-xffmpeg-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": "xilinix-xffmpeg"
        }
    ],
    "volumes": [
        {
            "name": "video_file",
            "host": {"sourcePath": "/home/ec2-user"}
        }
    ]
}
```

2. Registra la definizione dell'attività.

```
aws ecs register-task-definition --family vt1-3xlarge-xffmpeg-processor --cli-
input-json file://vt1-3xlarge-xffmpeg-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-xffmpeg-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": "xilinix-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": "xilinix-xffmpeg"
        }
    ],
    "volumes": [
        {
            "name": "video_file",
            "host": {"sourcePath": "/home/ec2-user"}
        }
    ]
}
```

2. Registra la definizione dell'attività.

```
aws ecs register-task-definition --family vt1-24xlarge-xffmpeg-processor --cli-
input-json file://vt1-24xlarge-xffmpeg-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 le istanze Amazon EC2 Trn1, Amazon EC2 Inf1 e Amazon EC2 Inf2 sui cluster per carichi di lavoro di machine learning.

Le istanze Trn1 di Amazon EC2 si basano su chip AWS 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 Inf1 e Inf2 di Amazon EC2 si basano sui chip AWS Inferentia, che offrono prestazioni elevate e inferenze a costi contenuti 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 TensorFlow PyTorch, e Apache MXNet.

Considerazioni

Prima di iniziare a implementare Neuron su Amazon ECS, tieni presente quanto segue:

- I cluster possono contenere una combinazione di istanze Trn1, Inf1, Inf2 e altre.
- È necessaria un'applicazione Linux in un contenitore che utilizza un framework di apprendimento automatico che supporti Neuron. AWS

  > **Important**
  >
  > Le applicazioni che utilizzano altri framework potrebbero non avere prestazioni migliorate sulle istanze Trn1, Inf1 e Inf2.

- Puoi eseguire soltanto un'attività di inferenza o di addestramento all'inferenza su ogni chip AWS Trainium o AWS Inferentia. Per Inf1, ogni chip ne ha 4. NeuronCores Per Trn1 e Inf2 ogni chip ne ha 2. NeuronCores Puoi eseguire tante attività quanti sono i chip per ciascuna istanza Trn1, Inf1 e Inf2.

- Durante l'esecuzione di un servizio o di un'attività autonoma, puoi utilizzare gli attributi del tipo di istanza quando configuri i vincoli di posizionamento dell'attività. Ciò garantisce che l'attività venga avviata sull'istanza di container specificata. In questo modo puoi ottimizzare l'utilizzo complessivo delle risorse e garantire che le attività per i carichi di lavoro di inferenza si trovino nelle istanze Trn1, Inf1 e Inf2. Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

  Nell'esempio seguente viene eseguita un'attività su una istanza `Inf1.xlarge` sul cluster `default`.

  ```
  aws ecs run-task \
       --cluster default \
       --task-definition ecs-inference-task-def \
       --placement-constraints type=memberOf,expression="attribute:ecs.instance-type ==
   Inf1.xlarge"
  ```

- I requisiti di risorse Neuron non possono essere definiti in una definizione di attività. Invece, configuri un contenitore per utilizzare chip AWS Trainium o AWS Inferentia specifici disponibili sull'istanza del contenitore host. Puoi farlo usando il parametro `linuxParameters` e specificando i dettagli del dispositivo. Per ulteriori informazioni, consulta Requisiti di definizione di attività.

Usa l'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS

Amazon ECS fornisce un'AMI ottimizzata per Amazon ECS basata su Amazon Linux 2023 per carichi di lavoro AWS Trainium e AWS Inferentia. Viene fornito con i driver AWS Neuron e il runtime per Docker. Questa AMI semplifica l'esecuzione dei carichi di lavoro di inferenza di machine learning su Amazon ECS.

Ti consigliamo di utilizzare l'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS all'avvio delle istanze Amazon EC2 Trn1, Inf1 e Inf2.

Puoi recuperare l'attuale AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS AWS CLI utilizzando il comando seguente.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/neuron/
recommended
```

L'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS è supportata nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)

- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)
- Europa (Stoccolma)
- Sud America (San Paolo)

Usa l'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS

Amazon ECS fornisce un'AMI ottimizzata per Amazon ECS basata su Amazon Linux 2 per carichi di lavoro AWS Trainium e AWS Inferentia. Viene fornito con i driver AWS Neuron e il runtime per Docker. Questa AMI semplifica l'esecuzione dei carichi di lavoro di inferenza di machine learning su Amazon ECS.

Ti consigliamo di utilizzare l'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS quando avvii le istanze Trn1, Inf1 e Inf2 di Amazon EC2.

Puoi recuperare l'attuale AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS utilizzando il comando AWS CLI seguente.

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/inf/
recommended
```

L'AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS è supportata nelle seguenti regioni:

- Stati Uniti orientali (Virginia settentrionale)
- Stati Uniti orientali (Ohio)
- Stati Uniti occidentali (California settentrionale)
- Stati Uniti occidentali (Oregon)
- Asia Pacifico (Mumbai)
- Asia Pacific (Osaka)
- Asia Pacific (Seul)
- Asia Pacifico (Tokyo)
- Asia Pacifico (Singapore)
- Asia Pacifico (Sydney)
- Canada (Centrale)
- Europa (Francoforte)
- Europa (Irlanda)
- Europe (London)
- Europe (Paris)
- Europa (Stoccolma)
- Sud America (San Paolo)

Requisiti di definizione di attività

Per distribuire Neuron su Amazon ECS, la definizione dell'attività deve contenere la definizione del contenitore per un contenitore predefinito che serve il modello di inferenza per. TensorFlow È fornito da AWS Deep Learning Containers. Questo contenitore contiene il runtime AWS Neuron e l'applicazione TensorFlow Serving. All'avvio, questo contenitore recupera il modello da Amazon S3, avvia TensorFlow Neuron Serving con il modello salvato e attende le richieste di previsione. Nell'esempio seguente, l'immagine del contenitore ha 1.15 e Ubuntu 18.04. TensorFlow È disponibile un elenco completo di Deep Learning Containers predefiniti ottimizzati per Neuron. GitHub Per ulteriori informazioni, consulta Using AWS TensorFlow Neuron Serving.

```
763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-inference-neuron:1.15.4-neuron-py37-ubuntu18.04
```

In alternativa, puoi creare la tua immagine di container sidecar di Neuron. Per ulteriori informazioni, consulta Tutorial: Neuron TensorFlow Serving nella Guida per gli AWS Deep Learning AMI 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	vCPU	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	vCPU	RAM (GiB)	AWS Chip acceleratori ML	Percorsi dispositivi
				`/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	vCPU	RAM (GiB)	AWS Chip acceleratori ML	Percorsi dispositivi
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	vCPU	RAM (GiB)	AWS Chip acceleratori ML	Percorsi dispositivi
inf2.24xlarge	96	384	6	`/dev/neuron0 ,/dev/neuron1 ,/dev/neuron2 ,/dev/neuron3 ,/dev/neuron4 ,/dev/neuron5 ,`
inf2.48xlarge	192	768	12	`/dev/neuron0 ,/dev/neuron1 ,/dev/neuron2 ,/dev/neuron3 ,/dev/neuron4 ,/dev/neuron5 ,/dev/neuron6 ,/dev/neuron7 ,/dev/neuron8 ,/dev/neuron9 ,/dev/neuron10 ,/dev/neuron11`

Specificare l'apprendimento automatico di AWS Neuron in una definizione di attività Amazon ECS

Di seguito è riportato un esempio di definizione di attività Linux per `inf1.xlarge` che riporta la sintassi da utilizzare.

```
{
    "family": "ecs-neuron",
    "requiresCompatibilities": ["EC2"],
    "placementConstraints": [
        {
            "type": "memberOf",
            "expression": "attribute:ecs.os-type == linux"
        },
        {
            "type": "memberOf",
            "expression": "attribute:ecs.instance-type == inf1.xlarge"
        }
    ],
    "executionRoleArn": "${YOUR_EXECUTION_ROLE}",
    "containerDefinitions": [
        {
            "entryPoint": [
                "/usr/local/bin/entrypoint.sh",
                "--port=8500",
                "--rest_api_port=9000",
                "--model_name=resnet50_neuron",
                "--model_base_path=s3://your-bucket-of-models/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 carichi di lavoro di deep learning su Amazon ECS, registra le istanze DL1 di Amazon EC2 nei cluster. Le istanze DL1 di Amazon EC2 sono alimentate da 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 a implementare DL1 su Amazon ECS, tieni presente quanto segue:

- I cluster possono contenere una combinazione di istanze DL1 e non 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 assicura che le risorse vengano utilizzate in modo efficace e che le attività per i carichi di lavoro di deep learning si trovino

nelle istanze DL1. Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

Nell'esempio seguente viene eseguito un processo su una istanza `dl1.24xlarge` sul cluster `default`.

```
aws ecs run-task \
     --cluster default \
     --task-definition ecs-dl1-task-def \
     --placement-constraints type=memberOf,expression="attribute:ecs.instance-type ==
 dl1.24xlarge"
```

Utilizzo di un'AMI DL1

Sono disponibili tre opzioni per l'esecuzione di un'AMI su istanze DL1 di Amazon EC2 per Amazon ECS:

- Marketplace AWS Le AMI fornite da Habana sono disponibili qui.
- Le AMI di deep learning Habana fornite 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 GitHubrepository. Per ulteriori informazioni, consulta la documentazione di Packer.

Specificare il deep learning in una definizione di attività Amazon ECS

Per eseguire i contenitori di deep learning accelerato di Habana Gaudi su Amazon ECS, la definizione dell'attività deve contenere la definizione del contenitore per un contenitore predefinito che serve il modello di deep learning per TensorFlow o che PyTorch utilizza Habana SynapseAI fornito da Deep Learning Containers. AWS

L'immagine seguente del contenitore ha 2.7.0 e Ubuntu 20.04. TensorFlow Viene mantenuto un elenco completo di Deep Learning Containers predefiniti ottimizzati per gli acceleratori Habana Gaudi. GitHub Per ulteriori informazioni, consulta Habana Training Containers (Container di training Habana).

```
763104351884.dkr.ecr.us-east-1.amazonaws.com/tensorflow-training-habana:2.7.0-hpu-py38-
synapseai1.2.0-ubuntu20.04
```

Di seguito è riportata una definizione di attività di esempio per container Linux su Amazon EC2, che riporta la sintassi da utilizzare. In questo esempio viene utilizzata un'immagine contenente l'Habana Labs System Management Interface Tool (HL-SMI) che trovi qui: `vault.habana.ai/gaudi-docker/1.1.0/ubuntu20.04/habanalabs/tensorflow-installer-tf-cpu-2.6.0:1.1.0-614`

```
{
    "family": "dl-test",
    "requiresCompatibilities": ["EC2"],
    "placementConstraints": [
        {
            "type": "memberOf",
            "expression": "attribute:ecs.os-type == linux"
        },
        {
            "type": "memberOf",
            "expression": "attribute:ecs.instance-type == dl1.24xlarge"
        }
    ],
    "networkMode": "host",
    "cpu": "10240",
    "memory": "1024",
    "containerDefinitions": [
        {
            "entryPoint": [
                "sh",
                "-c"
            ],
            "command": ["hl-smi"],
            "cpu": 8192,
            "environment": [
                {
                    "name": "HABANA_VISIBLE_DEVICES",
                    "value": "all"
                }
            ],
            "image": "vault.habana.ai/gaudi-docker/1.1.0/ubuntu20.04/habanalabs/
tensorflow-installer-tf-cpu-2.6.0:1.1.0-614",
            "essential": true,
            "name": "tensorflow-installer-tf-hpu"
        }
    ]
```

```
}
```

Definizioni delle attività di Amazon ECS per carichi di lavoro ARM a 64 bit

Amazon ECS supporta l'utilizzo di applicazioni ARM a 64 bit. Puoi eseguire le tue applicazioni sulla piattaforma basata sui processori Graviton2.AWS adatti per un'ampia varietà di carichi di lavoro, tra cui server applicativi, micro-servizi, elaborazione ad alte prestazioni, inferenza di machine learning basata su CPU, codifica video, automazione della progettazione elettronica, giochi, database open-source e cache in memoria.

Considerazioni

Prima di iniziare a implementare le definizioni di attività che utilizzano l'architettura ARM a 64 bit, tieni in considerazione quanto segue:

- Le applicazioni possono utilizzare i tipi di avvio Fargate o EC2.
- Le attività Linux con architettura ARM64 non supportano il provider di capacità Fargate Spot.
- Le applicazioni possono utilizzare solo il sistema operativo Linux.
- Per il tipo Fargate, le applicazioni devono utilizzare la versione `1.4.0` o versione successiva della piattaforma Fargate.
- Le applicazioni possono utilizzare Fluent Bit o CloudWatch per il monitoraggio.
- Per il tipo di lancio Fargate, quanto segue Regioni AWS non supporta i carichi di lavoro ARM a 64 bit:
  - Stati Uniti orientali (Virginia settentrionale), la zona di disponibilità `use1-az3`
- Per il tipo di avvio Amazon EC2, consulta quanto segue per verificare che la tua regione 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 comando `describe-instance-type-offerings` di Amazon EC2 con un filtro per visualizzare l'offerta di istanze per la tua regione.

```
aws ec2 describe-instance-type-offerings --filters Name=instance-
type,Values=instance-type --region region
```

Nell'esempio seguente viene verificata la disponibilità del tipo di istanza M6 nella regione Stati Uniti orientali (Virginia settentrionale) (us-east-1).

```
aws ec2 describe-instance-type-offerings --filters "Name=instance-type,Values=m6*" --
region us-east-1
```

Per ulteriori informazioni, consulta describe-instance-type-offerings Amazon EC2 Command Line Reference.

Specificazione dell'architettura ARM in una definizione di attività Amazon ECS

Per utilizzare l'architettura ARM, specifica `ARM64` per il parametro di definizione di attività `cpuArchitecture`.

Nell'esempio seguente, l'architettura ARM viene specificata in una definizione di attività. È in formato JSON.

```
{
    "runtimePlatform": {
        "operatingSystemFamily": "LINUX",
        "cpuArchitecture": "ARM64"
    },
...
}
```

Nell'esempio seguente, viene riportata una definizione di attività per l'architettura ARM che visualizza "hello world".

```
{
 "family": "arm64-testapp",
 "networkMode": "awsvpc",
 "containerDefinitions": [
    {
        "name": "arm-container",
        "image": "arm64v8/busybox",
        "cpu": 100,
```

```
        "memory": 100,
        "essential": true,
        "command": [ "echo hello world" ],
        "entryPoint": [ "sh", "-c" ]
    }
 ],
 "requiresCompatibilities": [ "FARGATE" ],
 "cpu": "256",
 "memory": "512",
 "runtimePlatform": {
        "operatingSystemFamily": "LINUX",
        "cpuArchitecture": "ARM64"
  },
 "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
}
```

Invia i log di Amazon ECS a CloudWatch

È possibile configurare i contenitori delle attività per inviare informazioni di registro ai CloudWatch registri. Se utilizzi il tipo di avvio Fargate per le tue attività, puoi visualizzare i log provenienti dai tuoi container. Se utilizzi il tipo di avvio EC2, puoi visualizzare diversi log dei container in un'unica comoda posizione, evitando inoltre che i log dei container occupino spazio su disco nelle istanze di container.

> **Note**
>
> Il tipo di informazioni registrate dai container nell'attività dipende per lo più dal relativo comando `ENTRYPOINT`. Per impostazione predefinita, i log acquisiti mostrano l'output di comando che normalmente viene visualizzato in un terminale interattivo se si esegue il container in locale, ovvero i flussi I/O `STDOUT` e `STDERR`. Il driver di `awslogs` registro passa semplicemente questi log da Docker a Logs. CloudWatch Per ulteriori informazioni su come vengono elaborati i log Docker, inclusi metodi alternativi per acquisire diversi flussi o dati di file, consulta l'articolo relativo alla visualizzazione di log per un container o servizio nella documentazione di Docker.

Per inviare i log di sistema dalle istanze di container Amazon ECS a CloudWatch Logs, consulta Monitoring Log Files e CloudWatch Logs quote nella Amazon Logs User Guide. CloudWatch

Tipo di avvio di Fargate

Se utilizzi il tipo di avvio Fargate per le tue attività, per attivare il driver di log `logConfiguration` devi aggiungere i parametri `awslogs` necessari alla definizione di attività. Per ulteriori informazioni, consulta Esempio di definizione di attività Amazon ECS: indirizza i log verso CloudWatch.

Per il contenitore Windows su Fargate, eseguite una delle seguenti opzioni quando uno dei parametri di definizione dell'attività contiene caratteri speciali come: `& \ < > ^ |`

- Aggiungi un escape (\) con virgolette doppie attorno all'intera stringa di parametri

  Esempio

  ```
  "awslogs-multiline-pattern": "\"^[|DEBUG|INFO|WARNING|ERROR\"",
  ```

- Aggiungi un carattere escape (^) attorno a ogni carattere speciale

  Esempio

  ```
  "awslogs-multiline-pattern": "^^[^|DEBUG^|INFO^|WARNING^|ERROR",
  ```

Tipo di avvio EC2

Se utilizzi il tipo di avvio EC2 per le attività e desideri attivare il driver di log `awslogs`, è necessario che le istanze di container Amazon ECS dispongano almeno della versione 1.9.0 dell'agente del container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta Aggiornamento dell'agente del container Amazon ECS.

> **Note**
>
> È necessario utilizzare un'AMI ottimizzata per Amazon ECS o un'AMI personalizzata con almeno la versione `1.9.0-1` del `ecs-init` pacchetto. Quando si utilizza un'AMI personalizzata, è necessario specificare che il driver di `awslogs` registrazione è disponibile sull'istanza Amazon EC2 all'avvio dell'agente utilizzando la seguente variabile di ambiente nell'istruzione o nel docker run file della variabile di ambiente.
>
> ```
> ECS_AVAILABLE_LOGGING_DRIVERS=["json-file","awslogs"]
> ```

Le istanze di container di Amazon ECS richiedono inoltre l'autorizzazione `logs:CreateLogStream` e `logs:PutLogEvents` nel ruolo IAM con cui avvii le istanze di container. Se hai creato il ruolo dell'istanza di container di Amazon ECS prima che fosse abilitato il supporto per il driver di log `awslogs` in Amazon ECS, potrebbe essere necessario aggiungere questa autorizzazione. `ecsTaskExecutionRole` viene utilizzato quando viene assegnato all'attività e deve contenere le autorizzazioni corrette. Per informazioni sul ruolo di esecuzione delle attività, consulta. Ruolo IAM di esecuzione di attività Amazon ECS Se nelle istanze di container viene utilizzata l'apposita policy IAM gestita, le tue istanze di container dispongono delle autorizzazioni corrette. Per informazioni sulla politica IAM gestita per le istanze di container, consultaRuolo IAM delle istanze di container Amazon ECS.

Esempio di definizione di attività Amazon ECS: indirizza i log verso CloudWatch

Prima che i contenitori possano inviare i log a CloudWatch, è necessario specificare il driver di `awslogs` registro per i contenitori nella definizione dell'attività. Per ulteriori informazioni sui parametri di registro, vedere Archiviazione e registrazione

La definizione di attività JSON che segue ha un oggetto `logConfiguration` specificato per ogni container. Uno è per il WordPress contenitore che invia i log a un gruppo di log chiamato`awslogs-wordpress`. L'altro è per un container MySQL che invia i log a un gruppo di log chiamato `awslogs-mysql`. Entrambi i container utilizzano il prefisso `awslogs-example` per il flusso di log.

```
{
    "containerDefinitions": [
        {
            "name": "wordpress",
            "links": [
                "mysql"
            ],
            "image": "wordpress",
            "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": "mysql",
            "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"
}
```

Dopo aver registrato una definizione di attività con il driver di `awslogs` registro in una configurazione di log di definizione del contenitore, 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, consulta Esecuzione di un'applicazione come attività Amazon ECS e Creazione di un servizio Amazon ECS utilizzando la console.

Inviare i log di Amazon ECS a un servizio o AWSAWS 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.

Considera quanto segue quando utilizzi FireLens Amazon ECS:

- Ti consigliamo di `my_service_` aggiungere il nome del contenitore di log in modo da poter distinguere facilmente i nomi dei contenitori nella console.
- Amazon ECS aggiunge per impostazione predefinita una dipendenza dall'ordine iniziale del contenitore tra i contenitori dell'applicazione e il FireLens contenitore. Quando specifichi un ordine del contenitore tra i contenitori dell'applicazione e il FireLens contenitore, l'ordine di avvio del contenitore predefinito viene sovrascritto.
- FireLensper 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 usare dei AWS CloudFormation modelli per configurare FireLens Amazon ECS. Per ulteriori informazioni, consulta AWS::ECS::TaskDefinition FirelensConfigurationla Guida per l'utente AWS CloudFormation
- FireLensascolta 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 attività che utilizzano la modalità di rete `host`, questo è il gruppo di sicurezza associato all'istanza Amazon EC2 che ospita l'attività. Per attività che utilizzano la modalità di rete `bridge`, non creare mappature di porte che utilizzano la porta `24224`.
- Per le attività che utilizzano la modalità di `bridge` rete, il contenitore con la FireLens configurazione deve avviarsi prima dell'avvio di qualsiasi contenitore di applicazioni che si basa su di essa. Per controllare l'ordine di avvio dei container, utilizza le condizioni di dipendenza nella definizione di attività. Per ulteriori informazioni, consulta Dipendenze per i container.

> **Note**
>
> Se utilizzi i parametri delle condizioni di dipendenza nelle definizioni dei contenitori con una FireLens configurazione, assicurati che ogni contenitore abbia un requisito di `HEALTHY` condizione `START` or.

- Per impostazione predefinita, FireLens aggiunge il nome della definizione del cluster e dell'attività e il nome della risorsa Amazon (ARN) del cluster come chiavi di metadati per i log del container stdout/stderr. Di seguito è riportato un esempio del formato dei metadati.

```
"ecs_cluster": "cluster-name",
"ecs_task_arn": "arn:aws:ecs:region:111122223333:task/cluster-
name/f2ad7dba413f45ddb4EXAMPLE",
"ecs_task_definition": "task-def-name:revision",
```

  Se non desideri i metadati nei tuoi log, imposta `enable-ecs-log-metadata` su `false` nella sezione `firelensConfiguration` della definizione di attività.

```
"firelensConfiguration":{
   "type":"fluentbit",
   "options":{
      "enable-ecs-log-metadata":"false",
      "config-file-type":"file",
      "config-file-value":"/extra.conf"
}
```

Per utilizzare questa funzionalità, devi creare un ruolo IAM per le tue attività che fornisca le autorizzazioni necessarie per utilizzare tutti AWS i servizi richiesti dalle attività. Ad esempio, se un contenitore sta instradando i log verso Firehose, l'operazione richiede l'autorizzazione per chiamare l'API. `firehose:PutRecordBatch` Per ulteriori informazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente di IAM.

La tua attività potrebbe anche richiedere il ruolo di esecuzione delle attività di Amazon ECS nelle seguenti condizioni. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

- Se la tua attività è ospitata su Fargate e stai estraendo immagini di container da Amazon ECR o stai facendo riferimento AWS Secrets Manager a dati sensibili dalla tua configurazione di registro, devi includere il ruolo IAM di esecuzione dell'attività.
- Quando utilizzi un file di configurazione personalizzato ospitato in Amazon S3, il ruolo IAM per l'esecuzione delle attività deve includere l'`s3:GetObject`autorizzazione.

Per informazioni su come utilizzare più file di configurazione con Amazon ECS, inclusi i file ospitati o i file in Amazon S3, consulta Processo di avvio per Fluent Bit on ECS, supporto multi-config.

Configurazione dei log di Amazon ECS per un throughput elevato

Quando si crea una definizione di attività, è possibile specificare il numero di righe di registro memorizzate nel buffer di memoria specificando il valore in. `log-driver-buffer-limit` Per ulteriori informazioni, consulta la pagina Driver di registro di Fluentd nella documentazione Docker.

Utilizza questa opzione quando la velocità effettiva è elevata, poiché Docker potrebbe esaurire la memoria buffer e scartare i messaggi del buffer, in modo da poter aggiungere nuovi messaggi.

Considera quanto segue quando utilizzi FireLens Amazon ECS con l'opzione di limite di buffer:

- Questa opzione è supportata nel tipo di avvio Amazon EC2 e sul tipo di avvio Fargate con versione `1.4.0` o successiva della piattaforma.
- L'opzione è valida solo quando `logDriver` è impostato su `awsfirelens`.
- Il limite di buffer predefinito è `1048576` rappresentato dalle righe di registro.
- I valori validi sono `0` e le righe di `536870912` registro.
- La quantità massima di memoria utilizzata per questo buffer è il prodotto della dimensione di ogni riga di registro e della dimensione del buffer. Ad esempio, se le righe di registro dell'applicazione sono in media 2 KB, un limite di buffer di 4096 utilizzerebbe al massimo 8 MiB. La quantità totale di memoria allocata a livello di attività deve essere maggiore della quantità di memoria allocata per tutti i contenitori oltre al buffer di memoria del driver di registro.

Quando il driver di log `awsfirelens` è specificato in una definizione di attività, l'agente Amazon ECS inserisce le seguenti variabili di ambiente nel container:

`FLUENT_HOST`

L'indirizzo IP assegnato al contenitore. FireLens

`FLUENT_PORT`

La porta su cui il protocollo Fluent Forward è in ascolto.

Le variabili di ambiente `FLUENT_HOST` e `FLUENT_PORT` consentono di accedere direttamente al router di log dal codice anziché passare attraverso `stdout`. Per ulteriori informazioni, vedere fluent-logger-golang on. GitHub

Di seguito viene mostrata la sintassi per specificare il. `log-driver-buffer-limit` Sostituisci `my_service_` con il nome del tuo servizio:

```
{
    "containerDefinitions": [
        {
            "essential": true,
            "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-
bit:stable",
            "name": "my_service_log_router",
            "firelensConfiguration": {
                "type": "fluentbit"
            },
            "logConfiguration": {
                "logDriver": "awslogs",
                "options": {
                    "awslogs-group": "firelens-container",
                    "awslogs-region": "us-west-2",
                    "awslogs-create-group": "true",
                    "awslogs-stream-prefix": "firelens"
                }
            },
            "memoryReservation": 50
        },
        {
            "essential": true,
            "image": "httpd",
            "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 Bitimmagine 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 nella maggior parte dei casi per l'alta disponibilità. Regioni AWS

Galleria pubblica di Amazon ECR

L'Fluent Bitimmagine AWS per è disponibile nella Amazon ECR Public Gallery. Questa è la posizione consigliata per scaricare l'Fluent Bitimmagine AWS for perché è un archivio pubblico e può essere utilizzato da tutti. Regioni AWS Per maggiori dettagli, consulta aws-for-fluent-bit nella galleria pubblica di Amazon ECR.

Linux

L'Fluent Bitimmagine AWS for nella galleria pubblica di Amazon ECR supporta il sistema operativo Amazon Linux con l'`x86-64`architetturaARM `64`, 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
> ```

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. Queste immagini sono disponibili nella maggior parte dei casi Regioni AWS, tra cui. AWS GovCloud (US)

Linux

L'URI dell'immagine stabile più recente AWS per Fluent Bit può essere recuperata utilizzando il seguente comando.

```
aws ssm get-parameters \
      --names /aws/service/aws-for-fluent-bit/stable \
      --region us-east-1
```

Tutte le versioni dell'immagine AWS for Fluent Bit possono essere elencate utilizzando il comando seguente per interrogare il parametro Systems Manager Parameter Store.

```
aws ssm get-parameters-by-path \
      --path /aws/service/aws-for-fluent-bit \
      --region us-east-1
```

È possibile fare riferimento all'immagine stabile più recente AWS per Fluent Bit in un AWS CloudFormation modello facendo riferimento al nome dell'archivio dei parametri di Systems Manager. Di seguito è riportato un esempio:

```
Parameters:
  FireLensImage:
    Description: Fluent Bit image for the FireLens Container
    Type: AWS::SSM::Parameter::Value<String>
    Default: /aws/service/aws-for-fluent-bit/stable
```

Windows

L'ultima versione stabile dell'URI AWS di immagine Fluent Bit può essere recuperata utilizzando il seguente comando.

```
aws ssm get-parameters \
      --names /aws/service/aws-for-fluent-bit/windowsservercore-stable \
      --region us-east-1
```

Tutte le versioni dell'immagine AWS for Fluent Bit possono essere elencate utilizzando il comando seguente per interrogare il parametro Systems Manager Parameter Store.

```
aws ssm get-parameters-by-path \
      --path /aws/service/aws-for-fluent-bit/windowsservercore \
      --region us-east-1
```

È possibile fare riferimento all'ultima immagine stabile AWS per Fluent Bit in un AWS CloudFormation modello facendo riferimento al nome dell'archivio dei parametri di Systems Manager. Di seguito è riportato un esempio:

```
Parameters:
  FireLensImage:
    Description: Fluent Bit image for the FireLens Container
    Type: AWS::SSM::Parameter::Value<String>
    Default: /aws/service/aws-for-fluent-bit/windowsservercore-stable
```

Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens

Per utilizzare il routing 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 il AWS Management Console, è disponibile una sezione di FireLens integrazione che semplifica l'aggiunta di un contenitore di log router. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

Amazon ECS converte la configurazione del log e genera la configurazione di output Fluentd o Fluent Bit. La configurazione di output è montata nel container di routing dei log in corrispondenza di `/fluent-bit/etc/fluent-bit.conf` for Fluent Bit e `/fluentd/etc/fluent.conf` for Fluentd.

> **Important**
>
> FireLens 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 attività che utilizzano la modalità di rete `host`, questo è il gruppo di sicurezza associato all'istanza Amazon EC2 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'istanza Amazon EC2 su cui il container è ospitato. Questo campo è valido solo per processi che utilizzano il tipo di avvio EC2.

Puoi impostarlo su `false` se non desideri i metadati. `enable-ecs-log-metadata`

Il seguente esempio di definizione di attività definisce un contenitore di log router che utilizza Fluent Bit per indirizzare i log verso Logs. CloudWatch Definisce inoltre un contenitore di applicazioni che utilizza una configurazione di log per indirizzare i log verso Amazon Data Firehose e imposta la memoria utilizzata per bufferizzare gli eventi su 2 MiB.

> **Note**
>
> Per ulteriori esempi di definizioni di attività, consulta gli FireLensesempi di Amazon ECS su GitHub.

```
{
  "family": "firelens-example-firehose",
  "taskRoleArn": "arn:aws:iam::123456789012:role/ecs_task_iam_role",
  "containerDefinitions": [
    {
      "essential": true,
      "image": "906394416424.dkr.ecr.us-west-2.amazonaws.com/aws-for-fluent-
bit:stable",
      "name": "log_router",
      "firelensConfiguration": {
        "type": "fluentbit",
        "options": {
          "enable-ecs-log-metadata": "true"
        }
      },
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "firelens-container",
          "awslogs-region": "us-west-2",
          "awslogs-create-group": "true",
          "awslogs-stream-prefix": "firelens"
        }
      },
      "memoryReservation": 50
    },
    {
      "essential": true,
      "image": "httpd",
```

```
      "name": "app",
      "logConfiguration": {
        "logDriver": "awsfirelens",
        "options": {
          "Name": "firehose",
          "region": "us-west-2",
          "delivery_stream": "my-stream",
          "log-driver-buffer-limit": "2097152"
        }
      },
      "memoryReservation": 100
    }
  ]
}
```

Le coppie chiave/valore specificate come opzioni nell'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 si specifica la `match` configurazione nella definizione dell'attività.

Utilizza un file di configurazione personalizzato

È possibile specificare un file di configurazione personalizzato. Il formato del file di configurazione è il formato nativo per il router di log in uso. Per ulteriori informazioni, consulta Fluentd Config File Syntax e Fluent Bit Configuration File.

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:::mybucket/fluent.conf | filepath"
        }
      }
    }
  ]
}
```

> **Note**
>
> Le attività ospitate su supportano AWS Fargate solo il tipo `file` di file di configurazione.

Utilizzo di immagini non AWS containerizzate in Amazon ECS

Utilizzate il registro privato per memorizzare le vostre credenziali e AWS Secrets Manager quindi fate riferimento ad esse nella definizione dell'attività. In questo modo è possibile fare riferimento alle immagini dei contenitori presenti in registri privati AWS che non richiedono l'autenticazione nelle definizioni delle attività. Questa funzionalità è supportata da processi ospitati su istanze Fargate, Amazon EC2 e 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 istanze Amazon EC2, questa funzione richiede la versione `1.19.0` 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 i processi ospitati su Fargate, questa funzione richiede la versione `1.2.0`o successiva della piattaforma. Per informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.

All'interno della definizione del container, specifica l'oggetto `repositoryCredentials` con i dettagli del segreto che hai creato. Il segreto a cui si fa riferimento può provenire da un account diverso Regione AWS o diverso da quello dell'attività che lo utilizza.

> **Note**
>
> Quando utilizzi l'API o l' AWS SDK di Amazon ECS, se il segreto esiste nella Regione AWS stessa attività che stai avviando, puoi utilizzare l'ARN completo o il nome del segreto. AWS CLI Se il segreto esiste in un altro account, occorre specificare l'ARN completo del segreto. Quando si utilizza AWS Management Console, è necessario specificare sempre l'ARN completo del segreto.

Di seguito viene riportato un frammento di una definizione di attività che mostra i parametri obbligatori:

Sostituisci private-repo con il nome host del repository privato e private-image con il nome dell'immagine.

```
"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 i processi ospitati su istanze Amazon EC2. Per ulteriori informazioni, consulta Configurazione delle istanze di container Amazon ECS per immagini Docker private .

Per utilizzare il registro privato

1. La definizione dell'attività deve avere un ruolo di esecuzione dell'attività. In questo modo l'agente container può recuperare l'immagine del container. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

   Le autorizzazioni seguenti devono essere aggiunte manualmente come policy inline al ruolo per l'esecuzione di attività, per fornire l'accesso ai segreti che crei. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

   - `secretsmanager:GetSecretValue`
   - `kms:Decrypt`: obbligatorio solo se la chiave utilizza una chiave KMS personalizzata e non quella di default. Il nome della risorsa Amazon (ARN) per la chiave personalizzata deve essere aggiunto come risorsa.

   Di seguito viene riportata una policy inline che aggiunge le autorizzazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [

 "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
                "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
```

```
            ]
        }
    ]
}
```

2. AWS Secrets Manager Utilizzatelo per creare un segreto per le credenziali del registro privato. Per informazioni su come creare un segreto, consulta Create an AWS Secrets Manager secret nella Guida per l'AWS Secrets Manager utente.

   Immettete le credenziali del registro privato utilizzando il seguente formato:

   ```
   {
     "username" : "privateRegistryUsername",
     "password" : "privateRegistryPassword"
   }
   ```

3. Registra una definizione di attività. Per ulteriori informazioni, consulta the section called "Creazione di una definizione di attività attraverso la nuova console".

Passa una singola variabile di ambiente a un contenitore Amazon ECS

> **Important**
>
> Ti consigliamo di archiviare i dati sensibili nei parametri AWS Secrets Manager segreti o AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.
> Le variabili di ambiente specificate nella definizione di attività sono leggibili da tutti gli utenti IAM e i ruoli per i quali è consentita l'operazione `DescribeTaskDefinition` per la definizione di attività.

Puoi passare le variabili di ambiente ai container nei seguenti modi:

- Individualmente utilizzando il parametro di definizione del container `environment`. Questo viene mappato all'opzione `--env` su docker run.
- In blocco, utilizzando il parametro di definizione del container `environmentFiles` per elencare uno o più file contenenti le variabili di ambiente. Il file deve essere ospitato in Amazon S3. Questo viene mappato all'opzione `--env-file` su docker run.

Di seguito è riportato uno snippet di una definizione di attività che mostra come specificare singole variabili di ambiente.

```
{
    "family": "",
    "containerDefinitions": [
        {
            "name": "",
            "image": "",
            ...
            "environment": [
                {
                    "name": "variable",
                    "value": "value"
                }
            ],
            ...
        }
    ],
    ...
}
```

Passa le variabili di ambiente a un contenitore Amazon ECS

> **Important**
>
> Ti consigliamo di archiviare i dati sensibili nei parametri AWS Secrets Manager segreti o AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.
> I file variabili di ambiente sono oggetti in Amazon S3 e si applicano tutte le considerazioni sulla sicurezza di Amazon S3.
> Non è possibile utilizzare il `environmentFiles` parametro sui contenitori Windows e sui contenitori Windows su Fargate.

Puoi creare un file di variabili di ambiente e archiviarlo in Amazon S3 per passare le variabili di ambiente al tuo contenitore.

Specificando le variabili di ambiente in un file, puoi inserire in blocco le variabili di ambiente. All'interno della definizione del container, specifica l'oggetto `environmentFiles` con un elenco di bucket Amazon S3 contenenti i file delle variabili di ambiente.

Amazon ECS non impone un limite di dimensione per le variabili di ambiente, ma un file di variabili di ambiente di grandi dimensioni potrebbe occupare lo spazio su disco. Ogni processo che utilizza un file di variabili di ambiente fa sì che una copia del file venga scaricata sul disco. Amazon ECS rimuove il file come parte della pulizia delle attività.

Per informazioni sulle variabili di ambiente supportate, consulta Parametri avanzati di definizione del container - Ambiente.

Quando si specifica un file di variabili di ambiente in una definizione di container, considera quanto segue.

- Per le attività Amazon ECS su Amazon EC2, per utilizzare questa funzione le istanze del container richiedono la versione `1.39.0` o successiva dell'agente del 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 le attività di Amazon ECS su AWS Fargate, le attività devono utilizzare una `1.4.0` versione della piattaforma o successiva (Linux) per utilizzare questa funzionalità. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.

  Verifica che la variabile sia supportata per la piattaforma del sistema operativo. Per ulteriori informazioni, consulta 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.
- 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, consulta Declare default environment variables in file (Dichiarazione delle variabili di ambiente in un file).

  Di seguito è riportata la sintassi appropriata.

```
#This is a comment and will be ignored
VARIABLE=VALUE
```

```
ENVIRONMENT=PRODUCTION
```

- Se sono specificate variabili di ambiente utilizzando il parametro `environment` in una definizione di container, queste hanno la precedenza sulle variabili contenute in un file di ambiente.
- Se vengono specificati più file di ambiente che contengono la stessa variabile, vengono elaborati in ordine di inserimento. Ciò significa che viene utilizzato il primo valore della variabile e i valori successivi delle variabili duplicate vengono ignorati. Consigliamo di utilizzare nomi di variabili univoci.
- Se un file di ambiente viene specificato come override di container, viene utilizzato. Inoltre, tutti gli altri file di ambiente specificati nella definizione del container vengono ignorati.
- Le seguenti regole si applicano al tipo di avvio Fargate:
  - Il file viene gestito come un file env Docker nativo.
  - 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`.

Autorizzazioni IAM richieste

Per utilizzare questa funzionalità, è necessario il ruolo per l'esecuzione del processo di Amazon ECS. 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.

Per fornire l'accesso agli oggetti Amazon S3 che crei, aggiungi manualmente le seguenti autorizzazioni come policy in linea al ruolo per l'esecuzione del processo. Utilizza il parametro `Resource` per definire l'ambito dell'autorizzazione per i bucket Amazon S3 che contengono i file delle variabili di ambiente. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

- `s3:GetObject`
- `s3:GetBucketLocation`

Nel seguente esempio, le autorizzazioni vengono aggiunte a una policy in linea.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/folder_name/env_file_name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket"
      ]
    }
  ]
}
```

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:::s3_bucket_name/envfile_object_name.env",
                    "type": "s3"
                }
            ],
            ...
        }
    ],
    ...
}
```

Trasferisci dati sensibili a un contenitore Amazon ECS

Puoi inviare in sicurezza dati sensibili, come le credenziali di un database, al tuo container.

È possibile utilizzare Secrets Manager o come parametro in Systems Manager Parameter Store per memorizzare il segreto.

È possibile recuperare i segreti a livello di codice dall'applicazione o utilizzando variabili di ambiente.

Per iniziare, memorizza prima i dati sensibili come segreti in Secrets Manager o come parametro in Systems Manager Parameter Store. Utilizza quindi uno dei modi seguenti per esporre il segreto al container.

Argomenti

Le migliori pratiche per la gestione dei segreti in Amazon ECS

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 limitati a principali IAM specifici che utilizzano IAM e inseriti nei container durante il runtime.

I segreti possono essere inseriti senza problemi nei contenitori da AWS Secrets Manager Amazon EC2 Systems Manager Parameter Store. È possibile fare riferimento a questi segreti nell'attività in qualsiasi dei modi seguenti.

1. Come variabili di ambiente che utilizzano il parametro di definizione di container `secrets`.
2. Come `secretOptions` se la piattaforma per la registrazione di log richiede l'autenticazione. Per ulteriori informazioni, consulta le opzioni di configurazione della registrazione di log.

3. Come segreti estratti da immagini che utilizzano il parametro di definizione di container `repositoryCredentials` se il registro da cui viene estratto il container richiede l'autenticazione. Utilizza questo metodo quando estrai immagini da Amazon ECR Public Gallery. Per ulteriori informazioni, consulta Autenticazione del registro privato per le attività.

Segreti e raccomandazioni

Consigliamo di completare la procedura seguente quando configuri la gestione dei segreti.

Usa AWS Secrets Manager Amazon EC2 Systems Manager Parameter Store per archiviare materiali segreti

È necessario archiviare in modo sicuro le chiavi API, le credenziali del database e altri materiali segreti in AWS Secrets Manager o come parametri crittografati in Amazon EC2 Systems Manager Parameter Store. Questi servizi sono simili perché sono entrambi archivi chiave-valore gestiti che vengono utilizzati AWS KMS per crittografare dati sensibili. AWS Secrets Manager, tuttavia, include anche la possibilità di ruotare automaticamente i segreti, generare segreti casuali e condividere segreti tra gli account. AWS Se ritieni che queste funzionalità siano importanti, utilizza AWS Secrets Manager , altrimenti utilizza i parametri crittografati.

> **Note**
>
> Le attività che fanno riferimento a un indirizzo segreto AWS Secrets Manager o a Amazon EC2 Systems Manager Parameter Store richiedono un Task Execution Role con una policy che garantisca ad Amazon ECS l'accesso al segreto desiderato e, se applicabile, AWS KMS alla chiave utilizzata per crittografare e decrittografare quel segreto.

> **Important**
>
> I segreti a cui si fa riferimento nelle attività non vengono ruotati in automatico. 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:
>
> - AWS Secrets Manager: inserimento di dati come variabili di ambiente
> - Archivio dei parametri Systems Manager per Amazon EC2: inserimento di dati come variabili di ambiente

Recupera dati da un bucket Amazon S3 crittografato

Poiché il valore delle variabili di ambiente può fuoriuscire inavvertitamente nei log ed essere rivelato durante l'esecuzione di `docker inspect`, è consigliabile archiviare i segreti in un bucket Amazon S3 crittografato e utilizzare i ruoli di attività per limitare l'accesso a tali segreti. Quando esegui questa operazione, l'applicazione deve essere scritta in modo tale da leggere il segreto dal bucket Amazon S3. Per ricevere istruzioni, consulta Impostazione del comportamento predefinito della crittografia lato server per i bucket Amazon S3.

Montaggio del segreto in un volume con un container sidecar

Poiché esiste un rischio elevato di perdita di dati a causa delle variabili di ambiente, è consigliabile utilizzare un contenitore secondario che legga i segreti AWS Secrets Manager e li scriva su un volume condiviso. Questo container può essere eseguito e terminato prima del container dell'applicazione utilizzando l'ordinamento di container Amazon di ECS. Quando esegui questa operazione, il container dell'applicazione monta successivamente il volume in cui il segreto è stato scritto. Analogamente al metodo bucket Amazon S3, l'applicazione deve essere scritta per leggere il segreto dal volume condiviso. Poiché il suo ambito è limitato all'attività, il volume viene eliminato in automatico dopo l'interruzione dell'attività. Per un esempio di container sidecar, consulta il progetto aws-secret-sidecar-injector.

> **Note**
>
> In Amazon EC2, il volume su cui il segreto viene scritto può essere crittografato con una chiave AWS KMS gestita dal cliente. Attivato AWS Fargate, lo storage di volumi viene crittografato automaticamente utilizzando una chiave gestita dal servizio.

Risorse aggiuntive

- Trasmissione di segreti ai container in un'attività Amazon ECS
- Chamber è un wrapper per l'archiviazione di segreti in Archivio dei parametri Systems Manager per Amazon EC2

Recupera i segreti di Secrets Manager in modo programmatico in Amazon ECS

Utilizza Gestione dei segreti per proteggere i dati sensibili e ruotare, gestire e recuperare credenziali di database, chiavi API e altri segreti durante tutto il loro ciclo di vita.

Invece di codificare le informazioni sensibili in testo normale nell'applicazione, è possibile utilizzare Secrets Manager per archiviare i dati sensibili.

Consigliamo questo metodo di recupero dei dati sensibili perché, se il segreto di Gestione dei segreti verrà aggiornato in futuro, l'applicazione recupererà automaticamente l'ultima versione del segreto.

Creazione di un segreto in Secrets Manager. Dopo aver creato un segreto di Gestione dei segreti, aggiorna il codice dell'applicazione per recuperarlo.

Prima di proteggere i dati sensibili in Gestione dei segreti, considera i seguenti aspetti.

- Sono supportati solo i segreti che memorizzano dati di testo, ovvero segreti creati con il `SecretString` parametro dell'CreateSecretAPI. I segreti che memorizzano dati binari, ovvero segreti creati con il `SecretBinary` parametro dell'CreateSecretAPI, non sono supportati.
- Utilizza gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza. È necessario creare gli endpoint VPC di interfaccia per Gestione dei segreti. Per informazioni sull'endpoint VPC, consulta Creazione di endpoint VPC nella Guida per l'utente di AWS Secrets Manager .
- Il VPC utilizzato dalla tua attività deve utilizzare la risoluzione DNS.

Autorizzazioni IAM richieste

Per utilizzare questa funzionalità, è necessario disporre del ruolo di esecuzione dei processi di Amazon ECS e fare riferimento allo stesso nella definizione di attività. Per ulteriori informazioni, consulta Ruolo IAM dell'attività Amazon ECS.

Per fornire l'accesso ai segreti di Gestione di segreti che crei, aggiungi manualmente la seguente autorizzazione al ruolo di esecuzione dell'attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente IAM.

- `secretsmanager:GetSecretValue`: obbligatorio se si fa riferimento a un segreto di Gestione dei segreti. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

La policy dell'esempio seguente aggiunge le autorizzazioni necessarie.

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"
      ]
    }
  ]
}
```

Creazione del segreto di Gestione dei segreti

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni sulla creazione dei segreti, consulta Creazione di un segreto AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager .

Aggiornamento dell'applicazione per il recupero programmatico dei segreti di Gestione dei segreti

Puoi recuperare i segreti con una chiamata alle API di Gestione dei segreti direttamente dall'applicazione. Per informazioni, consulta Recupera segreti dalla AWS Secrets Manager Guida per l'AWS Secrets Manager utente.

Per recuperare i dati sensibili archiviati in AWS Secrets Manager, consulta Esempi di codice per l' AWS Secrets Manager utilizzo degli SDK nella libreria di codici AWSAWS SDK Code Examples.

Recupera 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. Puoi memorizzare dati come password, stringhe di database, ID di istanza EC2 e ID AMI e codici di licenza come valori dei parametri. Puoi memorizzare i valori in testo semplice o crittografati.

Invece di codificare le informazioni sensibili in testo normale nell'applicazione, è possibile utilizzare Secrets Manager per archiviare i dati sensibili.

Consigliamo questo metodo di recupero dei dati sensibili perché se il parametro Systems Manager Parameter Store viene successivamente aggiornato, l'applicazione recupera automaticamente la versione più recente.

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 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.

Autorizzazioni IAM richieste

Per utilizzare questa funzionalità, è necessario disporre del ruolo di esecuzione dei processi di Amazon ECS e fare riferimento allo stesso nella definizione di attività. Ciò consente all'agente del container di recuperare le risorse di Systems Manager necessarie. Per ulteriori informazioni, consulta Ruolo IAM dell'attività Amazon ECS.

> **Important**
>
> Per i processi che utilizzano il tipo di avvio EC2, devi utilizzare la variabile di configurazione dell'agente ECS `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` per utilizzare questa funzionalità. Puoi aggiungerlo al file `./etc/ecs/ecs.config` durante la creazione dell'istanza di container oppure aggiungerlo a un'istanza esistente e quindi riavviare l'agente ECS. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Per fornire l'accesso ai parametri di Archivio dei parametri Systems Manager che hai creato, aggiungi manualmente le autorizzazioni seguenti come policy al ruolo per l'esecuzione di attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente IAM.

- `ssm:GetParameters`: obbligatorio quando si fa riferimento a un parametro di Archivio dei parametri Systems Manager in una definizione di attività. Aggiunge l'autorizzazione per recuperare i parametri di Systems Manager.
- `secretsmanager:GetSecretValue`: obbligatorio quando si fa riferimento a un segreto di Gestione dei segreti direttamente o se il parametro di Archivio dei parametri Systems Manager

fa riferimento a un segreto di Gestione dei segreti in una definizione di attività. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave gestita personalizzata e non quella predefinita. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa. Aggiunge l'autorizzazione per decrittografare la chiave gestita dal cliente.

La seguente policy di esempio aggiunge le autorizzazioni necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:region:aws_account_id:parameter/parameter_name",
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name",
        "arn:aws:kms:region:aws_account_id:key/key_id"
      ]
    }
  ]
}
```

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 Systems Manager utilizzando gli SDK nella libreria di codici AWSAWS SDK Code Examples.

Recupera 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 o una versione specifica di un segreto da inserire. Questo processo consente di controllare i dati sensibili esposti al container. Per ulteriori informazioni sul controllo delle versioni dei segreti, consulta i termini e concetti chiave per AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager .

Quando si utilizza una variabile di ambiente per iniettare un segreto di Secrets Manager in un contenitore, è necessario considerare quanto segue.

- I dati sensibili vengono inseriti nel container quando viene inizialmente avviato. Se il segreto o parametro viene in seguito aggiornato o ruotato, il container non riceverà automaticamente il valore aggiornato. È necessario avviare una nuova attività o se l'attività è parte di un servizio, è possibile aggiornare il servizio e utilizzare la nuova opzione Force new deployment (Forza nuova implementazione) per forzare il servizio ad avviare una nuova attività.
- Per le attività di Amazon ECS su AWS Fargate, è necessario considerare quanto segue:
  - Per inserire il contenuto completo di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma `11.3.0` o successiva. Per informazioni, consulta Versioni della piattaforma Fargate Linux 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 Linux per Amazon ECS.
- Per i processi 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 del container deve avere la versione `1.37.0` o successiva dell'agente del container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta Aggiornamento dell'agente del container Amazon ECS.

    Per inserire il contenuto completo di un segreto come variabile di ambiente o per inserire un segreto in una configurazione di log, l'istanza del container deve avere la versione `1.22.0` o successiva dell'agente del container.
- Usa gli endpoint VPC dell'interfaccia per migliorare i controlli di sicurezza e connettiti a Secrets Manager tramite una sottorete privata. È necessario creare gli endpoint VPC di interfaccia per Gestione dei segreti. Per informazioni sull'endpoint VPC, consulta Creazione di endpoint VPC

nella Guida per l'utente di AWS Secrets Manager . Per ulteriori informazioni sull'utilizzo di Secrets Manager e Amazon VPC, consulta Come connettersi al servizio Secrets Manager all'interno di un Amazon VPC.

- Per le attività di Windows configurate per utilizzare il driver di registrazione `awslogs`, è necessario impostare anche la variabile di ambiente `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE` nell'istanza di container. Ciò può essere eseguito con Dati utente utilizzando la seguente sintassi:

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE",
 $TRUE, "Machine")
Initialize-ECSAgent -Cluster <cluster name> -EnableTaskIAMRole -LoggingDrivers
 '["json-file","awslogs"]'
</powershell>
```

Autorizzazioni IAM

Per utilizzare questa funzione, è necessario disporre del ruolo di esecuzione dei processi di Amazon ECS e fare riferimento allo stesso nella definizione di attività. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

Le autorizzazioni seguenti devono essere aggiunte manualmente come policy in linea al ruolo per l'esecuzione del processo, per fornire l'accesso ai segreti di Secrets Manager che crei. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

- `secretsmanager:GetSecretValue`: obbligatorio se si fa riferimento a un segreto di Secrets Manager. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.
- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave gestita dal cliente e non quella predefinita. L'ARN per la chiave gestita dal cliente dovrebbe essere aggiunto come risorsa. Aggiunge l'autorizzazione per decrittografare la chiave gestita dal cliente.

La seguente policy di esempio aggiunge le autorizzazioni necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
```

```
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name",
        "arn:aws:kms:region:aws_account_id:key/key_id"
      ]
    }
  ]
}
```

Creazione del segreto di AWS Secrets Manager

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni, consulta Creare un AWS Secrets Manager segreto nella Guida per l'AWS Secrets Manager utente.

Aggiunta della variabile di ambiente alla definizione di container

All'interno della definizione del container, è possibile specificare quanto segue:

- L'oggetto `secrets` contenente il nome della variabile di ambiente da impostare nel container
- L'Amazon Resource Name (ARN) del segreto di Secrets Manager
- Parametri aggiuntivi che contengono i dati sensibili da presentare al container

Nell'esempio seguente viene illustrata la sintassi completa che deve essere specificata per il segreto di Secrets Manager.

```
arn:aws:secretsmanager:region:aws_account_id:secret:secret-name:json-key:version-stage:version-id
```

Nella sezione seguente vengono descritti i parametri aggiuntivi. Questi parametri sono facoltativi, ma se non li si utilizza, è necessario includere i due punti `:` per utilizzare i valori predefiniti. Esempi sono forniti di seguito per un maggiore contesto.

`json-key`

Specificare il nome della chiave in una coppia chiave-valore con il valore che si desidera impostare come valore della variabile di ambiente. Sono supportati solo i valori in formato JSON. Se non si specifica una chiave JSON, viene utilizzato il contenuto completo del segreto.

`version-stage`

Specificare l'etichetta di gestione temporanea della versione di un segreto che si desidera utilizzare. Se viene specificata un'etichetta di gestione temporanea della versione, non è possibile specificare un ID versione. Se non viene specificata alcuna fase di versione, il comportamento predefinito consiste nel recuperare il segreto con l'etichetta `AWSCURRENT` di gestione temporanea.

Le etichette di gestione temporanea vengono utilizzate per tenere traccia di diverse versioni di un segreto quando vengono aggiornate o ruotate. Ogni versione di un segreto ha una o più etichette di gestione temporanea e un ID. Per ulteriori informazioni, consulta Termini e concetti chiave di AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager .

`version-id`

Specifica l'identificatore univoco della versione del segreto che intendi utilizzare. Se viene specificato un ID versione, non è possibile specificare un'etichetta di gestione temporanea della versione. Se non viene specificato alcun ID versione, il comportamento predefinito consiste nel recuperare il segreto con l'etichetta `AWSCURRENT` di gestione temporanea.

Gli ID di versione vengono utilizzati per tenere traccia di diverse versioni di un segreto quando vengono aggiornati o ruotati. Ogni versione di un segreto ha un ID. Per ulteriori informazioni, consulta Termini e concetti chiave di AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager .

Esempio di definizioni del container

Negli esempi seguenti vengono illustrati i modi in cui è possibile fare riferimento ai segreti di Secrets Manager nelle definizioni del container.

Example riferimento a un segreto completo

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un segreto di Secrets Manager.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
    }]
```

```
  }]
}
```

Per accedere al valore di questo segreto dall'interno del container, è necessario effettuare la chiamata a `$environment_variable_name`.

Example fare riferimento a una chiave specifica all'interno di un segreto

Di seguito viene illustrato un output di esempio da un comando get-secret-value che visualizza il contenuto di un segreto insieme all'etichetta di gestione temporanea della versione e all'ID della 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"
    ],
    "CreatedDate": 1581968848.921
}
```

Fare riferimento a una chiave specifica dell'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::"
    }]
  }]
}
```

Example riferimento a una versione segreta specifica

Di seguito viene illustrato un output di esempio da un comando describe-secret che visualizza il contenuto non crittografato di un segreto insieme ai metadati per tutte le versioni del segreto.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "Description": "Example of a secret containing application authorization data.",
    "RotationEnabled": false,
    "LastChangedDate": 1581968848.926,
    "LastAccessedDate": 1581897600.0,
    "Tags": [],
    "VersionIdsToStages": {
        "871d9eca-18aa-46a9-8785-981ddEXAMPLE": [
            "AWSCURRENT"
        ],
        "9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE": [
            "AWSPREVIOUS"
        ]
    }
}
```

Fare riferimento a un'etichetta di gestione temporanea della versione specifica dall'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf::AWSPREVIOUS:"
    }]
  }]
}
```

Fare riferimento a un ID di versione specifico dall'output precedente in una definizione di container specificando il nome della chiave alla fine dell'ARN.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:::9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE"
    }]
  }]
```

```
}
```

Example riferimento a una chiave specifica e un'etichetta di gestione temporanea della versione di un segreto

Di seguito viene illustrato come fare riferimento sia a una chiave specifica all'interno di un segreto che a una specifica etichetta di gestione temporanea della versione.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1:AWSPREVIOUS:"
    }]
  }]
}
```

Per specificare una chiave e un ID di versione specifici, utilizzare la seguente sintassi.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead3EXAMPLE"
    }]
  }]
}
```

Per informazioni su come creare una definizione di attività con il segreto specificato in una variabile di ambiente, vedereCreazione di una definizione di attività Amazon ECS utilizzando la console.

Recupera i parametri di Systems Manager tramite le variabili di ambiente Amazon ECS

Amazon ECS ti consente di iniettare dati sensibili nei tuoi contenitori memorizzando i dati sensibili nei AWS Systems Manager parametri di Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore.

Considerate quanto segue quando utilizzate una variabile di ambiente per iniettare un segreto di Systems Manager in un contenitore.

- I dati sensibili vengono inseriti nel container quando viene inizialmente avviato. Se il segreto o parametro viene in seguito aggiornato o ruotato, il container non riceverà automaticamente il valore aggiornato. È necessario avviare una nuova attività o se l'attività è parte di un servizio, è possibile aggiornare il servizio e utilizzare la nuova opzione Force new deployment (Forza nuova implementazione) per forzare il servizio ad avviare una nuova attività.
- Per le attività di Amazon ECS su AWS Fargate, è necessario considerare quanto segue:
  - Per inserire il contenuto completo di un segreto come variabile di ambiente o in una configurazione di log, è necessario utilizzare la versione della piattaforma `11.3.0` o successiva. Per informazioni, consulta Versioni della piattaforma Fargate Linux 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 Linux per Amazon ECS.
- Per i processi 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 del 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 Creazione di endpoint VPC nella Guida per l'utente di AWS Systems Manager .
- 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. Questo può essere fatto con Dati utente utilizzando la seguente sintassi:

```
<powershell>
[Environment]::SetEnvironmentVariable("ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE",
 $TRUE, "Machine")
Initialize-ECSAgent -Cluster <cluster name> -EnableTaskIAMRole -LoggingDrivers
 '["json-file","awslogs"]'
</powershell>
```

Autorizzazioni IAM

Per utilizzare questa funzione, è necessario disporre del ruolo di esecuzione dei processi di Amazon ECS e fare riferimento allo stesso nella definizione di attività. Ciò consente all'agente del container di recuperare le risorse di Systems Manager necessarie. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

> **Important**
>
> Per i processi che utilizzano il tipo di avvio EC2, devi 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 del Parameter Store di Systems Manager che create, aggiungete manualmente le seguenti autorizzazioni 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.

- `ssm:GetParameters`: obbligatorio quando si fa riferimento a un parametro di Archivio dei parametri Systems Manager in una definizione di attività. Aggiunge l'autorizzazione per recuperare i parametri di Systems Manager.
- `secretsmanager:GetSecretValue`: obbligatorio quando si fa riferimento a un segreto di Gestione dei segreti direttamente o se il parametro di Archivio dei parametri Systems Manager fa riferimento a un segreto di Gestione dei segreti in una definizione di attività. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.
- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave gestita personalizzata e non quella predefinita. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa. Aggiunge l'autorizzazione per decrittografare la chiave gestita dal cliente.

La seguente policy di esempio aggiunge le autorizzazioni necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:region:aws_account_id:parameter/parameter_name",
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name",
        "arn:aws:kms:region:aws_account_id:key/key_id"
      ]
    }
  ]
}
```

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 specifica `secrets` con il nome della variabile di ambiente per impostare il container e l'ARN del parametro dell'archivio parametri di Systems Manager contenente i dati sensibili da presentare al container. Per ulteriori informazioni, consulta `secrets`.

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un parametro dell'archivio parametri di Systems Manager. Se il parametro dell'archivio parametri di Systems Manager esiste nella stessa regione del processo in fase di avvio, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro si trova in una regione diversa, specifica l'ARN completo.

```
{
  "containerDefinitions": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
```

```
}
```

Per informazioni su come creare una definizione di attività con il segreto specificato in una variabile di ambiente, vedereCreazione di una definizione di attività Amazon ECS utilizzando la console.

Recupera i segreti per la configurazione dei log di Amazon ECS

È possibile utilizzare il `secretOptions` parametro in `logConfiguration` per trasmettere dati sensibili utilizzati per la registrazione.

È possibile memorizzare il segreto in Secrets Manager o Systems Manager.

Usa Secrets Manager

Nella definizione del container, quando specifichi `logConfiguration` è possibile specificare `secretOptions` con il nome dell'opzione del driver di log per impostare il container e l'ARN completo del segreto di Secrets Manager contenente i dati riservati da presentare al container.

Di seguito è riportato un frammento di una definizione di processo che mostra il formato quando si fa riferimento a un segreto Secrets Manager.

```
{
  "containerDefinitions": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://your_splunk_instance:8088"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}
```

Utilizzo di Systems Manager

Puoi inserire dati sensibili in una configurazione di log. Nella definizione del container, quando specifichi un `logConfiguration` è possibile specificare `secretOptions` con il nome dell'opzione

del driver di log per impostare il container e l'ARN completo del parametro dell'archivio parametri di Systems Manager contenente i dati sensibili da presentare al container.

> **Important**
>
> Se il parametro dell'archivio parametri di Systems Manager esiste nella stessa regione del processo in fase di avvio, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro si trova in una regione diversa, specifica l'ARN completo.

Di seguito è riportato un frammento di una definizione di attività che mostra il formato quando si fa riferimento a un parametro dell'archivio parametri di Systems Manager.

```
{
  "containerDefinitions": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter:/parameter_name"
      }]
    }]
  }]
}
```

Specificazione di dati sensibili utilizzando i segreti di Secrets Manager in Amazon ECS

Amazon ECS ti consente di iniettare dati sensibili nei tuoi contenitori archiviando i dati sensibili in modo AWS Secrets Manager segreto e quindi facendone riferimento nella definizione del contenitore. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

Scopri come creare un segreto di Secrets Manager, fare riferimento al segreto in una definizione di attività Amazon ECS e quindi verificarne il funzionamento interrogando la variabile di ambiente all'interno di un contenitore che mostra il contenuto del segreto.

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.
- Il tuo AWS utente dispone delle autorizzazioni IAM necessarie per creare le risorse Secrets Manager e Amazon ECS descritte.

Fase 1: creazione di un segreto di Gestione dei segreti

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 Tutorial: Creazione di un segreto di base nella Guida per l'utente di AWS Secrets Manager .

Le coppie chiave/valore da memorizzare in questo segreto rappresentano il valore della variabile di ambiente nel container alla fine del tutorial.

Salva l'ARN del segreto a cui fare riferimento nella policy IAM di esecuzione dell'attività e nella definizione dell'attività nelle fasi successive.

Fase 2: aggiornamento del ruolo IAM di esecuzione dell'attività

Per consentire ad Amazon ECS di recuperare i dati sensibili dal segreto di Secrets Manager, è necessario disporre del ruolo di esecuzione del processo Amazon ECS e fare riferimento a esso nella definizione di attività. Ciò consente all'agente del container di recuperare le risorse di Secrets Manager necessarie. Se non hai già creato il ruolo IAM di esecuzione del processo, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

Le seguenti fasi presuppongono che il ruolo IAM di esecuzione del processo sia già stato creato e correttamente configurato.

Come aggiornare il ruolo IAM di esecuzione del processo

Utilizza la console IAM per aggiornare il ruolo di esecuzione del processo con le autorizzazioni richieste.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, seleziona Ruoli.
3. Cercare l'elenco di ruoli per `ecsTaskExecutionRole` e selezionarlo.
4. Scegliere Permissions (Autorizzazioni), Add inline policy (Aggiungi policy inline).
5. Scegli la scheda JSON e specifica il seguente testo JSON, assicurandoti di specificare l'ARN completo del segreto di Secrets Manager creato nella fase 1.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [

 "arn:aws:secretsmanager:region:aws_account_id:secret:username_value"
            ]
        }
    ]
}
```

6. Scegli Verifica policy. Per Name (Nome) specificare `ECSSecretsTutorial`, quindi scegliere Create policy (Crea policy).

Fase 3: creazione di una definizione di attività Amazon ECS

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 all'indirizzo 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, inserisci il seguente testo JSON della definizione di attività, assicurandoti di specificare l'ARN completo del segreto di Secrets Manager creato nella fase 1 e il ruolo IAM dell'esecuzione del processo aggiornato nella fase 2. Selezionare 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": "httpd:2.4",
            "essential": true,
            "name": "ecs-secrets-container"
        }
    ],
    "family": "ecs-secrets-tutorial"
}
```

6. Scegli Crea.

Fase 4: creazione di un cluster Amazon ECS

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 sulla creazione di un cluster per il tipo di lancio EC2, consulta the section called “Creazione di un cluster per il tipo di lancio di Amazon EC2”.

Fase 5: esecuzione di un'attività di Amazon ECS

Puoi utilizzare la console Amazon ECS per eseguire un processo utilizzando la definizione di attività creata. Per questo tutorial verrà eseguito un processo utilizzando il tipo di avvio EC2, mediante il cluster creato nella fase 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
   b. Nel riquadro di navigazione, seleziona Cluster e quindi il cluster che hai creato.
   c. Seleziona Infrastruttura e quindi l'istanza di container.
   d. Registra l'IP pubblico o il DNS pubblico per l'istanza.
2. Se utilizzi un computer macOS o Linux, connettiti all'istanza con il comando seguente, sostituisci il percorso con la chiave privata e l'indirizzo pubblico dell'istanza:

   ```
   $ ssh -i /path/to/my-key-pair.pem ec2-user@ec2-198-51-100-1.compute-1.amazonaws.com
   ```

   Per ulteriori informazioni sull'utilizzo di un computer Windows, consulta Connessione all'istanza Linux da Windows tramite PuTTY nella Guida per l'utente di Amazon EC2.

> **Important**
>
> Per ulteriori informazioni su eventuali problemi di connessione alla tua istanza, consulta Risoluzione dei problemi di connessione alla tua istanza nella Guida per l'utente di Amazon EC2.

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: eliminare

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 all'indirizzo https://console.aws.amazon.com/ecs/v2.

2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Scegli Elimina Cluster.
5. Nella casella di conferma, immetti elimina ***`nome cluster`***, 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 ECS. SecretsTutorial Scegli Rimuovi.
10. Apri la console di Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
11. Selezionare il segreto username_value creato e scegliere Actions (Operazioni), Delete secret (Elimina segreto).

Parametri di definizione delle attività di Amazon ECS

Le definizioni delle attività sono suddivise in parti separate: la famiglia di attività, il ruolo dell'attività AWS Identity and Access Management (IAM), la modalità di rete, le definizioni dei contenitori, i volumi, i vincoli di posizionamento delle attività e i tipi di avvio. Le definizioni della famiglia e del container sono richieste in una definizione di attività. Al contrario, il ruolo dell'attività, la modalità di rete, i volumi, i vincoli di posizionamento delle attività e il tipo di avvio sono facoltativi.

Puoi utilizzare questi parametri in un file JSON per configurare la definizione del processo.

Di seguito sono riportate descrizioni più dettagliate per ogni parametro di definizione di attività.

Family

`family`

Tipo: stringa

Campo obbligatorio: sì

Quando registri una definizione di attività, le assegni una famiglia, ovvero una sorta di nome per più versioni della definizione di attività, specificando un numero di revisione. Alla prima definizione di attività registrata in una determinata famiglia viene assegnato il numero di revisione 1 e a

qualsiasi definizione di attività registrata successivamente viene assegnato un numero di revisione sequenziale.

Tipi di avvio

Quando registri una definizione di attività, puoi specificare un tipo di avvio che Amazon ECS deve convalidare per la definizione di attività. Se la definizione di attività non viene convalidata in base alle compatibilità specificate, viene restituita un'eccezione client. Per ulteriori informazioni, consulta Tipi di avvio di Amazon ECS.

Il parametro seguente è permesso in una definizione di attività.

`requiresCompatibilities`

Tipo: array di stringhe

Campo obbligatorio: no

Valori validi: `EC2` | `FARGATE` | `EXTERNAL`

Il tipo di avvio per il quale è stata convalidata la definizione di attività. In questo modo viene avviato un controllo per garantire che tutti i parametri utilizzati nella definizione di attività soddisfino i requisiti del tipo di avvio.

Ruolo del processo

`taskRoleArn`

-Tipo: stringa

Campo obbligatorio: no

Quando registri una definizione di attività, puoi specificare un ruolo di processo per un ruolo IAM che conceda ai container nel processo l'autorizzazione per chiamare le API AWS specificate nelle relative policy associate per tuo conto. 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 Windows di Amazon EC2.

Ruolo per l'esecuzione del processo

`executionRoleArn`

- Tipo: stringa

Obbligatorio: condizionale

L'Amazon Resource Name (ARN) del ruolo di esecuzione dell'attività che concede all'agente container Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto.

> **Note**
>
> Il ruolo IAM di esecuzione del processo è richiesto in base ai requisiti del processo. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

Modalità di rete

`networkMode`

- Tipo: stringa

Campo obbligatorio: no

La modalità di rete Docker da utilizzare per i container nel processo. Per le attività Amazon ECS ospitate su istanze Linux di Amazon EC2, i valori validi sono `none`, `bridge`, `awsvpc` e `host`. Se non viene specificata alcuna modalità di rete, la modalità di rete di default è `bridge`. Per i processi Amazon ECS ospitati su istanze Windows di Amazon EC2, i valori validi sono `default` e `awsvpc`. Se non viene specificata alcuna modalità di rete, viene utilizzata la modalità di rete `default`. Per le attività di Amazon ECS ospitate su Fargate, è richiesta `awsvpc` la modalità di rete.

Se la modalità di rete è impostata su `none`, i container dell'attività non dispongono di connettività esterna e non è possibile specificare le mappature delle porte nella definizione del container.

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 istanza Amazon EC2 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 ignora la rete virtuale integrata di Docker e mappa le porte del container direttamente all'interfaccia di rete elastica (ENI) dell'istanza Amazon EC2 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 sarà possibile eseguire più attività con la stessa definizione di attività su una singola istanza Amazon EC2.

> **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.

Per i tipi di avvio di Amazon EC2, se la modalità di rete è`awsvpc`, all'attività viene assegnata un'interfaccia di rete elastica ed è necessario specificare a `NetworkConfiguration` quando si crea un servizio o si esegue un'attività con la definizione dell'attività. Per ulteriori informazioni, consulta Opzioni di task networking di Amazon ECS per il tipo di lancio 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 istanza Amazon EC2 che ospita l'attività. La rete virtuale integrata su Windows utilizza il driver di rete `nat` Docker.

Per i tipi di avvio di Fargate, quando la modalità di rete è attiva`awsvpc`, all'operazione viene assegnata un'interfaccia di rete elastica ed è necessario specificare a `NetworkConfiguration` quando si crea un servizio o si esegue un'attività con la definizione dell'attività. Per ulteriori informazioni, vedere Fargate Task Networking. La modalità di rete `awsvpc` offre le massime prestazioni di rete per i container perché utilizza lo stack di rete di Amazon EC2. Le porte dei container esposte vengono mappate direttamente alla porta dell'interfaccia di rete elastica collegata. Pertanto, non puoi utilizzare le mappature delle porte host dinamiche.

Le modalità di rete `host` e `awsvpc` offrono le massime prestazioni di rete per i container perché utilizzano lo stack di rete di Amazon EC2. 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.

Con il tipo di avvio Fargate, la modalità di rete `awsvpc` è obbligatoria. Se utilizzi il tipo di avvio EC2, la modalità di rete consentita dipende dal sistema operativo dell'istanza EC2 sottostante. Con Linux, può essere utilizzata qualsiasi modalità di rete. Se Windows, possono essere utilizzate le modalità `default` e `awsvpc`.

Piattaforma di runtime

`operatingSystemFamily`

- Tipo: stringa

Obbligatorio: condizionale

Di default: LINUX

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 per le attività Amazon ECS ospitate su Fargate sono `LINUX`, `WINDOWS_SERVER_2019_FULL` `WINDOWS_SERVER_2019_CORE`, `WINDOWS_SERVER_2022_FULL` e `WINDOWS_SERVER_2022_CORE`.

I valori validi per i processi Amazon ECS ospitati su EC2 sono `LINUX`, `WINDOWS_SERVER_2022_CORE`, `WINDOWS_SERVER_2022_FULL`, `WINDOWS_SERVER_2019_FULL` e `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: stringa

Obbligatorio: condizionale

Valore di default: X86_64

Questo parametro è richiesto per i processi Amazon ECS ospitati su Fargate. 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 processi Linux per il tipo di avvio Fargate o per il tipo di avvio EC2, puoi impostare il valore su `ARM64`. Per ulteriori informazioni, consulta the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”.

Dimensioni processo

Quando registri una definizione di attività, puoi specificare la quantità totale di CPU e memoria utilizzata per l'attività. Questo valore è separato dai valori `cpu` e `memory` a livello di definizione del container. Per le attività ospitate su istanze Amazon EC2, questi campi sono facoltativi. Per le attività ospitate su Fargate (sia Linux sia Windows), questi campi sono obbligatori e sono supportati valori specifici per `cpu` e `memory`.

> **Note**
>
> I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows. Ti consigliamo di specificare risorse a livello di container per i container Windows.

Il parametro seguente è permesso in una definizione di attività:

`cpu`

- Tipo: stringa

Obbligatorio: condizionale

> **Note**
>
> Questo parametro non è supportato per i container Windows.

Il limite rigido di unità CPU da presentare per il processo. È possibile specificare i valori della CPU nel file JSON come stringa in unità CPU o CPU virtuali (vCPU). Ad esempio, è possibile specificare un valore CPU come `1024` nelle unità CPU o `1 vCPU` nelle vCPU. Quando la definizione di attività è registrata, un valore vCPU viene convertito in un numero intero che indica le unità CPU.

Per le attività in esecuzione su istanze Amazon EC2 o esterne, 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 per le attività eseguite su istanze EC2 o esterne sono compresi tra `0.125` vCPU e `10` vCPU.

Per le attività in esecuzione su Fargate (sia container Linux che Windows), 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

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
Note Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.		
16384 (16vCPU) Note Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

`memory`

-Tipo: stringa

Obbligatorio: condizionale

> **Note**
>
> Questo parametro non è supportato per i container Windows.

Il limite rigido di memoria da presentare all'attività. È possibile specificare i valori di memoria nella definizione dell'attività come stringa in mebibyte (MiB) o gigabyte (GB). Ad esempio, è possibile specificare un valore di memoria `3072` in MiB o `3 GB` in GB. Quando la definizione di attività è registrata, un valore GB viene convertito in un numero intero che indica il MiB.

Per le attività ospitate su istanze Amazon EC2, questo campo è facoltativo e può essere utilizzato qualsiasi valore. Se viene specificato un valore di memoria a livello di attività, il valore di memoria

a livello di container è facoltativo. Se il cluster non dispone di istanze di container registrate con la memoria richiesta disponibile, l'attività non va a buon fine. Puoi ottimizzare l'utilizzo delle risorse assegnando alle attività quanta più memoria possibile per un determinato tipo di istanza. Per ulteriori informazioni, consulta Riservare la memoria delle istanze del contenitore Amazon ECS Linux .

Per i processi ospitati su Fargate (sia container Linux che Windows), questo campo è obbligatorio e devi 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
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	8192 (8 vCPU)	Linux

Valore di memoria (in MiB, con valore equivalente approssimativo in GB)	Valore CPU	Sistemi operativi supportati per Fargate
Note Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.		
Tra 32 GB e 120 GB in incrementi di 8 GB **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	16384 (16vCPU)	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

I seguenti parametri di definizione di attività sono obbligatori o utilizzati nella maggior parte delle definizioni del container.

Argomenti

Nome

`name`

Tipo: stringa

Campo obbligatorio: sì

Il nome di un container. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. Se colleghi più container in una definizione di attività, il parametro `name` di uno dei container può essere inserito nel parametro `links` di un altro container. Questo per collegare i container.

Immagine

`image`

Tipo: stringa

Campo obbligatorio: sì

L'immagine utilizzata per avviare un container. Questa stringa viene 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 a `Image` nella sezione Crea un container dell'API remota Docker e al parametro `IMAGE` di docker run.

- Quando viene avviato un nuovo processo, l'agente del container Amazon ECS esegue il pull della versione più recente dell'immagine e del tag specificati per il container da utilizzare. Tuttavia, gli aggiornamenti successivi a un'immagine del repository non vengono propagate alle attività già in esecuzione.
- 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`).

Memoria

`memory`

Tipo: integer

Campo obbligatorio: no

La quantità (in MiB) della memoria da presentare al container. Se il container tenta di superare la memoria specificata qui, viene terminato. La quantità totale di memoria prenotata per tutti i container all'interno di un processo deve essere inferiore al valore `memory` del processo, se specificato. Questo parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run.

Se utilizzi il tipo di avvio Fargate, questo parametro è facoltativo.

Se utilizzi il tipo di avvio EC2, devi specificare un valore di memoria a livello di attività o un valore di memoria 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 Riservare la memoria delle istanze del contenitore Amazon ECS Linux .

`memoryReservation`

Tipo: integer

Campo obbligatorio: no

Il limite flessibile (in MiB) della memoria da prenotare per il container. Quando la memoria di sistema è in conflitto, Docker tenta di 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 sezione Create a container di Docker Remote API e l'opzione `--memory-reservation` a docker run.

Se non viene specificato un valore di memoria a livello di attività, è necessario specificare un numero intero diverso da zero per uno o entrambi i codici `memory` o `memoryReservation` in una definizione del container. Se specifichi entrambe, `memory` deve essere superiore a `memoryReservation`. Se specifichi `memoryReservation`, tale valore viene sottratto dalle risorse di memoria disponibili per l'istanza di container in cui il container viene posizionato. in caso contrario, viene utilizzato il valore `memory`.

Ad esempio, supponiamo che il container normalmente utilizza 128 MiB di memoria, ma con picchi occasionali di 256 MiB di memoria per brevi periodi di tempo. Puoi impostare un valore di `memoryReservation` di 128 MiB e un limite rigido `memory` di 300 MiB. Questa configurazione consente al container di riservare 128 MiB di memoria solo dalle risorse restanti nell'istanza di container. Allo stesso tempo, questa configurazione consente al container di utilizzare più risorse di memoria quando necessario.

> **Note**
>
> Questo parametro non è supportato per i container Windows.

Il daemon Docker 20.10.0 o versione successiva prenota un minimo di 6 MiB di memoria per un container. Pertanto, non specificare meno di 6 MiB di memoria per i container.

Il daemon Docker 19.03.13-ce o versione precedente prenota un minimo di 4 MiB di memoria per un container. Pertanto, non specificare meno di 4 MiB di memoria per i container.

> **Note**
>
> Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta Riservare la memoria delle istanze del contenitore Amazon ECS Linux .

Mappature di porte

`portMappings`

Tipo: array di oggetti

Campo obbligatorio: no

La mappatura delle porte consente ai container di accedere alle porte nell'istanza di container dell'host per inviare o ricevere traffico.

Per le definizioni di attività che utilizzano la modalità di rete `awsvpc`, specifica solo il parametro `containerPort`. Il valore di `hostPort` può essere lasciato vuoto o deve essere lo stesso valore di `containerPort`.

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`, `protocol`) è associata a `PortBindings` nella sezione Creazione di un container dell'API Docker Remote e l'opzione `--publish` a 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

Campo obbligatorio: no

Il protocollo dell'applicazione utilizzato per la mappatura delle porte. Questo parametro si applica solo a Service Connect. Ti consigliamo di impostare questo parametro in maniera coerente con il protocollo utilizzato dall'applicazione. Se imposti questo parametro, Amazon ECS aggiunge la gestione delle connessioni specifica del protocollo al proxy Service Connect. Se imposti questo parametro, Amazon ECS aggiunge la telemetria specifica del protocollo nella console Amazon ECS e. CloudWatch

Se non imposti un valore per questo parametro, viene utilizzato TCP. Amazon ECS, tuttavia, non aggiunge la telemetria specifica del protocollo TCP.

Per ulteriori informazioni, consulta the section called “ Service Connect”.

Valori di protocollo validi: `"HTTP" | "HTTP2" | "GRPC"`

`containerPort`

Tipo: integer

Obbligatorio: sì, quando si utilizzano `portMappings`

Il numero di porta nel container associato alla porta dell'host definito dall'utente o assegnata automaticamente.

Se utilizzi container in un'attività con il tipo di avvio Fargate, le porte esposte devono essere specificate utilizzando `containerPort`.

Per i container Windows su Fargate, non è possibile utilizzare la porta 3150 per `containerPort`. Questo perché la porta è riservata.

Supponiamo di utilizzare i container in un'attività con il tipo di avvio EC2 e di specificare una porta di container ma non una porta dell'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

Campo obbligatorio: no

L'intervallo dei numeri di porta nel container associato all'intervallo di porte host mappato in maniera dinamica.

È possibile impostare questo parametro solo utilizzando l'API `register-task-definition`. L'opzione è disponibile nel parametro `portMappings`. Per ulteriori informazioni, consulta register-task-definition nella Documentazione di riferimento sull'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 i tipi di avvio EC2 e AWS Fargate.

- Questo parametro è disponibile per sistemi operativi sia Lunix 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, 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

Campo 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

Campo obbligatorio: no

Il numero di porta nell'istanza di container per prenotare per il container.

Se usi container in un'attività con il tipo di avvio Fargate, `hostPort` può essere lasciato vuoto oppure può avere lo stesso valore di `containerPort`.

Supponiamo di utilizzare i container in un'attività con il tipo di avvio EC2. 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.

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: stringa

Obbligatorio: no, necessario per configurare Service Connect in un servizio

Il nome utilizzato per la mappatura delle porte. Questo parametro si applica solo a Service Connect. Questo parametro è il nome utilizzato nella configurazione di Service Connect 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.

```
"portMappings": [
    {
        "name": string,
        "containerPort": integer
    }
]
```

`protocol`

-Tipo: stringa

Campo obbligatorio: no

Il protocollo utilizzato per la mappatura delle porte. I valori validi sono `tcp` e `udp`. Il valore predefinito è `tcp`.

> **Important**
>
> È supportato solo `tcp` per Service Connect. Ricorda che `tcp` è implicito se questo campo non è impostato.

> **Important**
>
> Il supporto UDP è disponibile solo nelle istanze di container che sono state avviate con la versione 1.2.0 dell'agente del container Amazon ECS (ad esempio l'AMI `amzn-ami-2015.03.c-amazon-ecs-optimized`) o successiva o con agenti del container

che sono stati aggiornati alla versione 1.3.0 o successiva. Per passare all'ultima versione dell'agente del container, consulta Aggiornamento dell'agente del container Amazon ECS.

Se specifichi una porta dell'host, utilizza la seguente sintassi.

```
"portMappings": [
    {
        "containerPort": integer,
        "hostPort": integer
    }
    ...
]
```

Se desideri una porta dell'host assegnata automaticamente, utilizza la seguente sintassi.

```
"portMappings": [
    {
        "containerPort": integer
    }
    ...
]
```

Credenziali del repository privato

`repositoryCredentials`

Tipo: oggetto RepositoryCredentials

Campo obbligatorio: no

Le credenziali dell'archivio per l'autenticazione di un registro privato.

Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.

`credentialsParameter`

- Tipo: stringa

Obbligatorio: sì, quando si utilizzano `repositoryCredentials`

L'Amazon Resource Name (ARN) del segreto contenente le credenziali dell'archivio privato.

Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.

> **Note**
>
> Quando utilizzi l'API o gli AWS SDK di Amazon ECS, se il segreto esiste nella stessa regione dell'attività che stai avviando, puoi utilizzare l'ARN completo o il nome del segreto. AWS CLI Quando si utilizza il AWS Management Console, è necessario specificare l'ARN completo del segreto.

Di seguito viene riportato un frammento di una definizione di attività che mostra i parametri obbligatori:

```
"containerDefinitions": [
    {
        "image": "private-repo/private-image",
        "repositoryCredentials": {
            "credentialsParameter":
 "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"
        }
    }
]
```

Parametri avanzati di definizione del container

I seguenti parametri avanzati di definizione del container forniscono funzionalità estese al comando docker run che viene utilizzato per lanciare container nelle istanze di container di Amazon ECS.

Argomenti

- Controllo dello stato
- Ambiente
- Impostazioni di rete
- Archiviazione e registrazione

- Sicurezza
- Limiti delle risorse
- Etichette Docker

Controllo dello stato

`healthCheck`

Comando di controllo dell'integrità del container e parametri di configurazione associati per il container. Per ulteriori informazioni, consulta Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container.

`command`

Matrice di stringhe che rappresenta il comando eseguito dal container per determinare l'integrità. La matrice di stringhe può iniziare con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell predefinita del container. Se non è specificato nessuno dei due, viene utilizzato `CMD`.

Quando si registra una definizione di attività in AWS Management Console, utilizzare un elenco di comandi separati da virgole. Questi comandi vengono convertiti in stringa dopo la creazione della definizione delle attività. Di seguito è riportato un esempio di input per il controllo dell'integrità.

```
CMD-SHELL, curl -f http://localhost/ || exit 1
```

Quando registrate una definizione di attività utilizzando il pannello AWS Management Console JSON, o le API AWS CLI, racchiudete 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. Per ulteriori informazioni consulta, `HealthCheck` nella sezione Create a container di Docker Remote API.

`interval`

Intervallo di tempo (in secondi) tra ogni controllo dell'integrità. Puoi specificare un valore compreso tra 5 e 300 secondi. Il valore di predefinito è 30 secondi.

`timeout`

Periodo di tempo in secondi per cui attendere che un controllo dell'integrità venga superato prima di considerarlo un errore. Puoi specificare un valore compreso tra 2 e 60 secondi. Il valore di default è 5 secondi.

`retries`

Numero di tentativi per cui riprovare un controllo dello stato non riuscito prima che il container venga considerato non integro. Puoi specificare un valore compreso tra 1 e 10 tentativi. Il valore predefinito è tre tentativi.

`startPeriod`

Periodo di tolleranza facoltativo entro il quale concedere ai container il tempo necessario per il bootstrap prima che i controlli dell'integrità non riusciti vengano conteggiati rispetto al numero massimo di nuovi tentativi. Puoi specificare un valore compreso tra 0 e 300 secondi. Per impostazione predefinita, `startPeriod` è disabilitato.

Ambiente

`cpu`

Tipo: integer

Campo 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 (Creazione di un container) di Docker Remote API e l'opzione `--cpu-shares` a docker run.

Questo campo è facoltativo solo per le attività che utilizzano il tipo di avvio Fargate. La quantità totale di CPU riservata per tutti i container all'interno di un'attività deve essere inferiore al valore `cpu` a livello di attività.

> **Note**
>
> Puoi determinare il numero di unità CPU disponibili per ciascun tipo di istanza Amazon EC2. A tale scopo, moltiplica il numero di vCPU elencate per il tipo di istanza nella pagina di dettaglio delle istanze Amazon EC2 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. 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 minori di 2 (incluso un valore null), il comportamento varia in base alla versione dell'agente del container Amazon ECS:

- Versioni dell'agente <= 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 >= 1.2.0: i valori di CPU null, zero e 1 vengono trasmessi a Docker come due quote di 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

Campo obbligatorio: no

Il numero di `GPUs` fisiche che l'agente del container Amazon ECS riserva per il container. Il numero di GPU riservate per tutti i container in un'attività non deve superare il numero di GPU

disponibili nell'istanza di container in 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 o per i container ospitati su Fargate.

`Elastic Inference accelerator`

Tipo: oggetto ResourceRequirement

Campo obbligatorio: no

Per il tipo `InferenceAccelerator`, il `value` corrisponde a `deviceName` per un `InferenceAccelerator` specificato in una definizione di attività. Per ulteriori informazioni, consulta the section called “Nome dell'acceleratore di inferenza elastica”.

> **Note**
>
> A partire dal 15 aprile 2023, non AWS effettuerà l'onboarding di nuovi clienti in Amazon Elastic Inference (EI) e aiuterà i clienti attuali a migrare i propri carichi di lavoro verso opzioni che offrono prezzi e prestazioni migliori. Dopo il 15 aprile 2023, i nuovi clienti non saranno in grado di avviare istanze con acceleratori Amazon EI su Amazon, SageMaker Amazon ECS o Amazon EC2. Tuttavia, i clienti che hanno utilizzato Amazon EI almeno una volta negli ultimi 30 giorni sono considerati clienti attuali e potranno continuare a usufruire del servizio.

> **Note**
>
> Questo parametro non è supportato per i container Windows o per i container ospitati su Fargate.

`essential`

Tipo: Booleano

Campo obbligatorio: no

Si supponga che il parametro `essential` di un container sia contrassegnato come `true` e che tale container abbia esito negativo o si arresti per qualsiasi motivo. Di conseguenza, tutti gli altri container che fanno parte dell'attività vengono arrestati. Se il parametro `essential` di un container è contrassegnato come `false`, il suo esito negativo non influenza il resto dei container in un'attività. Se questo parametro viene omesso, un container si considera essenziale.

Tutti i processi devono avere almeno un container essenziale. Supponiamo di avere un'applicazione composta da più container. In questo caso, si raggruppano i container utilizzati per uno scopo comune in componenti e si separano i diversi componenti in più definizioni delle attività. Per ulteriori informazioni, consulta Progetta la tua applicazione per Amazon ECS.

```
"essential": true|false
```

`entryPoint`

> **Important**
>
> Le versioni precedenti dell'agente del container Amazon ECS non gestiscono correttamente i parametri `entryPoint`. In caso di problemi durante l'utilizzo `entryPoint`, aggiorna l'agente del container o inserisci i comandi e gli argomenti come elementi di matrice `command`.

Tipo: array di stringhe

Campo obbligatorio: no

Il punto di ingresso che viene trasmesso al container. Questo parametro è mappato a `Entrypoint` nella sezione Create a container di Docker Remote API e l'opzione `--entrypoint` a docker run. Per ulteriori informazioni sul parametro Docker `ENTRYPOINT`, consulta https://docs.docker.com/engine/reference/builder/#entrypoint.

```
"entryPoint": ["string", ...]
```

`command`

Tipo: array di stringhe

Campo obbligatorio: no

Il comando che viene inviato al container. Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd. In caso di più argomenti, ognuno di questi deve essere una stringa separata nella matrice.

```
"command": ["string", ...]
```

`workingDirectory`

-Tipo: stringa

Campo obbligatorio: no

La directory di lavoro nel container in cui eseguire i comandi. Questo parametro è mappato a `WorkingDir` nella sezione Create a container di Docker Remote API e l'opzione `--workdir` a docker run.

```
"workingDirectory": "string"
```

`environmentFiles`

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di file contenenti le variabili di ambiente da passare a un container. Questo parametro è mappato all'opzione `--env-file` su docker run

Questo non è disponibile per i Windows contenitori e i contenitori Windows su Fargate

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. Per ulteriori informazioni sulla sintassi appropriata del file delle variabili di ambiente, consulta Declare default environment variables in file (Dichiarare le variabili d'ambiente predefinite in un file).

Se nella definizione del container sono specificate singole variabili di ambiente, hanno la precedenza sulle variabili contenute in un file di ambiente. Se vengono specificati più file di ambiente che contengono la stessa variabile, vengono elaborati dall'alto verso il basso.

Consigliamo di utilizzare nomi di variabili univoci. Per ulteriori informazioni, consulta Passa una singola variabile di ambiente a un contenitore Amazon ECS.

`value`

Tipo: stringa

Campo obbligatorio: sì

L'Amazon Resource Name (ARN) dell'oggetto Amazon S3 contenente il file della variabile di ambiente.

`type`

Tipo: stringa

Campo obbligatorio: sì

Il tipo di file da utilizzare L'unico valore supportato è `s3`.

`environment`

Tipo: array di oggetti

Campo obbligatorio: no

Le variabili di ambiente da passare a un container. Questo parametro è mappato a `Env` nella sezione Create a container di Docker Remote API e l'opzione `--env` a docker run.

> **Important**
>
> Non è consigliabile utilizzare variabili di ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

`name`

-Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il nome della variabile di ambiente.

`value`

-Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `environment`

Il valore della variabile di ambiente.

```
"environment" : [
    { "name" : "string", "value" : "string" },
    { "name" : "string", "value" : "string" }
]
```

`secrets`

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da esporre al container. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

`name`

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

`valueFrom`

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del AWS Secrets Manager segreto o l'ARN completo del parametro nel Parameter Store. AWS Systems Manager

> **Note**
>
> Se il parametro Systems Manager Parameter Store o il parametro Secrets Manager esiste nella Regione AWS stessa operazione che si sta avviando, è possibile utilizzare l'ARN completo o il nome del segreto. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

```
"secrets": [
    {
        "name": "environment_variable_name",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }
]
```

Impostazioni di rete

`disableNetworking`

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è true, le reti sono disabilitate all'interno del container. Questo parametro è mappato in `NetworkDisabled` nella sezione Create a container di Docker Remote API.

> **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

Campo obbligatorio: no

Il parametro `link` consente ai container di comunicare tra loro senza la necessità di mappatura delle porte. Questo parametro è supportato solo se la modalità di rete di una definizione delle attività è impostata su `bridge`. Il 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 caratteri di sottolineatura. Per ulteriori informazioni sul collegamento dei

container Docker, consulta https://docs.docker.com/engine/userguide/networking/default_network/dockerlinks/. Questo parametro è mappato a `Links` nella sezione Create a container di Docker Remote API e l'opzione `--link` a docker run.

> **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

Campo obbligatorio: no

Il nome host da utilizzare per il container. Questo parametro è mappato a `Hostname` nella sezione Create a container di Docker Remote API e l'opzione `--hostname` a docker run.

> **Note**
>
> Se utilizzi la modalità di rete `awsvpc`, il parametro `hostname` non è supportato.

```
"hostname": "string"
```

`dnsServers`

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di server DNS presentato al container. Questo parametro è mappato a `Dns` nella sezione Create a container di Docker Remote API e l'opzione `--dns` a docker run.

> **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

Campo obbligatorio: no

Modello: ^[a-zA-Z0-9-.]{0,253}[a-zA-Z0-9]$

Un elenco di domini di ricerca DNS presentato al container. Questo parametro è mappato a `DnsSearch` nella sezione Create a container di Docker Remote API e l'opzione `--dns-search` a docker run.

> **Note**
>
> Questo parametro non è supportato per i container o le attività di Windows che utilizzano la modalità di rete `awsvpc`.

```
"dnsSearchDomains": ["string", ...]
```

`extraHosts`

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di nomi host e mappature di indirizzi IP da aggiungere al file `/etc/hosts` nel container.

Questo parametro è mappato a `ExtraHosts` nella sezione Create a container di Docker Remote API e l'opzione `--add-host` a docker run.

> **Note**
>
> Questo parametro non è supportato per i container o le attività di Windows che utilizzano la modalità di rete `awsvpc`.

```
"extraHosts": [
      {
        "hostname": "string",
        "ipAddress": "string"
      }
      ...
    ]
```

`hostname`

- Tipo: stringa

Obbligatorio: sì, quando si utilizzano `extraHosts`

Il nome host da utilizzare nella voce `/etc/hosts`.

`ipAddress`

- Tipo: stringa

Obbligatorio: sì, quando si utilizzano `extraHosts`

L'indirizzo IP da utilizzare nella voce `/etc/hosts`.

Archiviazione e registrazione

`readonlyRootFilesystem`

Tipo: Booleano

Campo obbligatorio: no

Se il parametro è true, al container viene assegnato l'accesso in sola lettura al file system radice. Questo parametro è mappato a `ReadonlyRootfs` nella sezione Create a container di Docker Remote API e l'opzione `--read-only` a docker run.

> **Note**
>
> Questo parametro non è supportato per i container Windows.

Il valore predefinito è `false`.

```
"readonlyRootFilesystem": true|false
```

`mountPoints`

Tipo: array di oggetti

Campo obbligatorio: no

I punti di montaggio per i volumi di dati nel contenitore. Questo parametro è mappato a `Volumes` nella sezione Create a container di Docker Remote API e l'opzione `--volume` a docker run.

I container Windows possono montare intere directory sulla stessa unità di `$env:ProgramData`. I contenitori Windows non possono montare le directory su un'unità diversa e i punti di montaggio non possono essere utilizzati su più unità. È necessario specificare i punti di montaggio per collegare un volume Amazon EBS direttamente a un'attività Amazon ECS.

`sourceVolume`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il nome del volume da montare.

`containerPath`

- Tipo: stringa

  Obbligatorio: sì, quando si utilizzano `mountPoints`

  Il percorso nel contenitore in cui verrà montato il volume.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

`volumesFrom`

Tipo: array di oggetti

Campo obbligatorio: no

I volumi di dati da montare da un altro container. Questo parametro è mappato a `VolumesFrom` nella sezione Create a container di Docker Remote API e l'opzione `--volumes-from` a docker run.

`sourceContainer`

-Tipo: stringa

Obbligatorio: sì, quando viene utilizzato `volumesFrom`

Il nome del container da cui montare volumi.

`readOnly`

Tipo: Booleano

Campo obbligatorio: no

Se il valore è `true`, il container avrà accesso in sola lettura al volume. Se il valore è `false`, il container avrà accesso in scrittura al volume. Il valore predefinito è `false`.

```
"volumesFrom": [
                {
                  "sourceContainer": "string",
                  "readOnly": true|false
                }
              ]
```

`logConfiguration`

Tipo: LogConfigurationoggetto

Campo obbligatorio: no

La specifica di configurazione dei log per il container.

Per definizioni dell'attività di esempio che utilizzano una configurazione di log, consulta Esempi di definizioni di attività Amazon ECS.

Questo parametro è mappato a `LogConfig` nella sezione Create a container di Docker Remote API e l'opzione `--log-driver` a `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). Per ulteriori informazioni sulle opzioni per diversi driver di log supportati, consulta Configurazione dei driver di log nella documentazione di Docker.

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. Ulteriori driver di log potranno essere disponibili nei rilasci futuri dell'agente del container Amazon ECS.
- Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.
- Per le attività che utilizzano il tipo di avvio EC2, l'agente del container Amazon ECS in esecuzione in un'istanza di container deve registrare i driver di log disponibili in quell'istanza con la variabile di ambiente `ECS_AVAILABLE_LOGGING_DRIVERS`, prima che i container posizionati su tale istanza possano utilizzare queste opzioni di configurazione di log. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.
- Per le attività che utilizzano il tipo di avvio Fargate, è necessario installare qualsiasi software aggiuntivo al di fuori dell'operazione. 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","logentries","splunk","syslog","awsfirelens",
      "options": {"string": "string"
        ...},
 "secretOptions": [{
  "name": "string",
  "valueFrom": "string"
```

```
 }]
}
```

`logDriver`

- Tipo: stringa

Valori validi: `"awslogs","fluentd","gelf","json-file","journald","logentries","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.

Per le attività che utilizzano il tipo di avvio Fargate, i driver di log supportati sono `awslogs`, `splunk` e `awsfirelens`.

Per le attività che utilizzano il tipo di avvio EC2, i driver di log supportati sono `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `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

Campo obbligatorio: no

La mappa chiave/valore delle opzioni di configurazione per inviare il driver di log.

Quando lo utilizzi FireLens per indirizzare i log verso una AWS Partner Network destinazione Servizio AWS o per l'archiviazione e l'analisi dei log, puoi impostare `log-driver-buffer-limit` l'opzione per limitare il numero di eventi che vengono memorizzati nel buffer in memoria, prima di essere inviati al contenitore del log router. Può aiutarti a risolvere potenziali problemi di perdita di log perché una velocità di trasmissione effettiva elevata 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”.

Questo parametro richiede la versione 1.19 o successiva di Docker Remote API sull'istanza di container.

`secretOptions`

Tipo: array di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il segreto da inviare alla configurazione di log. I segreti utilizzati nella configurazione di log possono includere un token di autenticazione, un certificato o una chiave di crittografia. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

`name`

Tipo: stringa

Campo obbligatorio: sì

Il valore da impostare come variabile di ambiente sul container.

`valueFrom`

Tipo: stringa

Campo obbligatorio: sì

Il segreto da esporre alla configurazione di log del container.

```
"logConfiguration": {
 "logDriver": "splunk",
 "options": {
  "splunk-url": "https://cloud.splunk.com:8080",
  "splunk-token": "...",
  "tag": "...",
  ...
 },
 "secretOptions": [{
  "name": "splunk-token",
  "valueFrom": "/ecs/logconfig/splunkcred"
 }]
}
```

`firelensConfiguration`

Tipo: oggetto FirelensConfiguration

Campo obbligatorio: no

La FireLens configurazione per il contenitore. Si utilizza per specificare e configurare un router di log per i log del container. Per ulteriori informazioni, consulta Inviare i log di Amazon ECS a un servizio o AWSAWS Partner.

```
{
    "firelensConfiguration": {
        "type": "fluentd",
        "options": {
            "KeyName": ""
        }
    }
}
```

`options`

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

La mappa chiave/valore delle opzioni da utilizzare durante la configurazione del router di log. Questo campo è facoltativo e può essere utilizzato per aggiungere ulteriori metadati, ad esempio il processo, la definizione di attività, il cluster e i dettagli dell'istanza di container all'evento di log. Se specificato, la sintassi da utilizzare è `"options":{"enable-ecs-`

`log-metadata":"true|false","config-file-type:"s3|file","config-file-value":"arn:aws:s3:::mybucket/fluent.conf|filepath"}`. Per ulteriori informazioni, consulta Esempio di definizione di attività Amazon ECS: indirizza i log verso FireLens.

`type`

Tipo: stringa

Campo obbligatorio: sì

Il router di log da utilizzare. I valori validi sono `fluentd` o `fluentbit`.

Sicurezza

Per ulteriori informazioni sulla sicurezza di container, consulta Sicurezza delle attività e dei container nella Guida alle best practice di Amazon ECS.

`credentialSpecs`

Tipo: array di stringhe

Campo obbligatorio: no

Un elenco di ARN in SSM o Amazon S3 per un file di specifica delle credenziali (`CredSpec`) che configura il container per l'autenticazione Active Directory. Consigliamo di utilizzare questo parametro anziché `dockerSecurityOptions`. Il numero massimo di ARN è 1.

Esistono due formati per ogni ARN.

credentialspecdomainless:MyARN

Utilizza `credentialspecdomainless:MyARN` per fornire un file `CredSpec` con una sezione aggiuntiva per un segreto in Secrets Manager. Specifica le credenziali di accesso al dominio nel campo segreto.

Ogni attività eseguita su qualsiasi istanza di container può aggiungere domini diversi.

Puoi utilizzare questo formato senza aggiungere l'istanza di container a un dominio.

credentialspec:MyARN

Utilizza `credentialspec:MyARN` per fornire un nome a un file `CredSpec` per un singolo dominio.

Devi aggiungere l'istanza di container al dominio prima di iniziare qualsiasi attività che utilizzi questa definizione delle attività.

In entrambi i formati, sostituisci `MyARN` con l'ARN in SSM o Amazon S3.

Il file `credspec` deve fornire un ARN in Secrets Manager per un segreto contenente il nome utente, la password e il dominio a cui collegarsi. Per una maggiore sicurezza, l'istanza non viene aggiunta al dominio per l'autenticazione senza dominio. Le altre applicazioni sull'istanza non possono utilizzare le credenziali senza dominio. Puoi utilizzare questo parametro per eseguire attività sulla stessa istanza, anche se le attività devono aggiungere domini diversi. Per ulteriori informazioni, consulta Utilizzo di gMSA per i container Windows e Utilizzo di gMSA per i container Linux.

`privileged`

Tipo: Booleano

Campo obbligatorio: no

Se il parametro è true, al container vengono assegnati privilegi elevati nell'istanza di container host (simile all'utente `root`). Si consiglia di non far funzionare i container con `privileged`. Nella maggior parte dei casi, puoi specificare i privilegi esatti necessari utilizzando i parametri specifici anziché `privileged`.

Questo parametro è mappato a `Privileged` nella sezione Create a container di Docker Remote API e l'opzione `--privileged` a 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

Campo obbligatorio: no

L'utente da usare all'interno del container. Questo parametro è mappato a `User` nella sezione Create a container di Docker Remote API e l'opzione `--user` a 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

Campo obbligatorio: no

Un elenco di stringhe per fornire una configurazione personalizzata per più sistemi di sicurezza. Per ulteriori informazioni sui valori validi, consulta l'argomento relativo alla configurazione della sicurezza dell'esecuzione di Docker. Questo campo non è valido per container in attività che utilizzano il tipo di avvio Fargate.

Per le attività Linux su EC2, questo parametro può essere utilizzato per fare riferimento a etichette personalizzate per i sistemi di sicurezza su più livelli SELinux e AppArmor .

Per le attività su EC2, 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, consulta Scopri come usare GMSAS per contenitori EC2 Windows per Amazon ECS e Utilizzo gMSA per Linux contenitori EC2 su Amazon ECS.

Questo parametro è mappato a `SecurityOpt` nella sezione Create a container di Docker Remote API e l'opzione `--security-opt` a docker.

```
"dockerSecurityOptions": ["string", ...]
```

> **Note**
>
> L'agente del container Amazon ECS che viene eseguito su un'istanza di container deve registrarsi con le variabili di ambiente `ECS_SELINUX_CAPABLE=true` o `ECS_APPARMOR_CAPABLE=true` prima che i container posizionati su tale istanza possano utilizzare queste opzioni di sicurezza. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Limiti delle risorse

`ulimits`

Tipo: array di oggetti

Campo obbligatorio: no

Un elenco di valori `ulimit` da definire per un container. Questo valore sovrascrive l'impostazione della quota di risorse predefinite per il sistema operativo. Questo parametro è mappato a `Ulimits` nella sezione Create a container di Docker Remote API e l'opzione `--ulimit` a 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 `1024` 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"|"rtp
        "softLimit": integer,
        "hardLimit": integer
      }
      ...
    ]
```

`name`

- Tipo: stringa

  Valori validi: `"core" | "cpu" | "data" | "fsize" | "locks" | "memlock" | "msgqueue" | "nice" | "nofile" | "nproc" | "rss" | "rtprio" | "rttime" | "sigpending" | "stack"`

  Obbligatorio: sì, quando si utilizzano `ulimits`

  `type` di `ulimit`.

`hardLimit`

  Tipo: integer

Obbligatorio: sì, quando si utilizzano `ulimits`

Il limite rigido per il tipo `ulimit`.

`softLimit`

Tipo: integer

Obbligatorio: sì, quando si utilizzano `ulimits`

Il limite flessibile per il tipo `ulimit`.

Etichette Docker

`dockerLabels`

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Una mappa chiave/valore di etichette da aggiungere al container. Questo parametro è mappato a `Labels` nella sezione Create a container di Docker Remote API e l'opzione `--label` a docker run.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container.

```
"dockerLabels": {"string": "string"
      ...}
```

Altri parametri di definizione del container

I seguenti parametri di definizione del container possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS utilizzando l'opzione Configure via JSON (Configura tramite JSON). Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

Argomenti

- Parametri Linux

- Dipendenze per i container
- Timeout del container
- Controlli di sistema
- Interactive
- Pseudoterminale

Parametri Linux

`linuxParameters`

Tipo: oggetto LinuxParameters

Campo obbligatorio: no

Linux-opzioni specifiche che vengono applicate al contenitore, ad esempio. KernelCapabilities

> **Note**
>
> Questo parametro non è supportato per i container Windows.

```
"linuxParameters": {
      "capabilities": {
        "add": ["string", ...],
        "drop": ["string", ...]
        }
      }
```

`capabilities`

Tipo: oggetto KernelCapabilities

Campo obbligatorio: no

Le funzionalità di Linux per il container che vengono aggiunte o eliminate dalla configurazione predefinita fornita da Docker. Per ulteriori informazioni sulle funzionalità predefinite e quelle non predefinite disponibili, consulta Privilegi di runtime e funzionalità di Linux nei Riferimenti per l'esecuzione di 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 sezione Crea un container di API Docker Remote e l'opzione `--cap-add` a docker run.

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

`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 sezione Crea un container di API Docker Remote e l'opzione `--cap-add` a 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 sezione Crea un container di API Docker Remote e l'opzione `--cap-drop` a docker run.

`devices`

Qualsiasi dispositivi host da esporre nel container. Questo parametro è mappato a `Devices` nella sezione Crea un container di API Docker Remote e l'opzione `--device` a docker run.

> **Note**
>
> Il parametro `devices` non è supportato quando utilizzi il tipo di avvio Fargate o i container Windows.

Tipo: matrice di oggetti Device

Campo obbligatorio: no

`hostPath`

Il percorso per il dispositivo nell'istanza di container dell'host.

Tipo: stringa

Campo obbligatorio: sì

`containerPath`

Il percorso nel container in cui esporre il dispositivo dell'host.

- Tipo: stringa

  Campo obbligatorio: no

`permissions`

Le autorizzazioni esplicite da fornire al container per il dispositivo. Di default, il container dispone di autorizzazioni per `read`, `write` e `mknod` sul dispositivo.

Tipo: matrice di stringhe

Valori validi: `read` | `write` | `mknod`

`initProcessEnabled`

Esegui un processo `init` nel container che inoltra segnali e raccoglie i processi. Questo parametro è mappato all'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 omesso, il container utilizza la configurazione di swap per l'istanza di container su cui è in esecuzione. È necessario impostare un valore `maxSwap` per il parametro `swappiness` da utilizzare.

> **Note**
>
> Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro `maxSwap` non è supportato.

`sharedMemorySize`

Valore per le dimensioni (in MiB) del volume `/dev/shm`. Questo parametro è mappato all'opzione `--shm-size` su docker run.

> **Note**
>
> Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro `sharedMemorySize` non è supportato.

Tipo: integer

`swappiness`

Puoi utilizzare questo parametro per ottimizzare il funzionamento swappiness della memoria di un container. Un valore `swappiness` di `0` impedisce che si verifichi lo scambio, a meno che non sia necessario. Un valore `swappiness` di `100` fa sì che le pagine vengano scambiate frequentemente. I valori accettati sono numeri interi compresi tra `0` e `100`. Se non specifichi un valore, viene utilizzato il valore predefinito di `60`. Inoltre, se non specifichi un valore per `maxSwap`, questo parametro verrà ignorato. Questo parametro è mappato all'opzione `--memory-swappiness` su docker run.

> **Note**
>
> Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro `swappiness` non è supportato.
>
> 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.

> **Note**
>
> Se utilizzi attività che ricorrono al tipo di avvio Fargate, il parametro `tmpfs` non è supportato.

Tipo: matrice di oggetti Tmpfs

Campo obbligatorio: no

`containerPath`

Percorso assoluto in cui deve essere montato il volume tmpfs.

Tipo: stringa

Campo obbligatorio: sì

`mountOptions`

L'elenco delle opzioni di montaggio del volume tmpfs.

Tipo: matrice di stringhe

Campo obbligatorio: no

Valori validi: `"defaults" | "ro" | "rw" | "suid" | "nosuid" | "dev" | "nodev" | "exec" | "noexec" | "sync" | "async" | "dirsync" | "remount" | "mand" | "nomand" | "atime" | "noatime" | "diratime" | "nodiratime" | "bind" | "rbind" | "unbindable" | "runbindable" | "private" | "rprivate" | "shared" | "rshared" | "slave" | "rslave" | "relatime" | "norelatime" | "strictatime" | "nostrictatime" | "mode" | "uid" | "gid" | "nr_inodes" | "nr_blocks" | "mpol"`

`size`

Le dimensioni massime (in MiB) del volume tmpfs.

Tipo: integer

Campo obbligatorio: sì

Dipendenze per i container

`dependsOn`

Tipo: matrice di oggetti ContainerDependency

Campo obbligatorio: no

Le dipendenze definite per l'avvio e la chiusura dei container. Un container può contenere più dipendenze. Se una dipendenza è definita per l'avvio del container, per la sua chiusura è invertita. Per un esempio, consulta Dipendenze per i container.

> **Note**
>
> Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Per le attività Amazon ECS ospitate su istanze Amazon EC2, le istanze richiedono almeno una 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 AMI Linux ottimizzate per Amazon ECS.

Per le attività Amazon ECS ospitate su Fargate, 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

Campo obbligatorio: sì

Il nome del container che deve soddisfare la condizione specificata.

`condition`

Tipo: stringa

Campo obbligatorio: sì

La condizione di dipendenza del container. Di seguito sono elencate le condizioni disponibili e il loro comportamento:

- `START`: questa condizione emula il comportamento dei collegamenti e dei volumi. La condizione convalida l'avvio di un container dipendente prima di consentire l'avvio di altri container.
- `COMPLETE`: questa condizione verifica l'esecuzione fino al completamento (uscita) di un container dipendente prima di consentire l'avvio di altri container. Può rivelarsi utile per container non essenziali che eseguono uno script e quindi escono. Questa condizione non può essere impostata su un container essenziale.
- `SUCCESS`: questa condizione è uguale a `COMPLETE`, ma richiede anche che il container esca con stato `zero`. Questa condizione non può essere impostata su un container essenziale.
- `HEALTHY`: questa condizione verifica che il container dipendente superi il controllo dell'integrità del container prima di consentire l'avvio di altri container. Ciò richiede che per il container dipendente siano configurati i controlli dell'integrità nella definizione di attività. Questa condizione viene confermata solo all'avvio dell'attività.

Timeout del container

`startTimeout`

Tipo: integer

Campo obbligatorio: no

Valori di esempio: `120`

Tempo di attesa (in secondi) prima di rinunciare a risolvere le dipendenze per un container.

Ad esempio, vengono specificati due container in una definizione di attività: `containerA` ha una dipendenza sul `containerB` quando raggiunge lo stato `COMPLETE`, `SUCCESS` o `HEALTHY`. Se per `containerB` è specificato un valore `startTimeout` e non raggiunge lo stato desiderato entro tale periodo di tempo, allora `containerA` non viene avviato.

> **Note**
>
> Se un container non soddisfa un vincolo di dipendenza o si verifica un timeout prima di rispondere al vincolo, Amazon ECS non avanza i container dipendenti allo stato successivo.

Per le attività Amazon ECS ospitate su Fargate, 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

Campo obbligatorio: no

Valori di esempio: `120`

Durata (in secondi) di attesa prima che sia forzata la chiusura se il container non si arresta da solo normalmente.

Per le attività Amazon ECS ospitate su Fargate, 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.

Per le attività che utilizzano il tipo di avvio EC2, 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 AMI Linux ottimizzate per Amazon ECS.

Controlli di sistema

`systemControls`

Tipo: oggetto SystemControl

Campo obbligatorio: no

Un elenco di parametri del kernel dello spazio dei nomi da impostare nel contenitore. Questo parametro è mappato a `Sysctls` nella sezione Create a container di Docker Remote API e l'opzione `--sysctl` a 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`, tra cui Fargate, se hai impostato `systemControls` per qualsiasi container, questa impostazione si applica a tutti i container nell'attività. Se hai impostato diversi `systemControls` per più container in un'unica attività, il container che viene avviato per ultimo determina quale `systemControls` diventa effettivo.
- Per attività che utilizzano la modalità di rete `host`, lo spazio dei nomi della rete `systemControls` non è supportato.

Se stai impostando uno spazio dei nomi della risorsa IPC per utilizzare i container nell'attività, ai controlli di sistema si applicano le seguenti condizioni. Per ulteriori informazioni, consulta Modalità IPC.

- Per le attività che utilizzano la modalità `host` IPC, i valori `systemControls` dello spazio dei nomi IPC non sono supportati.
- Per le attività che utilizzano la modalità IPC `task`, i valori di `systemControls` dello spazio dei nomi IPC si applicano a tutti i container all'interno di un'attività.

> **Note**
>
> Questo parametro non è supportato per i container Windows.

> **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

  Campo obbligatorio: no

  Il parametro del kernel dello spazio dei nomi per cui impostare un. `value`

  Valori dello spazio dei nomi IPC validi: `"kernel.msgmax" | "kernel.msgmnb" | "kernel.msgmni" | "kernel.sem" | "kernel.shmall" | "kernel.shmmax" | "kernel.shmmni" | "kernel.shm_rmid_forced"` e `Sysctls` che iniziano con `"fs.mqueue.*"`

  Valori dello spazio dei nomi di rete validi: `Sysctls` che iniziano con `"net.*"`

  Tutti questi valori sono supportati da Fargate.

`value`

- Tipo: stringa

  Campo obbligatorio: no

  Il valore per il parametro del kernel dello spazio dei nomi specificato in. `namespace`

Interactive

`interactive`

Tipo: Booleano

Campo obbligatorio: no

Quando questo parametro è `true`, puoi implementare le applicazioni containerizzate che richiedono l'allocazione di `stdin` o `tty`. Questo parametro è mappato a `OpenStdin` nella sezione Create a container di Docker Remote API e l'opzione `--interactive` a docker run.

Il valore predefinito è `false`.

Pseudoterminale

`pseudoTerminal`

Tipo: Booleano

Campo obbligatorio: no

Quando il parametro è `true`, è allocato un TTY. Questo parametro è mappato a `Tty` nella sezione Create a container di Docker Remote API e l'opzione `--tty` a docker run.

Il valore predefinito è `false`.

Nome dell'acceleratore di inferenza elastica

> **Note**
>
> A partire dal 15 aprile 2023, non AWS effettuerà l'onboarding di nuovi clienti in Amazon Elastic Inference (EI) e aiuterà i clienti attuali a migrare i propri carichi di lavoro verso opzioni che offrono prezzi e prestazioni migliori. Dopo il 15 aprile 2023, i nuovi clienti non saranno in grado di avviare istanze con acceleratori Amazon EI su Amazon, SageMaker Amazon ECS o Amazon EC2. Tuttavia, i clienti che hanno utilizzato Amazon EI almeno una volta negli ultimi 30 giorni sono considerati clienti attuali e potranno continuare a usufruire del servizio.

Il requisito della risorsa dell'acceleratore di Elastic Inference per la definizione di attività. Per ulteriori informazioni, consulta What Is Amazon Elastic Inference? nella Amazon Elastic Inference Developer Guide.

I parametri seguenti sono permessi nella definizione di un processo:

`deviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del dispositivo dell'acceleratore di inferenza elastica. Il `deviceName` deve anche essere riferito in una definizione del container; consulta `Elastic Inference accelerator`.

`deviceType`

Tipo: stringa

Campo obbligatorio: sì

L'acceleratore di inferenza elastica da utilizzare.

Vincoli di posizionamento delle attività

Quando registri una definizione di attività, puoi specificare vincoli di posizionamento dei processi che definiscono il modo in cui Amazon ECS posiziona i processi.

Se utilizzi il tipo di avvio Fargate, i vincoli di posizionamento delle attività non sono supportati. Di default, i processi Fargate sono distribuiti tra le zone di disponibilità.

Per i processi che utilizzano il tipo di avvio EC2, puoi utilizzare i vincoli per posizionare i processi in base alla zona di disponibilità, al tipo di istanza o agli attributi personalizzati. Per ulteriori informazioni, consulta Definisci quali istanze di container Amazon ECS utilizza per le attività.

I seguenti parametri sono consentiti in una definizione del container:

`expression`

-Tipo: stringa

Campo obbligatorio: no

Un'espressione del linguaggio di query del cluster da applicare al vincolo. Per ulteriori informazioni, consulta Crea espressioni per definire istanze di container per le attività di Amazon ECS.

`type`

Tipo: stringa

Campo obbligatorio: sì

Il tipo di vincolo. Utilizza `memberOf` per limitare la selezione a un gruppo di candidati validi.

Configurazione del proxy

`proxyConfiguration`

Tipo: oggetto ProxyConfiguration

Campo obbligatorio: no

I dettagli di configurazione del proxy App Mesh.

Per le attività che utilizzano il tipo di avvio EC2, per abilitare una configurazione proxy le istanze di container richiedono almeno la versione 1.26.0 dell'agente del container e almeno la versione 1.26.0-1 del pacchetto `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 AMI Linux ottimizzate per Amazon ECS.

Per le attività che utilizzano il tipo di avvio Fargate, questa funzionalità richiede che l'attività o il servizio utilizzi la versione della piattaforma 1.3.0 o successiva.

> **Note**
>
> Questo parametro non è supportato per i container Windows.

```
"proxyConfiguration": {
    "type": "APPMESH",
    "containerName": "string",
    "properties": [
        {
           "name": "string",
           "value": "string"
        }
    ]
}
```

`type`

- Tipo: stringa

Valori del valore: `APPMESH`

Campo obbligatorio: no

Il tipo di proxy. L'unico valore supportato è `APPMESH`.

`containerName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del container che agirà come proxy App Mesh.

`properties`

Tipo: matrice di oggetti KeyValuePair

Campo obbligatorio: no

Il set di parametri di configurazione di rete per il plug-in Container Network Interface (CNI), specificati come coppie chiave-valore.

- `IgnoredUID`: (obbligatorio) l'ID utente (UID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato `IgnoredGID`, questo campo può rimanere vuoto.
- `IgnoredGID`: (obbligatorio) l'ID gruppo (GID) del container proxy come definito dal parametro `user` in una definizione del container. Serve a garantire che il proxy ignori il proprio traffico. Se è specificato `IgnoredUID`, questo campo può rimanere vuoto.
- `AppPorts` (obbligatorio): l'elenco delle porte utilizzate dall'applicazione. Il traffico di rete per queste porte viene inoltrato alle porte `ProxyIngressPort` e `ProxyEgressPort`.
- `ProxyIngressPort` (obbligatorio): specifica la porta a cui è diretto il traffico in entrata per le `AppPorts`.
- `ProxyEgressPort` (obbligatorio): specifica la porta a cui è diretto il traffico in uscita da `AppPorts`.
- `EgressIgnoredPorts`: (obbligatorio) il traffico in uscita diretto verso le porte specificate viene ignorato e non reindirizzato alla `ProxyEgressPort`. Può essere un elenco vuoto.
- `EgressIgnoredIPs`: (obbligatorio) il traffico in uscita diretto verso gli indirizzi IP specificati viene ignorato e non reindirizzato alla `ProxyEgressPort`. Può essere un elenco vuoto.

`name`

- Tipo: stringa

Campo obbligatorio: no

Nome della coppia chiave-valore.

`value`

- Tipo: stringa

Campo obbligatorio: no

Valore della coppia chiave-valore.

Volumi

Quando si registra una definizione di attività, è possibile specificare facoltativamente un elenco di volumi da passare al Docker demone su un'istanza di contenitore, che diventa quindi disponibile per l'accesso da parte di altri contenitori sulla stessa istanza di contenitore.

Di seguito sono elencati i tipi di volumi di dati che è possibile utilizzare:

- Volumi Amazon EBS: fornisce storage a blocchi conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati. Puoi collegare 1 volume Amazon EBS per attività Amazon ECS durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. I volumi Amazon EBS sono supportati per le attività Linux ospitate su istanze Fargate o Amazon EC2. Per ulteriori informazioni, consulta Usa i volumi Amazon EBS con Amazon ECS.
- Volumi Amazon EFS: offre uno spazio di archiviazione di file semplice, scalabile e persistente da utilizzare con i processi Amazon ECS. Con Amazon EFS, la capacità di storage è elastica. La capacità di storage aumenta e si riduce automaticamente quando si aggiungono e si rimuovono i file. Le tue applicazioni possono disporre dello spazio di archiviazione di cui hanno bisogno nel momento in cui ne hanno bisogno. I volumi Amazon EFS sono supportati per le attività ospitate su istanze di Fargate o Amazon EC2. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.
- Volumi FSx for Windows File Server: fornisce server di file di Microsoft Windows completamente gestiti. Questi server di file sono supportati da un file system Windows. Quando utilizzi FSx for Windows File Server insieme ad Amazon ECS, puoi eseguire il provisioning delle attività di Windows con storage dei file persistente, distribuito, condiviso e statico. Per ulteriori informazioni, consulta Usa 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'istanza host di Amazon 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 durante l'esecuzione di attività su istanze Amazon EC2. I contenitori Windows supportano solo l'uso del driver. `local` Per usare i volumi Docker, specifica `dockerVolumeConfiguration` nella definizione di attività. Per ulteriori informazioni, consulta l'articolo relativo all'utilizzo dei volumi.
- Bind mounts: un file o una directory sul computer host montato in un contenitore. I volumi host Bind Mount sono supportati durante l'esecuzione di attività su istanze AWS Fargate o Amazon EC2. 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 l'articolo relativo all'utilizzo dei montaggi vincolati.

Per ulteriori informazioni, consulta Opzioni di storage per le attività di Amazon ECS.

I seguenti parametri sono consentiti in una definizione del container.

`name`

▬Tipo: stringa

Campo obbligatorio: no

Nome del volume. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, trattini () e trattini bassi (). - _ A questo nome viene fatto riferimento nel parametro dell'oggetto di definizione del contenitore. `sourceVolume` `mountPoints`

`host`

Campo obbligatorio: no

Il parametro `host` viene utilizzato per legare il ciclo di vita del montaggio vincolato all'istanza host di Amazon EC2 anziché al processo, dove invece è archiviato. 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 istanze Amazon EC2.

`sourcePath`

-Tipo: stringa

Campo obbligatorio: no

Quando viene utilizzato il parametro `host`, specifica un `sourcePath` per dichiarare il percorso sull'istanza Amazon EC2 dell'host presentata al container. Se questo parametro è vuoto, il daemon Docker assegna automaticamente un percorso host. Se il parametro `host` contiene una posizione del file `sourcePath`, il volume di dati rimane nella posizione specificata sull'istanza Amazon EC2 dell'host finché non viene eliminato manualmente. Se il valore `sourcePath` non esiste nell'istanza Amazon EC2 dell'host, viene creato automaticamente dal daemon Docker. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

`configuredAtLaunch`

Tipo: Booleano

Campo obbligatorio: no

Speciifica se un volume è configurabile all'avvio. Se impostato su`true`, è possibile configurare il volume durante l'esecuzione di un'attività autonoma o durante la creazione o l'aggiornamento di un servizio. Se impostato su`true`, non sarà possibile fornire un'altra configurazione del volume nella definizione dell'attività. Questo parametro deve essere impostato per `true` configurare un volume Amazon EBS da allegare a un'attività. L'impostazione `configuredAtLaunch` `true` e il rinvio della configurazione del volume alla fase di avvio consentono di creare definizioni di attività che non sono limitate a un tipo di volume o a impostazioni di volume specifiche. In questo modo la definizione delle attività è riutilizzabile in diversi ambienti di esecuzione. Per ulteriori informazioni, consulta Volumi Amazon EBS.

`dockerVolumeConfiguration`

Tipo: oggetto DockerVolumedi configurazione

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi docker. I volumi Docker sono supportati solo durante l'esecuzione di attività su istanze EC2. I contenitori Windows supportano solo l'uso del driver. `local` Per utilizzare i montaggi vincolati, specifica invece un `host`.

`scope`

- Tipo: stringa

  Valori validi: `task` | `shared`

  Campo obbligatorio: no

  L'ambito del volume Docker che determina il suo ciclo di vita. I volumi Docker che rientrano nell'ambito `task` vengono automaticamente assegnati all'avvio del processo e distrutti quando il processo viene arrestato. I volumi Docker che vengono definiti come `shared` vengono mantenuti dopo l'arresto del processo.

`autoprovision`

  Tipo: Booleano

  Valore predefinito: `false`

  Campo obbligatorio: no

  Se questo valore è `true`, viene creato il volume Docker, se non è già presente. Questo campo viene utilizzato solo se `scope` è`shared`. Se `scope` è`task`, allora questo parametro deve essere omesso o impostato `false` su.

`driver`

- Tipo: stringa

  Campo obbligatorio: no

  Il driver del volume Docker da utilizzare. Il valore del driver deve corrispondere al nome del driver fornito da Docker perché questo nome viene utilizzato per il posizionamento delle attività. Se il driver è stato installato utilizzando la CLI del plug-in Docker, `docker plugin ls` utilizzalo per recuperare il nome del driver dall'istanza del contenitore. Se il driver è stato installato utilizzando un altro metodo, utilizza Docker plugin discovery per recuperare il nome del driver. Per ulteriori informazioni, consulta l'argomento relativo al rilevamento del plug-in Docker. Questo parametro fa riferimento a `Driver` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--driver` in `docker volume create`.

`driverOpts`

- Tipo: stringa

  Campo obbligatorio: no

  Una mappa delle opzioni specifiche del driver Docker da esaminare. Questo parametro fa riferimento a `DriverOpts` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--opt` in `docker volume create`.

`labels`

- Tipo: stringa

  Campo obbligatorio: no

  Metadati personalizzati da aggiungere al volume Docker. Questo parametro fa riferimento a `Labels` nella sezione Create a volume (Crea un volume) di Docker Remote API e all'opzione `--label` in `docker volume create`.

`efsVolumeConfiguration`

Tipo: VolumeConfiguration oggetto EFS

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

- Tipo: stringa

  Campo obbligatorio: no

  La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omesso, verrà utilizzata la radice del volume Amazon EFS. La specifica di `/` avrà lo stesso effetto dell'omissione di questo parametro.

> **Important**
>
> Se un punto di accesso EFS è specificato in`authorizationConfig`, il parametro della directory principale deve essere omesso o impostato su`/`, il che applicherà il percorso impostato sul punto di accesso EFS.

`transitEncryption`

- Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Specifica se abilitare o meno la crittografia per i dati Amazon EFS in transito tra l'host Amazon ECS e il server Amazon EFS. Se si utilizza l'autorizzazione IAM di Amazon EFS, è necessario abilitare la crittografia di transito. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di `DISABLED`. Per ulteriori informazioni, consulta Crittografia dei dati in transito nella Guida per l'utente di Amazon Elastic File System.

`transitEncryptionPort`

Tipo: integer

Campo obbligatorio: no

La porta da utilizzare per l'invio di dati crittografati tra l'host Amazon ECS e il server Amazon EFS. Se non specifichi una porta di crittografia di transito, l'attività utilizzerà la strategia di selezione delle porte utilizzata dall'helper di montaggio di Amazon EFS. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: AuthorizationConfiguration oggetto EFS

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

- Tipo: stringa

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory principale in `efsVolumeConfiguration` deve essere omesso o impostato su`/`, il che applicherà il percorso impostato sul punto di accesso EFS. Se si utilizza un punto di accesso, la crittografia di transito deve essere abilitata in `EFSVolumeConfiguration`. Per ulteriori informazioni, consulta Utilizzo dei punti di accesso Amazon EFS nella Guida per l'utente di Amazon Elastic File System.

`iam`

-Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Speciifica se utilizzare il ruolo IAM dell'attività Amazon ECS definito in una definizione di attività durante il montaggio del file system Amazon EFS. Se abilitato, la crittografia di transito deve essere abilitata nella casella `EFSVolumeConfiguration`. Se questo parametro viene omesso, viene utilizzato il comportamento predefinito di `DISABLED`. Per ulteriori informazioni consulta Ruoli IAM per le attività.

`FSxWindowsFileServerVolumeConfiguration`

Tipo: oggetto F SxWindows FileServer VolumeConfiguration

Campo obbligatorio: sì

Questo parametro viene specificato quando si utilizza un file system Amazon FSx for Windows File Server per lo storage delle attività.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system FSx for Windows File Server da utilizzare.

`rootDirectory`

Tipo: stringa

Campo obbligatorio: sì

La directory all'interno del file system FSx for Windows File Server da montare come directory root all'interno dell'host.

`authorizationConfig`

`credentialsParameter`

Tipo: stringa

Campo obbligatorio: sì

Le opzioni delle credenziali di autorizzazione.

opzioni:

- Amazon Resource Name (ARN) di un AWS Secrets Managersegreto.
- ARN di un AWS Systems Managerparametro.

`domain`

Tipo: stringa

Campo obbligatorio: sì

Un nome di dominio completo ospitato da una directory AWS Directory Service for Microsoft Active Directory(AWS Managed Microsoft AD) o da un Active Directory EC2 ospitato autonomamente.

Tag

Quando registri una definizione di attività, puoi facoltativamente specificare tag di metadati applicati alla definizione di attività. I tag consentono di suddividere in categorie e organizzare la definizione di attività. Ciascun tag è formato da una chiave e da un valore facoltativo. Li definisci entrambi. Per ulteriori informazioni, consulta Taggare le risorse Amazon ECS.

> **Important**
>
> Non aggiungere Informazioni personali di identificazione o altre informazioni riservate o sensibili nei tag. I tag sono accessibili a molti AWS servizi, inclusa la fatturazione. I tag non sono destinati a essere utilizzati per dati privati o sensibili.

I seguenti parametri sono consentiti in un oggetto di tag.

`key`

- Tipo: stringa

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

`value`

- Tipo: stringa

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

Altri parametri di definizione di attività

I seguenti parametri di definizione di attività possono essere utilizzati quando si registrano le definizioni di attività nella console Amazon ECS con l'opzione Configure via JSON (Configura tramite JSON). Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

Argomenti

- Archiviazione temporanea
- Modalità IPC
- Modalità PID

Archiviazione temporanea

`ephemeralStorage`

Tipo: oggetto EphemeralStorage

Campo obbligatorio: no

La quantità di archiviazione temporanea in GB da allocare per l'attività. Questo parametro viene utilizzato per espandere la quantità totale di archiviazione temporanea disponibile, oltre l'importo

predefinito, per le attività ospitate su AWS Fargate. Per ulteriori informazioni, consulta the section called “Montaggi vincolati”.

> **Note**
>
> Questo parametro è supportato solo per le attività ospitate su AWS Fargate che utilizzano la versione della piattaforma `1.4.0` o successive (Linux) oppure `1.0.0` o successive (Windows).

Modalità IPC

`ipcMode`

- Tipo: stringa

Campo obbligatorio: no

Lo spazio dei nomi della risorsa IPC da utilizzare per i container nell'attività. I valori validi sono `host`, `task` o `none`. Se è specificato `host`, tutti i container all'interno delle attività che hanno specificato la modalità `host` IPC sulla stessa istanza di container condividono le stesse risorse IPC con l'istanza host Amazon EC2. Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono le stesse risorse IPC. Se è stato specificato `none`, le risorse IPC all'interno dei container di un'attività sono private e non condivise con altri container in un'attività o sull'istanza di container. Se non è stato specificato alcun valore, la condivisione dello spazio dei nomi della risorsa IPC dipende dalle impostazioni del daemon Docker sull'istanza del container. Per ulteriori informazioni, consulta l'argomento relativo alle impostazioni IPC nella documentazione di riferimento di Docker run.

Se viene utilizzata la modalità `host` IPC, tieni presente che esiste un maggiore rischio di esposizione a spazi dei nomi IPC indesiderati. Per ulteriori informazioni, consulta Docker security.

Se stai impostando parametri kernel associati a uno spazio dei nomi utilizzando `systemControls` per i container nell'attività, allo spazio dei nomi della risorsa IPC si applica quanto segue. Per ulteriori informazioni, consulta Controlli di sistema.

- Per le attività che utilizzano la modalità `host` IPC, spazi dei nomi IPC correlati a `systemControls` non sono supportati.
- Per le attività che utilizzano la modalità `task` IPC, `systemControls` correlati allo spazio dei nomi IPC vengono applicati a tutti i container all'interno di un'attività.

> **Note**
>
> Questo parametro non è supportato per i container o le attività Windows che utilizzano il tipo di avvio Fargate.

Modalità PID

`pidMode`

- Tipo: stringa

  Valori validi: `host` | `task`

  Campo obbligatorio: no

  Lo spazio dei nomi del processo da utilizzare per i container nell'attività. I valori validi sono `host` o `task`. Per i container Fargate per 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 specificato `host`, tutti i container all'interno delle attività che hanno specificato la modalità `host` PID sulla stessa istanza di container condividono lo stesso spazio dei nomi dell'attività con l'istanza host Amazon EC2.

  Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono lo stesso spazio dei nomi del processo.

  Se non è stato specificato alcun valore, l'impostazione predefinita è uno spazio dei nomi privato per ogni container. Per ulteriori informazioni, consulta l'argomento relativo alle impostazioni PID nella documentazione di riferimento di Docker run.

  Se viene utilizzata la modalità `host` PID, esiste un maggiore rischio di esposizione a spazi dei nomi di attività indesiderati. Per ulteriori informazioni, consulta Docker security.

> **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.

Modello di definizione delle attività di Amazon ECS

Di seguito è riportato un modello di definizione di attività vuoto. Puoi utilizzare questo modello per creare la definizione dell'attività, che può quindi essere incollata nell'area di input JSON della console o salvata in un file e utilizzata con l'opzione. AWS CLI `--cli-input-json` Per ulteriori informazioni, consulta Parametri di definizione delle attività di Amazon ECS.

Modello del tipo di lancio di Amazon EC2

```
{
    "family": "",
    "taskRoleArn": "",
    "executionRoleArn": "",
    "networkMode": "none",
    "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": "COMPLETE"
            }
        ],
        "startTimeout": 0,
        "stopTimeout": 0,
        "hostname": "",
        "user": "",
        "workingDirectory": "",
        "disableNetworking": true,
        "privileged": true,
        "readonlyRootFilesystem": true,
        "dnsServers": [
            ""
        ],
        "dnsSearchDomains": [
```

```
                ""
            ],
            "extraHosts": [
                {
                    "hostname": "",
                    "ipAddress": ""
                }
            ],
            "dockerSecurityOptions": [
                ""
            ],
            "interactive": true,
            "pseudoTerminal": true,
            "dockerLabels": {
                "KeyName": ""
            },
            "ulimits": [
                {
                    "name": "nofile",
                    "softLimit": 0,
                    "hardLimit": 0
                }
            ],
            "logConfiguration": {
                "logDriver": "splunk",
                "options": {
                    "KeyName": ""
                },
                "secretOptions": [
                    {
                        "name": "",
                        "valueFrom": ""
                    }
                ]
            },
            "healthCheck": {
                "command": [
                    ""
                ],
                "interval": 0,
                "timeout": 0,
                "retries": 0,
                "startPeriod": 0
            },
```

```
            "systemControls": [
                {
                    "namespace": "",
                    "value": ""
                }
            ],
            "resourceRequirements": [
                {
                    "value": "",
                    "type": "InferenceAccelerator"
                }
            ],
            "firelensConfiguration": {
                "type": "fluentbit",
                "options": {
                    "KeyName": ""
                }
            }
        }
    ],
    "volumes": [
        {
            "name": "",
            "host": {
                "sourcePath": ""
            },
            "configuredAtLaunch": true,
            "dockerVolumeConfiguration": {
                "scope": "shared",
                "autoprovision": true,
                "driver": "",
                "driverOpts": {
                    "KeyName": ""
                },
                "labels": {
                    "KeyName": ""
                }
            },
            "efsVolumeConfiguration": {
                "fileSystemId": "",
                "rootDirectory": "",
                "transitEncryption": "DISABLED",
                "transitEncryptionPort": 0,
                "authorizationConfig": {
```

```
                "accessPointId": "",
                "iam": "ENABLED"
            }
        },
        "fsxWindowsFileServerVolumeConfiguration": {
            "fileSystemId": "",
            "rootDirectory": "",
            "authorizationConfig": {
                "credentialsParameter": "",
                "domain": ""
            }
        }
    }
],
"placementConstraints": [
    {
        "type": "memberOf",
        "expression": ""
    }
],
"requiresCompatibilities": [
    "EC2"
],
"cpu": "",
"memory": "",
"tags": [
    {
        "key": "",
        "value": ""
    }
],
"pidMode": "task",
"ipcMode": "task",
"proxyConfiguration": {
    "type": "APPMESH",
    "containerName": "",
    "properties": [
        {
            "name": "",
            "value": ""
        }
    ]
},
"inferenceAccelerators": [
```

```
        {
            "deviceName": "",
            "deviceType": ""
        }
    ],
    "ephemeralStorage": {
        "sizeInGiB": 0
    },
    "runtimePlatform": {
        "cpuArchitecture": "X86_64",
        "operatingSystemFamily": "WINDOWS_SERVER_20H2_CORE"
    }
}
```

Modello del tipo di lancio Fargate

Important

Per il tipo di lancio Fargate, è necessario includere il `operatingSystemFamily` parametro con uno dei seguenti valori:

- `LINUX`
- `WINDOWS_SERVER_2019_FULL`
- `WINDOWS_SERVER_2019_CORE`
- `WINDOWS_SERVER_2022_FULL`
- `WINDOWS_SERVER_2022_CORE`

```
{
    "family": "",
    "runtimePlatform": {"operatingSystemFamily": ""},
    "taskRoleArn": "",
    "executionRoleArn": "",
    "networkMode": "awsvpc",
    "platformFamily": "",
    "containerDefinitions": [
        {
            "name": "",
            "image": "",
```

```
            "repositoryCredentials": {"credentialsParameter": ""},
            "cpu": 0,
            "memory": 0,
            "memoryReservation": 0,
            "links": [""],
            "portMappings": [
                {
                    "containerPort": 0,
                    "hostPort": 0,
                    "protocol": "tcp"
                }
            ],
            "essential": true,
            "entryPoint": [""],
            "command": [""],
            "environment": [
                {
                    "name": "",
                    "value": ""
                }
            ],
            "environmentFiles": [
                {
                    "value": "",
                    "type": "s3"
                }
            ],
            "mountPoints": [
                {
                    "sourceVolume": "",
                    "containerPath": "",
                    "readOnly": true
                }
            ],
            "volumesFrom": [
                {
                    "sourceContainer": "",
                    "readOnly": true
                }
            ],
            "linuxParameters": {
                "capabilities": {
                    "add": [""],
                    "drop": [""]
```

```
                },
                "devices": [
                    {
                        "hostPath": "",
                        "containerPath": "",
                        "permissions": ["read"]
                    }
                ],
                "initProcessEnabled": true,
                "sharedMemorySize": 0,
                "tmpfs": [
                    {
                        "containerPath": "",
                        "size": 0,
                        "mountOptions": [""]
                    }
                ],
                "maxSwap": 0,
                "swappiness": 0
            },
            "secrets": [
                {
                    "name": "",
                    "valueFrom": ""
                }
            ],
            "dependsOn": [
                {
                    "containerName": "",
                    "condition": "HEALTHY"
                }
            ],
            "startTimeout": 0,
            "stopTimeout": 0,
            "hostname": "",
            "user": "",
            "workingDirectory": "",
            "disableNetworking": true,
            "privileged": true,
            "readonlyRootFilesystem": true,
            "dnsServers": [""],
            "dnsSearchDomains": [""],
            "extraHosts": [
                {
```

```
                "hostname": "",
                "ipAddress": ""
            }
        ],
        "dockerSecurityOptions": [""],
        "interactive": true,
        "pseudoTerminal": true,
        "dockerLabels": {"KeyName": ""},
        "ulimits": [
            {
                "name": "msgqueue",
                "softLimit": 0,
                "hardLimit": 0
            }
        ],
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {"KeyName": ""},
            "secretOptions": [
                {
                    "name": "",
                    "valueFrom": ""
                }
            ]
        },
        "healthCheck": {
            "command": [""],
            "interval": 0,
            "timeout": 0,
            "retries": 0,
            "startPeriod": 0
        },
        "systemControls": [
            {
                "namespace": "",
                "value": ""
            }
        ],
        "resourceRequirements": [
            {
                "value": "",
                "type": "GPU"
            }
        ],
```

```
            "firelensConfiguration": {
                "type": "fluentd",
                "options": {"KeyName": ""}
            }
        }
    ],
    "volumes": [
        {
            "name": "",
            "host": {"sourcePath": ""},
            "configuredAtLaunch":true,
            "dockerVolumeConfiguration": {
                "scope": "task",
                "autoprovision": true,
                "driver": "",
                "driverOpts": {"KeyName": ""},
                "labels": {"KeyName": ""}
            },
            "efsVolumeConfiguration": {
                "fileSystemId": "",
                "rootDirectory": "",
                "transitEncryption": "ENABLED",
                "transitEncryptionPort": 0,
                "authorizationConfig": {
                    "accessPointId": "",
                    "iam": "ENABLED"
                }
            }
        }
    ],
    "requiresCompatibilities": ["FARGATE"],
    "cpu": "",
    "memory": "",
    "tags": [
        {
            "key": "",
            "value": ""
        }
    ],
    "ephemeralStorage": {"sizeInGiB": 0},
    "pidMode": "task",
    "ipcMode": "none",
    "proxyConfiguration": {
        "type": "APPMESH",
```

```
        "containerName": "",
        "properties": [
            {
                "name": "",
                "value": ""
            }
        ]
    },
    "inferenceAccelerators": [
        {
            "deviceName": "",
            "deviceType": ""
        }
    ]
}
```

Puoi generare questo modello di definizione di attività utilizzando il seguente comando AWS CLI .

```
aws ecs register-task-definition --generate-cli-skeleton
```

Esempi di definizioni di attività Amazon ECS

È possibile copiare gli esempi e gli snippet per iniziare a creare definizioni di attività personalizzate.

Puoi copiare gli esempi e incollarli quando utilizzi l'opzione Configura tramite JSON nella console. Assicurati di personalizzare gli esempi, ad esempio l'utilizzo del tuo ID account. Puoi includere i frammenti di codice nella definizione dell'attività JSON. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console e Parametri di definizione delle attività di Amazon ECS.

Per altri esempi di definizione delle attività, consulta AWS Esempi di definizioni delle attività su. GitHub

Argomenti

- Server Web
- splunkdriver di registro
- fluentddriver di registro
- gelfdriver di registro
- Carichi di lavoro su istanze esterne

- Immagine Amazon ECR e definizione delle attività, ruolo IAM
- Punto di ingresso con comando
- Dipendenze per i container
- Definizioni di attività di esempio di Windows

Server Web

Di seguito è riportato un esempio di definizione di attività usando container Linux sul tipo di avvio Fargate che configura un server Web:

```
{
   "containerDefinitions": [
      {
         "command": [
            "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample App</
title> <style>body {margin-top: 40px; background-color: #333;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
 ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html && httpd-
foreground\""
         ],
         "entryPoint": [
            "sh",
            "-c"
         ],
         "essential": true,
         "image": "httpd:2.4",
         "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
               "awslogs-group" : "/ecs/fargate-task-definition",
               "awslogs-region": "us-east-1",
               "awslogs-stream-prefix": "ecs"
            }
         },
         "name": "sample-fargate-app",
         "portMappings": [
            {
               "containerPort": 80,
               "hostPort": 80,
               "protocol": "tcp"
```

```
            }
         ]
      }
   ],
   "cpu": "256",
   "executionRoleArn": "arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
   "family": "fargate-task-definition",
   "memory": "512",
   "networkMode": "awsvpc",
   "runtimePlatform": {
        "operatingSystemFamily": "LINUX"
    },
   "requiresCompatibilities": [
       "FARGATE"
    ]
}
```

Di seguito è riportato un esempio di definizione di attività usando container Windows sul tipo di avvio Fargate che configura un server Web:

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

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

splunkdriver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log `splunk` in una definizione di attività che invia i log a un servizio remoto. Il parametro di token Splunk è specificato come opzione segreta perché può essere trattato come dati sensibili. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

```
"containerDefinitions": [{
  "logConfiguration": {
   "logDriver": "splunk",
   "options": {
    "splunk-url": "https://cloud.splunk.com:8080",
    "tag": "tag_name",
   },
   "secretOptions": [{
    "name": "splunk-token",
    "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:splunk-token-
KnrBkD"
}],
```

fluentddriver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log `fluentd` in una definizione di attività che invia i log a un servizio remoto. Il valore `fluentd-address` è specificato come opzione segreta perché può essere trattato come dati sensibili. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

```
"containerDefinitions": [{
 "logConfiguration": {
  "logDriver": "fluentd",
  "options": {
   "tag": "fluentd demo"
  },
  "secretOptions": [{
   "name": "fluentd-address",
   "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:fluentd-address-
KnrBkD"
  }]
 },
 "entryPoint": [],
 "portMappings": [{
             "hostPort": 80,
             "protocol": "tcp",
             "containerPort": 80
             },
             {
  "hostPort": 24224,
  "protocol": "tcp",
  "containerPort": 24224
 }]
}],
```

`gelf`driver di registro

Il frammento di codice seguente mostra come utilizzare il driver di log `gelf` in una definizione di attività che invia i log a un host remoto. Tale host esegue Logstash e accetta i log di Gelf come input. Per ulteriori informazioni, consulta `logConfiguration`.

```
"containerDefinitions": [{
 "logConfiguration": {
  "logDriver": "gelf",
  "options": {
   "gelf-address": "udp://logstash-service-address:5000",
   "tag": "gelf task demo"
  }
 },
 "entryPoint": [],
 "portMappings": [{
   "hostPort": 5000,
```

```
   "protocol": "udp",
   "containerPort": 5000
  },
  {
   "hostPort": 5000,
   "protocol": "tcp",
   "containerPort": 5000
  }
 ]
}],
```

Carichi di lavoro su istanze esterne

Durante la registrazione di una definizione di attività di Amazon ECS, utilizza il parametro `requiresCompatibilities` e specifica `EXTERNAL`che verifica che la definizione di attività è compatibile per l'uso durante l'esecuzione di carichi di lavoro Amazon ECS sulle istanze esterne. Se registri una definizione di attività tramite la console, devi utilizzare l'editor JSON. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

> **Important**
>
> Se i processi richiedono un ruolo IAM di esecuzione del processo, assicurati che sia specificato nella definizione di attività.

Quando distribuisci il tuo carico di lavoro, utilizza il tipo di avvio `EXTERNAL` durante la creazione del servizio o l'esecuzione del processo autonomo.

Di seguito è riportata una definizione di tabella di esempio.

Linux

```
{
 "requiresCompatibilities": [
  "EXTERNAL"
 ],
 "containerDefinitions": [{
  "name": "nginx",
  "image": "public.ecr.aws/nginx/nginx:latest",
  "memory": 256,
  "cpu": 256,
```

```
  "essential": true,
  "portMappings": [{
   "containerPort": 80,
   "hostPort": 8080,
   "protocol": "tcp"
  }]
 }],
 "networkMode": "bridge",
 "family": "nginx"
}
```

Windows

```
{
 "requiresCompatibilities": [
  "EXTERNAL"
 ],
 "containerDefinitions": [{
  "name": "windows-container",
  "image": "mcr.microsoft.com/windows/servercore/iis:windowsservercore-ltsc2019",
  "memory": 256,
  "cpu": 512,
  "essential": true,
  "portMappings": [{
   "containerPort": 80,
   "hostPort": 8080,
   "protocol": "tcp"
  }]
 }],
 "networkMode": "bridge",
 "family": "windows-container"
}
```

Immagine Amazon ECR e definizione delle attività, ruolo IAM

Il frammento di codice seguente utilizza un'immagine Amazon ECR denominata `aws-nodejs-sample` con il tag `v1` dal registro `123456789012.dkr.ecr.us-west-2.amazonaws.com`. Il container in questo processo eredita le autorizzazioni IAM dal ruolo `arn:aws:iam::123456789012:role/AmazonECSTaskS3BucketRole`. Per ulteriori informazioni, consulta Ruolo IAM dell'attività Amazon ECS.

```
{
    "containerDefinitions": [
        {
            "name": "sample-app",
            "image": "123456789012.dkr.ecr.us-west-2.amazonaws.com/aws-nodejs-
sample:v1",
            "memory": 200,
            "cpu": 10,
            "essential": true
        }
    ],
    "family": "example_task_3",
    "taskRoleArn": "arn:aws:iam::123456789012:role/AmazonECSTaskS3BucketRole"
}
```

Punto di ingresso con comando

Il frammento di codice seguente mostra la sintassi per un container Docker che utilizza un punto di ingresso e un argomento del comando. Questo container esegue il ping di `google.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"
```

```
}
```

Definizioni di attività di esempio di Windows

Di seguito è riportata una definizione di attività di esempio che consente di iniziare a utilizzare i container Windows su Amazon ECS.

Example Applicazione di esempio della console Amazon ECS per Windows

La seguente definizione di attività è l'applicazione di esempio della console Amazon ECS che viene prodotta nella procedura guidata per la prima esecuzione di Amazon ECS; è stata esportata per utilizzare l'immagine del container Windows `microsoft/iis`.

```
{
  "family": "windows-simple-iis",
  "containerDefinitions": [
    {
      "name": "windows_sample_app",
      "image": "mcr.microsoft.com/windows/servercore/iis",
      "cpu": 1024,
      "entryPoint":["powershell", "-Command"],
      "command":["New-Item -Path C:\\inetpub\\wwwroot\\index.html -Type file -
Value '<html> <head> <title>Amazon ECS Sample App</title> <style>body {margin-top:
 40px; background-color: #333;} </style> </head><body> <div style=color:white;text-
align:center> <h1>Amazon ECS Sample App</h1> <h2>Congratulations!</h2> <p>Your
 application is now running on a container in Amazon ECS.</p>'; C:\\ServiceMonitor.exe
 w3svc"],
      "portMappings": [
        {
          "protocol": "tcp",
          "containerPort": 80
        }
      ],
      "memory": 1024,
      "essential": true
    }
  ],
  "networkMode": "awsvpc",
  "memory": "1024",
  "cpu": "1024"
}
```

Cluster Amazon ECS

Un cluster Amazon ECS è un raggruppamento logico di processi o servizi. Oltre alle attività e ai servizi, un cluster è composto dalle seguenti risorse:

- La capacità dell'infrastruttura, che può essere una combinazione delle seguenti:
  - Istanze Amazon EC2 nel cloud AWS
  - Serverless (AWS Fargate (Fargate)) nel cloud AWS
  - Macchine virtuali (VM) o server on-premise
- La rete (VPC e sottorete) su cui vengono eseguite le attività e i servizi

  Quando utilizzi istanze Amazon EC2 per la capacità, la sottorete può trovarsi in zone di disponibilità, locali, di lunghezza d'onda o AWS Outposts.
- Uno spazio dei nomi facoltativo

  Il namespace viene utilizzato per la service-to-service comunicazione con Service Connect.
- Un'opzione di monitoraggio

  CloudWatch Container Insights ha un costo aggiuntivo ed è un servizio completamente gestito. Raccoglie, aggrega e riepiloga automaticamente parametri e log di Amazon ECS.

Di seguito sono elencati i concetti generali sui cluster Amazon ECS.

- Amazon ECS crea un cluster predefinito. Puoi creare cluster aggiuntivi per separare le tue risorse.
- I cluster sono Regione AWS specifici.
- I cluster possono trovarsi in uno dei seguenti stati.

  ACTIVE

  Il cluster è pronto per accettare i processi e, se applicabile, puoi registrare le istanze del container con il cluster.

  PROVISIONING

  Al cluster sono associati provider di capacità e vengono create le risorse necessarie per il provider di capacità.

DEPROVISIONING

Al cluster sono associati provider di capacità e le risorse necessarie per il provider di capacità vengono eliminate.

Non riuscito

Al cluster sono associati provider di capacità e le risorse necessarie per il provider di capacità non sono state create.

INACTIVE

Il cluster è stato eliminato. I cluster con stato `INACTIVE` potrebbero rimanere individuabili nel tuo account per un periodo di tempo. Questo comportamento è soggetto a modifiche future, quindi assicurati di non fare affidamento sulla persistenza `INACTIVE` dei cluster.

- Un cluster può contenere una combinazione di attività ospitate su AWS Fargate istanze Amazon EC2 o istanze esterne. Le attività possono essere eseguite sull'infrastruttura Fargate o EC2 in base al tipo di avvio o al provider di capacità. Se utilizzi EC2 come tipo di lancio, Amazon ECS non monitora e non ridimensiona la capacità dei gruppi Amazon EC2 Auto Scaling. Per ulteriori informazioni sui tipi di avvio, consulta Tipi di avvio di Amazon ECS.
- Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Una strategia per i fornitori di capacità può contenere solo i fornitori di capacità del gruppo Auto Scaling o i fornitori di capacità Fargate.
- Puoi utilizzare diversi tipi di istanza per il tipo di avvio EC2 o per i provider di capacità del gruppo Auto Scaling. Un'istanza può essere registrata su un solo cluster alla volta.
- Puoi limitare l'accesso ai cluster creando policy IAM personalizzate. Per informazioni, consulta la Esempi di cluster Amazon ECS sezione inEsempi di policy basate su identità per Amazon Elastic Container Service.
- È possibile utilizzare Service Auto Scaling per scalare le attività di Fargate. Per ulteriori informazioni, consulta Ridimensiona automaticamente il tuo servizio Amazon ECS.
- È possibile configurare uno spazio dei nomi Service Connect predefinito per un cluster. Dopo aver impostato uno spazio dei nomi predefinito di Service Connect, tutti i nuovi servizi con Service Connect attivato creati nel cluster saranno aggiunti come servizi client nello spazio dei nomi. e non sono necessarie ulteriori configurazioni. Per ulteriori informazioni, consulta Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi.

Cluster Amazon ECS per il tipo di lancio Fargate

I provider di capacità Amazon ECS gestiscono il dimensionamento dell'infrastruttura per le attività nei cluster. Ogni cluster ha uno o più provider di capacità e una strategia di provider di capacità facoltativa. La strategia del provider di capacità determina il modo in cui le attività vengono distribuite tra i provider di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del provider di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster.

Quando si eseguono le attività AWS Fargate, non è necessario creare o gestire la capacità. È sufficiente associare uno dei seguenti provider di capacità predefiniti al cluster:

- Fargate
- Fargate Spot

Con Amazon ECS on AWS Fargate capacity provider, puoi utilizzare sia la capacità di Fargate che quella di Fargate Spot per le tue attività Amazon ECS.

Con Fargate Spot è possibile eseguire le attività di Amazon ECS con tolleranza alle interruzioni a una tariffa scontata rispetto al prezzo di Fargate. Fargate Spot esegue le attività nella capacità di elaborazione di riserva. Quando è AWS necessario ripristinare la capacità, le attività vengono interrotte con un avviso di due minuti. Fargate Spot supporta solo le attività Linux con l'architettura X86_64 sulla piattaforma versione 1.3.0 o successiva.

Quando le attività che utilizzano i provider di capacità Fargate e Fargate Spot vengono interrotte, l'evento di modifica dello stato dell'attività viene inviato ad Amazon. EventBridge Il motivo dell'interruzione descrive la causa. Per ulteriori informazioni, consulta Eventi di modifica dello stato delle attività di Amazon ECS.

Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e di Fargate. Tuttavia, una strategia di provider di capacità può contenere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi. Per ulteriori informazioni, consulta Auto Scaling Group Capacity Provider.

Quando si utilizzano provider di capacità:

- È necessario associare un provider di capacità a un cluster prima di associarlo alla strategia del fornitore di capacità.

- È possibile specificare un massimo di 20 provider di capacità per una strategia di provider di capacità.
- Non è possibile aggiornare un servizio che utilizza un provider di capacità di un gruppo con scalabilità automatica per utilizzare un provider di capacità Fargate. È vero anche il contrario.
- In una strategia del provider di capacità, se non viene specificato alcun valore di `weight` per un provider di capacità nella console, allora viene utilizzato il valore predefinito 1. Se si utilizza l'API o AWS CLI, `0` viene utilizzato il valore predefinito di.
- Quando più provider di capacità sono specificati nell'ambito di una strategia di provider di capacità, almeno uno dei provider deve avere un valore di peso maggiore di zero. I provider di capacità con un peso pari a zero non vengono utilizzati per collocare le attività. Se specifichi più provider di capacità in una strategia tutti con un peso pari a zero, allora qualsiasi operazione `RunTask` o `CreateService` che utilizza la strategia del provider di capacità avrà esito negativo.
- In una strategia di provider di capacità, solo un provider di capacità può avere un valore di base definito. Se non viene specificato alcun valore, viene utilizzato il valore predefinito zero.
- Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Tuttavia, una strategia di provider di capacità può includere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi.
- Un cluster può contenere una combinazione di servizi e attività autonome che utilizzano sia i provider di capacità che i tipi di avvio. Un servizio può essere aggiornato per utilizzare una strategia del provider di capacità anziché un tipo di avvio. Tuttavia, quando si esegue questa operazione è necessario forzare una nuova implementazione.

Avvisi di risoluzione di Fargate Spot

Nei periodi con domanda estremamente elevata, la capacità Fargate spot potrebbe non essere disponibile. Ciò può causare ritardi nelle attività di Fargate Spot. Quando ciò accade, i servizi Amazon ECS riprovano ad avviare le attività finché non diventa disponibile la capacità richiesta. Fargate non sostituisce la capacità spot con quella on demand.

Quando le attività che utilizzano la capacità Fargate Spot vengono interrotte a causa di un'interruzione Spot, viene inviato un avviso due minuti prima dell'arresto del processo. L'avviso viene inviato come evento di modifica dello stato dell'attività ad Amazon EventBridge e come segnale SIGTERM all'attività in esecuzione. Quando si utilizza Fargate Spot come parte di un servizio, lo scheduler del servizio riceve il segnale di interruzione e prova ad avviare altre attività su Fargate Spot

se c'è capacità disponibile. Un servizio con una sola attività verrà interrotto fino a quando la capacità non sarà disponibile. Per ulteriori informazioni su un arresto corretto, consulta Arresto regolare con ECS.

Per assicurarti che i container si chiudano correttamente prima che l'attività si interrompa, puoi configurare le seguenti impostazioni:

- È possibile specificare un valore `stopTimeout` di `120` secondi al massimo nella definizione del container utilizzata dall'attività. Il valore di `stopTimeout` predefinito è 30 secondi. È possibile specificare un valore `stopTimeout` più lungo per avere più tempo tra il momento in cui viene ricevuto l'evento di modifica dello stato delle attività e il momento in cui viene forzato l'arresto del container. Per ulteriori informazioni, consulta Timeout del container.
- Il segnale SIGTERM deve essere ricevuto dall'interno del container per eseguire qualsiasi operazione di pulizia. La mancata elaborazione di questo segnale comporterà la ricezione di un segnale SIGKILL dopo la configurazione di `stopTimeout` e può comportare la perdita o il danneggiamento dei dati.

Di seguito è riportato un frammento di un evento di modifica dello stato dell'attività. Questo frammento visualizza il motivo dell'arresto e il codice di arresto per un'interruzione di Fargate Spot.

```
{
  "version": "0",
  "id": "9bcdac79-b31f-4d3d-9410-fbd727c29fab",
  "detail-type": "ECS Task State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "resources": [
    "arn:aws:ecs:us-east-1:111122223333:task/b99d40b3-5176-4f71-9a52-9dbd6f1cebef"
  ],
  "detail": {
    "clusterArn": "arn:aws:ecs:us-east-1:111122223333:cluster/default",
    "createdAt": "2016-12-06T16:41:05.702Z",
    "desiredStatus": "STOPPED",
    "lastStatus": "RUNNING",
    "stoppedReason": "Your Spot Task was interrupted.",
    "stopCode": "SpotInterruption",
    "taskArn": "arn:aws:ecs:us-east-1:111122223333:task/
b99d40b3-5176-4f71-9a52-9dbd6fEXAMPLE",
    ...
  }
```

```
}
```

Di seguito è riportato uno schema di eventi utilizzato per creare una EventBridge regola per gli eventi di modifica dello stato delle attività di Amazon ECS. Facoltativamente, puoi specificare un cluster nel campo `detail`. Ciò significa che per quel cluster riceverai eventi di modifica dello stato delle attività. Per ulteriori informazioni, consulta Creating an EventBridge Rule nella Amazon EventBridge User Guide.

```
{
    "source": [
        "aws.ecs"
    ],
    "detail-type": [
        "ECS Task State Change"
    ],
    "detail": {
        "clusterArn": [
            "arn:aws:ecs:us-west-2:111122223333:cluster/default"
        ]
    }
}
```

Creazione di un cluster Amazon ECS per il tipo di lancio Fargate

Puoi creare un cluster Amazon ECS utilizzando la console Amazon ECS. Prima di iniziare, accertati di aver completato le fasi in Configurazione per l'uso di Amazon ECS e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta the section called “Esempi di cluster Amazon ECS”. La console Amazon ECS crea le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

La console associa automaticamente i provider di capacità Fargate e Fargate Spot al cluster.

Oltre al cluster, la console crea automaticamente le seguenti risorse:

- Uno spazio dei nomi predefinito AWS Cloud Map che ha lo stesso nome del cluster. Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive.

  Per ulteriori informazioni, consulta Interconnetti i servizi Amazon ECS.

È possibile modificare le seguenti opzioni:

- Modifica lo spazio dei nomi predefinito associato al cluster.
- Attiva Container Insights.

  CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights”.
- Aggiungi tag per facilitare l'identificazione del cluster.

Procedura

Creazione di un nuovo cluster (console Amazon ECS)

1. Apri la console all'indirizzo 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 Configurazione del cluster, configura gli elementi seguenti:
   - In Nome cluster, inserisci un nome univoco.

     Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.
   - (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. (Facoltativo) 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. Seleziona Crea.

Passaggi successivi

Dopo aver creato il cluster, è possibile creare le definizioni delle attività per le applicazioni e quindi eseguirle come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Definizioni dei processi di Amazon ECS
- Esecuzione di un'applicazione come attività Amazon ECS
- Creazione di un servizio Amazon ECS utilizzando la console

Provider di capacità Amazon ECS per il tipo di lancio EC2

Quando utilizzi le istanze Amazon EC2 per la capacità, puoi sfruttare i gruppi con dimensionamento automatico per gestire le istanze Amazon EC2 registrate nei cluster. Auto Scaling aiuta a garantire il numero corretto di istanze Amazon EC2 disponibili per gestire il carico dell'applicazione.

Puoi utilizzare la funzionalità di scalabilità gestita per consentire ad Amazon ECS di gestire le azioni di scalabilità in e orizzontale del gruppo Auto Scaling oppure puoi gestire tu stesso le azioni di scalabilità. Per ulteriori informazioni, consulta Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster.

Ti consigliamo di creare un nuovo gruppo Auto Scaling vuoto. Se utilizzi un gruppo con scalabilità automatica esistente, qualsiasi istanza Amazon EC2 associata al gruppo che era già in esecuzione e registrata con un cluster Amazon ECS prima dell'uso del gruppo con scalabilità automatica per creare un provider di capacità potrebbe non essere registrata correttamente con 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 lancio Amazon EC2 e un gruppo Auto Scaling per tuo conto come parte dello stack. AWS CloudFormation Sono preceduti da. `EC2ContainerService-<ClusterName>` Puoi utilizzare il gruppo con dimensionamento automatico come provider di capacità per tale cluster.

Ti consigliamo di utilizzare il drenaggio gestito delle istanze per consentire la chiusura graduale delle istanze Amazon EC2 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 su istanze EC2

Quando utilizzi i provider di capacità del gruppo con dimensionamento automatico nella console, è opportuno considerare quanto segue:

- Un gruppo Auto Scaling deve avere un valore `MaxSize` maggiore di zero per l'aumento orizzontale.
- Il gruppo Auto Scaling non può avere impostazioni di ponderazione delle istanze.
- Se il gruppo con dimensionamento automatico non è in grado di impiegare la scalabilità orizzontale per adattarsi al numero di esecuzioni di attività, le attività non riusciranno ad andare oltre lo stato `PROVISIONING`.
- Non modificare la risorsa della policy di scalabilità associata ai gruppi con scalabilità automatica gestiti dai provider di capacità.
- Se il dimensionamento gestito è attivato quando crei un provider di capacità, puoi impostare il conteggio per il gruppo con scalabilità automatica desiderato su `0`. Quando il dimensionamento gestito è attivato, Amazon ECS gestisce le operazioni di riduzione e aumento del gruppo con scalabilità automatica.
- È necessario associare il provider di capacità a un cluster prima di associarlo alla strategia del fornitore di capacità.
- È possibile specificare un massimo di 20 provider di capacità per una strategia di provider di capacità.
- Non è possibile aggiornare un servizio che utilizza un provider di capacità di un gruppo con scalabilità automatica per utilizzare un provider di capacità Fargate. È vero anche il contrario.
- In una strategia del provider di capacità, se non viene specificato alcun valore di `weight` per un provider di capacità nella console, allora viene utilizzato il valore predefinito 1. Se si utilizza l'API o AWS CLI, `0` viene utilizzato il valore predefinito di.

- Quando più provider di capacità sono specificati nell'ambito di una strategia di provider di capacità, almeno uno dei provider deve avere un valore di peso maggiore di zero. I provider di capacità con un peso pari a zero non vengono utilizzati per collocare le attività. Se specifichi più provider di capacità in una strategia tutti con un peso pari a zero, allora qualsiasi operazione `RunTask` o `CreateService` che utilizza la strategia del provider di capacità avrà esito negativo.
- In una strategia di provider di capacità, solo un provider di capacità può avere un valore di base definito. Se non viene specificato alcun valore, viene utilizzato il valore predefinito zero.
- Un cluster può contenere una combinazione di provider di capacità del gruppo con scalabilità automatica e provider di capacità Fargate. Tuttavia, una strategia di provider di capacità può includere solo i provider di capacità del gruppo con scalabilità automatica o Fargate, ma non entrambi.
- Un cluster può contenere una combinazione di servizi e attività autonome che utilizzano sia i provider di capacità che i tipi di avvio. Un servizio può essere aggiornato per utilizzare una strategia del provider di capacità anziché un tipo di avvio. Tuttavia, quando si esegue questa operazione è necessario forzare una nuova implementazione.
- Amazon ECS supporta i warm pool Amazon EC2 Auto Scaling. Un warm pool è un gruppo di istanze Amazon EC2 pre-inizializzate pronte per essere messe in servizio. Ogni volta che l'applicazione deve essere scalata orizzontalmente, Amazon EC2 Auto Scaling utilizza le istanze preinizializzate del pool caldo anziché lanciare istanze fredde. Ciò consente l'esecuzione di qualsiasi processo di inizializzazione finale prima che l'istanza venga messa in servizio. Per ulteriori informazioni, consulta Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling.

Per ulteriori informazioni su come creare un modello di avvio per Dimensionamento automatico Amazon EC2, consulta Modelli di avvio nella Guida per l'utente di Dimensionamento automatico Amazon EC2. Per ulteriori informazioni su come creare un gruppo di Dimensionamento automatico Amazon EC2, consulta Gruppi con scalabilità automatica nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Considerazioni sulla sicurezza delle istanze di container Amazon EC2 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 container eseguiti da attività che utilizzano la modalità di rete `awsvpc` di accedere alle informazioni sulle credenziali fornite al profilo dell'istanza Amazon EC2, continuando allo stesso tempo a concedere le autorizzazioni fornite dal ruolo dell'attività, imposta la variabile di configurazione dell'agente `ECS_AWSVPC_BLOCK_IMDS` su true nel file di configurazione dell'agente e riavvia l'agente.
- Usa Amazon GuardDuty Runtime Monitoring per rilevare le minacce per cluster e contenitori all'interno del tuo AWS ambiente. Runtime Monitoring utilizza un agente di GuardDuty sicurezza che aggiunge visibilità di runtime ai singoli carichi di lavoro Amazon ECS, ad esempio accesso ai file, esecuzione dei processi e connessioni di rete. Per ulteriori informazioni, consulta GuardDutyRuntime Monitoring nella Guida per l'GuardDuty utente.

Creazione di un cluster Amazon ECS per il tipo di lancio Amazon EC2

Puoi creare un cluster Amazon ECS utilizzando la console. Prima di iniziare, accertati di aver completato le fasi in Configurazione per l'uso di Amazon ECS e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta the section called “Esempi di cluster Amazon ECS”. La console Amazon ECS offre un modo semplice per creare le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

Per rendere il processo di creazione del cluster il più semplice possibile, la console dispone di selezioni predefinite per molte scelte che descriviamo di seguito. Ci sono anche pannelli di aiuto disponibili per la maggior parte delle sezioni della console che forniscono ulteriore contesto.

Puoi registrare le istanze Amazon EC2 quando crei il cluster o registrare ulteriori istanze con il cluster dopo averlo 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

- Modifica lo spazio dei nomi predefinito associato al cluster.

  Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive. Lo spazio dei nomi predefinito ha lo stesso nome del cluster. Per ulteriori informazioni, consulta Interconnetti i servizi Amazon ECS.

- Attiva Container Insights.

  CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights”.

- Aggiungi tag per facilitare l'identificazione del cluster.

Opzioni del gruppo Auto Scaling

Quando utilizzi istanze Amazon EC2, devi specificare un gruppo Auto Scaling per gestire l'infrastruttura su cui vengono eseguiti i processi e i 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 le istanze di Amazon EC2 che contengono processi e si trovano in un gruppo Auto Scaling vengano terminate durante un'azione di riduzione orizzontale. 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 coppie di chiavi Amazon EC2 e istanze Linux nella Amazon EC2 User Guide.
- Il numero minimo di istanze da avviare per il gruppo Auto Scaling.

- Il numero massimo di istanze avviate per il gruppo Auto Scaling.

  Affinché il gruppo venga aumentato orizzontalmente, il massimo deve essere superiore a 0.

Amazon ECS crea un modello di avvio di Amazon EC2 Auto Scaling e un gruppo Auto Scaling per tuo conto come parte dello stack AWS CloudFormation . I valori specificati per l'AMI, i tipi di istanza e la coppia di chiavi SSH fanno parte del modello di avvio. Viene utilizzato il prefisso `EC2ContainerService-<ClusterName>` per identificare i modelli con facilità. I gruppi Auto Scaling hanno il prefisso `<ClusterName>-ECS-Infra-ECSAutoScalingGroup`.

Le istanze avviate per il gruppo Auto Scaling utilizzano il modello di avvio.

Opzioni di rete

Per impostazione predefinita, le istanze vengono avviate nelle sottoreti predefinite per la regione. Vengono utilizzati i gruppi di sicurezza, che controllano il traffico verso le istanze dei container, attualmente associati alle sottoreti. Puoi modificare le sottoreti e i gruppi di sicurezza per le istanze.

È possibile scegliere una sottorete esistente. Puoi utilizzare un gruppo di sicurezza esistente o crearne uno nuovo. Se si crea un nuovo gruppo di sicurezza, è necessario specificare almeno una regola in entrata.

Le regole in entrata determinano il traffico che può raggiungere le istanze di container e includono le seguenti proprietà:

- Il protocollo da consentire
- L'intervallo di porte da autorizzare
- Il traffico in entrata (fonte)

Per consentire il traffico in entrata da un indirizzo o da un blocco CIDR specifico, usa Personalizzato per Origine con il CIDR consentito.

Per consentire il traffico in entrata da tutte le destinazioni, specifica Ovunque per Origine. Questa opzione aggiunge automaticamente i blocchi CIDR 0.0.0.0/0 IPv4 e ::/0 IPv6.

Per consentire il traffico in entrata dal computer locale, usa Gruppo di origine per Origine. Questa operazione aggiunge automaticamente l'indirizzo IP attuale del computer locale come origine consentita.

Creazione di un nuovo cluster (console Amazon ECS)

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

1. Apri la console all'indirizzo 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 Configurazione del cluster, configura gli elementi seguenti:

   - In Nome cluster, inserisci un nome univoco.

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

   - (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. Se desideri aggiungere istanze Amazon EC2 al cluster, espandi Infrastruttura, deseleziona AWS Fargate (serverless), quindi seleziona Istanze Amazon EC2. 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 le istanze On-demand o le istanze Spot.
      - Se scegli di utilizzare le istanze Spot, per la strategia di allocazione scegli quali pool di capacità Spot (tipi di istanze e zone di disponibilità) vengono utilizzati per le istanze.

        Per la maggior parte dei carichi di lavoro, puoi scegliere Price capacity optimized.

        Per ulteriori informazioni, consulta Strategie di allocazione per Istanze spot nella Guida per l'utente di Amazon EC2.
      - Per Operating system/Architecture (Sistema operativo/architettura), scegli l'AMI ottimizzata per Amazon ECS per le istanze del gruppo Auto Scaling.

- In EC2 instance type (Tipo di istanza EC2), 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.

- Per il ruolo dell'istanza EC2, scegli un ruolo di istanza container esistente oppure puoi crearne uno nuovo.

  Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

- In Capacity (Capacità), inserisci il numero minimo e massimo di istanze da avviare nel gruppo Auto Scaling.
- In SSH key pair (Coppia di chiavi SSH), scegli la coppia che dimostra la tua identità quando ti connetti all'istanza.
- Per consentire immagini e spazio di archiviazione più grandi, per la dimensione del volume Root EBS, inserisci il valore in GiB.

7. (Facoltativo) Per modificare il VPC e le sottoreti, in Reti per le istanze Amazon EC2, 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 Sottoreti, seleziona le sottoreti.
   - Scegli i gruppi di sicurezza. In Gruppo di sicurezza, scegli una delle seguenti opzioni:
     - Per utilizzare un gruppo di sicurezza esistente, scegli Usa un gruppo di sicurezza esistente, quindi selezionalo.
     - Per creare un nuovo gruppo di sicurezza, scegli Crea un nuovo gruppo di sicurezza. Scegli quindi Aggiungi regola per ogni regola in entrata.

       Per informazioni sulle regole in entrata, consulta Opzioni di rete.
   - Per assegnare automaticamente gli indirizzi IP pubblici alle istanze di container Amazon EC2, in Assegna automaticamente 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) Per attivare Container Insights, espandi Monitoring (Monitoraggio) e poi attiva Use Container Insights (Usa Container Insights).

9. (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:

   - Per Key, inserisci **`guardDutyRuntimeMonitoringManaged`**
   - In Valore, specifica **`true`**.

10. (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.

11. Seleziona Crea.

Passaggi successivi

Dopo aver creato il cluster, è possibile creare le definizioni delle attività per le applicazioni e quindi eseguirle come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Definizioni dei processi di Amazon ECS
- Esecuzione di un'applicazione come attività Amazon ECS
- Creazione di un servizio Amazon ECS utilizzando la console

Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster

Amazon ECS è in grado di gestire la scalabilità delle istanze Amazon EC2 registrate nel cluster. Questa operazione viene definita dimensionamento automatico del cluster Amazon ECS. Attivi la scalabilità gestita quando crei il provider di capacità di gruppo Amazon ECS Auto Scaling. Quindi, si imposta una percentuale target (la`targetCapacity`) per l'utilizzo dell'istanza in questo gruppo Auto Scaling. Amazon ECS crea due CloudWatch metriche personalizzate e una politica di scalabilità

di tracciamento mirata per il tuo gruppo Auto Scaling. Amazon ECS gestisce quindi le azioni di scalabilità verticale e orizzontale in base all'utilizzo delle risorse utilizzato dalle attività.

Per ogni provider di capacità del gruppo con scalabilità automatica associato a un cluster, Amazon ECS crea e gestisce le seguenti risorse.

- Un CloudWatch allarme con un valore metrico basso
- Un allarme con valore metrico elevato CloudWatch
- Una policy di scalabilità di monitoraggio dei target

> **Note**
>
> Amazon ECS crea la policy di scalabilità di monitoraggio dei target e la collega al gruppo con scalabilità automatica. Per aggiornare la policy di scalabilità di monitoraggio delle destinazioni, aggiornare le impostazioni di scalabilità gestita della policy e non aggiornare direttamente la policy di scalabilità.

Quando disattivi la scalabilità gestita o dissocii il fornitore di capacità da un cluster, Amazon ECS rimuove sia i CloudWatch parametri che le risorse della policy di scalabilità di tracciamento di destinazione.

Amazon ECS utilizza i seguenti parametri per determinare le azioni da intraprendere:

`CapacityProviderReservation`

La percentuale di istanze di container utilizzate per uno specifico fornitore di capacità. Questo parametro è generato da Amazon ECS.

Amazon ECS imposta il valore `CapacityProviderReservation` su un numero compreso tra 0 e 100. Amazon ECS utilizza la seguente formula per rappresentare il rapporto della capacità rimanente nel gruppo con dimensionamento automatico. Quindi, Amazon ECS pubblica la metrica su. CloudWatch Per ulteriori informazioni su come viene calcolata la metrica, consulta Deep Dive on Amazon ECS Cluster Auto Scaling.

```
CapacityProviderReservation = (number of instances needed) / (number of running
 instances) x 100
```

`DesiredCapacity`

La quantità di capacità del gruppo con dimensionamento automatico. Questa metrica non è stata pubblicata. CloudWatch

Amazon ECS pubblica la `CapacityProviderReservation` metrica CloudWatch nel namespace. `AWS/ECS/ManagedScaling` Il parametro `CapacityProviderReservation` provoca una delle seguenti operazioni:

Il valore **`CapacityProviderReservation`** è uguale a **`targetCapacity`**

Il gruppo con dimensionamento automatico non ha bisogno di essere ridimensionato o di impiegare la scalabilità orizzontale. La percentuale di utilizzo prevista è stata raggiunta.

Il valore **`CapacityProviderReservation`** è maggiore di **`targetCapacity`**

Ci sono più attività che utilizzano una percentuale di capacità più elevata rispetto alla percentuale `targetCapacity`. L'aumento del valore della `CapacityProviderReservation` metrica provoca l'attivazione dell'allarme associato. CloudWatch Questo allarme aggiorna il valore di `DesiredCapacity` per il gruppo Auto Scaling. Il gruppo Auto Scaling utilizza questo valore per avviare le istanze EC2 e quindi registrarle con il 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' CloudWatch allarme associato. Questo allarme aggiorna il valore di `DesiredCapacity` per il gruppo Auto Scaling. Il gruppo Auto Scaling utilizza questo valore per terminare le istanze di container EC2 e quindi annullarne la registrazione con il cluster.

Il gruppo con dimensionamento automatico utilizza le policy di terminazione, per determinare quali istanze terminare per prime durante gli eventi di ridimensionamento. Inoltre, evita le istanze con l'impostazione di protezione da ridimensionamento delle istanze attivata. Il dimensionamento automatico del cluster può gestire le istanze con l'impostazione di protezione

per il ridimensionamento se la protezione da terminazione gestita è attivata. Per ulteriori informazioni sulla protezione da terminazione gestita, consulta Controlla le istanze terminate da Amazon ECS. Per ulteriori informazioni sul modo in cui i gruppi con dimensionamento automatico terminano le istanze, consulta Controllo delle istanze con dimensionamento automatico che vengono terminate durante il ridimensionamento nella Guida per l'utente di Dimensionamento automatico Amazon 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.
- Amazon ECS utilizza il ruolo IAM `AWSServiceRoleForECS` collegato al servizio per le autorizzazioni necessarie per richiamare AWS Auto Scaling per tuo conto. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.
- Quando utilizzi provider di capacità con gruppi con scalabilità automatica, l'utente, il gruppo e il ruolo che crea i provider di capacità richiedono l'autorizzazione `autoscaling:CreateOrUpdateTags`. Questo perché Amazon ECS aggiunge un tag al gruppo Auto Scaling quando lo associ al provider di capacità.

  > **Important**
  >
  > Assicurati che qualsiasi strumento utilizzato non rimuova il tag `AmazonECSManaged` dal gruppo con scalabilità automatica. Se questo tag viene rimosso, Amazon ECS non è in grado di gestire la scalabilità.

- La scalabilità automatica del cluster non modifica l'MinimumCapacityor MaximumCapacityper il gruppo. Affinché il gruppo venga scalato orizzontalmente, il valore di MaximumCapacitydeve essere maggiore di zero.
- Quando è attivata la scalabilità automatica (dimensionamento gestito), un provider di capacità può essere connesso solo a un cluster contemporaneamente. Se per il provider di capacità la scalabilità gestita è disattivata, è possibile associarlo a più cluster.
- Quando il dimensionamento gestito è disattivato, il provider di capacità non esegue operazioni di dimensionamento con riduzione o aumento orizzontale. In questo caso, è possibile utilizzare una strategia del provider di capacità per bilanciare le attività tra i provider di capacità.
- La `binpack` strategia è la strategia più efficiente in termini di capacità.

- Quando la capacità obiettivo è inferiore al 100%, la strategia di collocamento deve utilizzare la `binpack` strategia senza che la `spread` strategia abbia un ordine superiore rispetto alla `binpack` strategia. Ciò impedisce al provider di capacità di espandersi fino al raggiungimento del limite o fino al raggiungimento di un'istanza dedicata per ciascuna attività.

Ottimizza la scalabilità automatica dei cluster Amazon ECS

I clienti che eseguono Amazon ECS su Amazon EC2 possono sfruttare la scalabilità automatica del cluster per gestire la scalabilità dei gruppi Amazon EC2 Auto Scaling. Con Cluster Auto Scaling, puoi configurare Amazon ECS per scalare automaticamente il tuo gruppo Auto Scaling e concentrarti solo sull'esecuzione delle tue attività. Amazon ECS garantirà la scalabilità interna e orizzontale del gruppo Auto Scaling secondo necessità senza ulteriori interventi. I provider di capacità di Amazon ECS vengono utilizzati per gestire l'infrastruttura del cluster assicurando che ci siano abbastanza istanze di container per soddisfare le esigenze dell'applicazione. Per scoprire come funziona la scalabilità automatica dei cluster, consulta Deep Dive on Amazon ECS Cluster Auto Scaling.

la scalabilità automatica del cluster si basa su un'integrazione CloudWatch basata con il gruppo Auto Scaling per la regolazione della capacità del cluster. Pertanto presenta una latenza intrinseca associata alla pubblicazione delle CloudWatch metriche, il tempo impiegato dalla metrica per `CapacityProviderReservation` violare gli CloudWatch allarmi (sia alti che bassi) e il tempo impiegato dal riscaldamento di un'istanza Amazon EC2 appena lanciata. È possibile intraprendere le seguenti azioni per rendere la scalabilità automatica del cluster più reattiva per implementazioni più rapide:

Dimensioni di scalabilità graduale del provider di capacità

I provider di capacità di Amazon ECS alla fine aumenteranno o ridurranno le istanze dei container per soddisfare le esigenze della tua applicazione. Il numero minimo di istanze che Amazon ECS avvierà è impostato su 1 per impostazione predefinita. Ciò può aggiungere ulteriore tempo alle distribuzioni, se sono necessarie più istanze per collocare le attività in sospeso. Puoi aumentarlo `minimumScalingStepSize`tramite l'API Amazon ECS per aumentare il numero minimo di istanze scalabili da Amazon ECS alla volta. Un `maximumScalingStepSize`valore troppo basso può limitare il numero di istanze di container scalabili contemporaneamente, il che può rallentare le implementazioni.

> **Note**
>
> Questa configurazione è attualmente disponibile solo tramite le o API. `CreateCapacityProviderUpdateCapacityProvider`

Periodo di riscaldamento dell'istanza

Il periodo di riscaldamento dell'istanza è il periodo di tempo dopo il quale un'istanza Amazon EC2 appena lanciata può contribuire CloudWatch ai parametri per il gruppo Auto Scaling. Una volta scaduto il periodo di riscaldamento specificato, l'istanza viene conteggiata nelle metriche aggregate del gruppo Auto Scaling e l'autoscaling del cluster procede con la successiva iterazione di calcoli per stimare il numero di istanze richieste.

Il valore predefinito per `instanceWarmupPeriod`è 300 secondi, che è possibile configurare su un valore inferiore tramite le API o per un ridimensionamento più reattivo. `CreateCapacityProviderUpdateCapacityProvider`

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 immediatamente le istanze Amazon EC2 e attendere che si avviino prima di poter avviare contenitori su di esse. Ciò può ridurre in modo significativo la frequenza di avvio delle attività. Hai due opzioni qui.

In questo caso, la capacità di riserva di Amazon EC2 già avviata e pronta per l'esecuzione delle attività aumenterà la velocità effettiva di avvio delle attività. Puoi utilizzare la `Target Capacity` configurazione per indicare che desideri mantenere la capacità inutilizzata nei tuoi cluster. Ad esempio, `Target Capacity` impostando l'80%, si indica che il cluster necessita sempre del 20% di capacità di riserva. Questa capacità di riserva può consentire l'avvio immediato di qualsiasi attività autonoma, assicurando che l'avvio delle attività non venga limitato. Il compromesso di questo approccio è il potenziale aumento dei costi legati al mantenimento della capacità di riserva del cluster.

Un approccio alternativo che puoi prendere in considerazione è quello di aggiungere spazio di crescita al tuo servizio, non al fornitore di capacità. Ciò significa che invece di ridurre la `Target Capacity` configurazione per avviare la capacità di riserva, è possibile aumentare il numero di repliche nel servizio modificando la metrica di tracciamento della scalabilità di destinazione o le soglie di scalabilità graduale del servizio auto scaling. Tieni presente che questo approccio sarà utile solo per i carichi di lavoro con picchi di lavoro, ma non avrà alcun effetto quando distribuisci nuovi

servizi e passi da 0 a N attività per la prima volta. Per ulteriori informazioni sulle politiche di scalabilità correlate, consulta Target Tracking Scaling Policies o Step Scaling Policies nella Amazon Elastic Container Service Developer Guide.

Controlla le istanze terminate da Amazon ECS

> **Important**
>
> Devi attivare la protezione da ridimensionamento dell'istanza del dimensionamento automatico nel gruppo con dimensionamento automatico per utilizzare la funzionalità di protezione da terminazione gestita del dimensionamento automatico del cluster.

La protezione gestita dalla terminazione consente la scalabilità automatica del cluster per controllare quali istanze vengono terminate. Quando hai utilizzato la protezione gestita dalle terminazioni, Amazon ECS termina solo le istanze EC2 che non hanno alcuna attività Amazon ECS in esecuzione. 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 le istanze EC2 nel 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 riduzione orizzontale e disattiva la protezione da riduzione orizzontale per l'istanza EC2. Il gruppo Auto Scaling può quindi terminare l'istanza.

La protezione da ridimensionamento dell'istanza con dimensionamento automatico controlla quali istanze EC2 possono essere terminate tramite il dimensionamento automatico. Le istanze con la funzione di riduzione orizzontale attivata non possono essere terminate durante il processo di riduzione orizzontale. Per maggiori informazioni sulla protezione da ridimensionamento dell'istanza con dimensionamento automatico, consulta Utilizzo della protezione da ridimensionamento dell'istanza nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Puoi impostare la `targetCapacity` percentuale in modo da disporre di capacità inutilizzata. Ciò consente di avviare più rapidamente le attività future perché il gruppo Auto Scaling non deve avviare più istanze. Amazon ECS utilizza il valore della capacità target per gestire la CloudWatch metrica creata dal servizio. Amazon ECS gestisce la CloudWatch metrica. Il gruppo Auto Scaling viene trattato come uno stato stazionario in modo che non sia richiesta alcuna azione

di ridimensionamento. I valori possono essere compresi tra 0 e 100%. Ad esempio, per configurare Amazon ECS per mantenere una capacità libera del 10% oltre a quella utilizzata dalle attività Amazon ECS, imposta il valore della capacità target al 90%. Quando imposti il valore di `targetCapacity` su un provider di capacità, tieni presenti le considerazioni seguenti.

- Un valore di `targetCapacity` inferiore al 100% rappresenta la quantità di capacità libera (istanze Amazon EC2) che deve essere presente nel cluster. Capacità libera significa che non ci sono attività in esecuzione.
- I vincoli di posizionamento come le zone di disponibilità, senza ulteriori `binpack`, costringono Amazon ECS a eseguire infine un'attività per ogni istanza, comportamento che potrebbe non essere quello desiderato.

Devi attivare la protezione da ridimensionamento dell'istanza del dimensionamento automatico nel gruppo con dimensionamento automatico per utilizzare la protezione da terminazione gestita. Se non attivi la protezione da ridimensionamento, l'attivazione della protezione da terminazione gestita può portare a comportamenti indesiderati. Ad esempio, è possibile che le istanze siano bloccate in uno stato di svuotamento. Per ulteriori informazioni, consulta Utilizzo della protezione con riduzione orizzontale delle istanze nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Quando utilizzi la protezione da terminazione con un provider di capacità, non eseguire operazioni manuali, come lo scollegamento dell'istanza, sul gruppo con dimensionamento automatico associato al provider di capacità. Le azioni manuali possono interrompere il processo di ridimensionamento del provider di capacità. Se scolleghi un'istanza dal gruppo con dimensionamento automatico, devi anche annullare la registrazione dell'istanza scollegata dal cluster Amazon ECS.

Comportamento dell'aumento orizzontale gestito

Se disponi di fornitori di capacità di gruppo Auto Scaling che utilizzano la scalabilità gestita, Amazon ECS stima il numero ottimale di istanze da aggiungere al cluster e utilizza il valore per determinare quante istanze richiedere.

Amazon ECS seleziona un fornitore di capacità per ogni attività seguendo la strategia del fornitore di capacità dal servizio, dall'attività autonoma o dall'impostazione predefinita del cluster. Amazon ECS segue il resto di questi passaggi per un singolo provider di capacità.

Le attività senza una strategia di fornitore di capacità vengono ignorate dai fornitori di capacità. Un'attività in sospeso che non prevede una strategia del provider di capacità non comporterà l'impiego della scalabilità orizzontale di alcun provider di capacità. Le attività o i servizi non possono configurare una strategia del provider di capacità se tale attività o servizio imposta un tipo di avvio.

Di seguito viene descritto il comportamento dell'aumento orizzontale in modo più dettagliato.

- Raggruppa tutti i processi di provisioning per questo provider di capacità in modo che ogni gruppo abbia gli stessi requisiti di risorse esatti.
- Quando utilizzi più istanze in un gruppo con dimensionamento automatico, le istanze all'interno di tale gruppo vengono ordinate in base ai relativi parametri. Questi parametri includono vCPU, memoria, interfacce di rete elastiche (ENI), porte e GPU. Vengono selezionati i tipi di istanza più piccoli e più grandi per ciascun parametro. Per ulteriori informazioni su come scegliere il tipo di istanza, consultaIstanze di container Amazon EC2 per Amazon ECS.

  > **Important**
  >
  > Se un gruppo di attività ha requisiti di risorse superiori al tipo di istanza più piccolo del gruppo con dimensionamento automatico, quel gruppo di attività non può essere eseguito con questo provider di capacità. Il provider di capacità non dimensiona il gruppo con dimensionamento automatico. Le attività rimangono nello stato `PROVISIONING`.
  > Per evitare che le attività rimangano nello stato `PROVISIONING`, consigliamo di creare gruppi con dimensionamento automatico e provider di capacità separati per diversi requisiti minimi di risorse. Quando esegui attività o crei servizi, aggiungi solo provider di capacità alla strategia del provider di capacità in grado di eseguire l'attività sul tipo di istanza più piccolo del gruppo con dimensionamento automatico. Per altri parametri, puoi utilizzare i vincoli di posizionamento

- Per ogni gruppo di attività, Amazon ECS calcola il numero di istanze necessarie per eseguire le attività non posizionate. Questo calcolo utilizza una strategia `binpack`. Questa strategia tiene conto dei requisiti di vCPU, memoria, interfacce di rete elastiche (ENI), porte e GPU delle attività. Tiene conto anche della disponibilità di risorse delle istanze Amazon EC2. I valori per i tipi di istanza più grandi sono considerati quale numero massimo di istanze calcolato. I valori per il tipo di istanza più piccolo vengono utilizzati come protezione. Se il tipo di istanza più piccolo non può eseguire almeno un'istanza dell'attività, il calcolo considera l'attività come non compatibile. Di conseguenza, l'attività viene esclusa dal calcolo dell'aumento orizzontale. Quando tutte le attività non sono compatibili con il tipo di istanza più piccolo, il dimensionamento automatico del cluster si interrompe e il valore `CapacityProviderReservation` rimane `targetCapacity`.
- Amazon ECS pubblica la `CapacityProviderReservation` metrica in CloudWatch relazione al `minimumScalingStepSize` caso in cui si verifichi una delle seguenti condizioni.
  - Il numero massimo di istanze calcolato è inferiore alla dimensione minima del passaggio di scalabilità.

  - Il valore più basso del numero massimo di istanze calcolato `maximumScalingStepSize` o quello massimo.
- CloudWatch gli allarmi utilizzano la `CapacityProviderReservation` metrica per i fornitori di capacità. Quando il parametro `CapacityProviderReservation` è maggiore del valore di `targetCapacity`, gli allarmi aumentano anche la `DesiredCapacity` del gruppo con scalabilità automatica. Il `targetCapacity` valore è un'impostazione del provider di capacità che viene inviata all' CloudWatch allarme durante la fase di attivazione dell'auto scaling del cluster.

  L'impostazione predefinita `targetCapacity` è 100%.
- Il gruppo Auto Scaling avvia istanze EC2 aggiuntive. Per evitare l'over-provisioning, Auto Scaling assicura che la capacità delle istanze EC2 lanciate di recente sia stabilizzata prima del lancio di nuove istanze. La scalabilità automatica verifica se tutte le istanze esistenti hanno superato il `instanceWarmupPeriod` (ora meno il tempo di avvio dell'istanza). La scalabilità orizzontale è bloccata per le istanze che si trovano all'interno di. `instanceWarmupPeriod`

  Il tempo di default per il riscaldamento di un'istanza appena avviata è di 300 secondi.

Per maggiori informazioni, consulta Approfondimento sulla scalabilità automatica del cluster Amazon ECS.

Considerazioni sull'aumento orizzontale

Considera quanto segue per il processo di aumento orizzontale:

- Sebbene esistano più vincoli di collocamento, è consigliabile utilizzare solo il vincolo `distinctInstance` per la collocazione delle attività. Ciò impedisce l'arresto del processo di aumento orizzontale in seguito all'utilizzo di un vincolo di posizionamento non compatibile con le istanze campionate.
- La scalabilità gestita funziona meglio se il gruppo Auto Scaling utilizza tipi di istanza uguali o simili.
- Quando è necessario un processo di scalabilità orizzontale e non sono presenti istanze di container attualmente in esecuzione, inizialmente Amazon ECS impiega sempre la scalabilità orizzontale fino a due istanze, quindi esegue processi di scalabilità orizzontale o ridimensionamento aggiuntivi. Qualsiasi ulteriore impiego della scalabilità orizzontale attende il periodo di preparazione dell'istanza. Per i processi di ridimensionamento, Amazon ECS attende 15 minuti dopo un processo di scalabilità orizzontale prima di avviare in qualsiasi momento i processi di ridimensionamento.

- La seconda fase di aumento orizzontale deve attendere fino allo scadere del `instanceWarmupPeriod`, che potrebbe influire sul limite di scalabilità complessivo. Se devi ridurre questo tempo, assicurati che `instanceWarmupPeriod` sia sufficientemente grande da consentire all'istanza EC2 di avviare e avviare l'agente Amazon ECS (che impedisce l'overprovisioning).
- Il dimensionamento automatico del cluster supporta la configurazione di avvio, i modelli di avvio e più tipi di istanze nel gruppo con dimensionamento automatico del provider di capacità. Puoi inoltre utilizzare la selezione del tipo di istanza basata su attributi senza molteplici tipi di istanze.
- Quando utilizzi un gruppo Auto Scaling con istanze on demando e più tipi di istanza o istanze Spot, posiziona i tipi di istanza più grandi più in alto nell'elenco di priorità e non specificare un peso. Al momento la specifica di un peso non è supportata. Per ulteriori informazioni, consulta Gruppi Auto Scaling con più tipi di istanze nella Guida per l'utente di AWS Auto Scaling .
- Amazon ECS avvierà quindi `minimumScalingStepSize`, se il conteggio massimo delle istanze calcolate è inferiore alla dimensione minima del passo di dimensionamento o il valore inferiore di `maximumScalingStepSize` o del valore massimo del conteggio delle istanze calcolato.
- Se un servizio Amazon ECS o `run-task` avvia un'attività e le istanze del container del Capacity Provider non dispongono di risorse sufficienti per avviare l'attività, Amazon ECS limita il numero di attività con questo stato per ogni cluster e impedisce che qualsiasi attività superi questo limite. Per ulteriori informazioni, consulta Quote del servizio.

Comportamento di riduzione orizzontale gestita

Amazon ECS monitora le istanze di container per ciascun provider di capacità all'interno di un cluster. Quando un'istanza di container non esegue alcuna attività, viene considerata vuota e Amazon ECS avvia il processo di ridimensionamento.

CloudWatch gli allarmi scale-in richiedono 15 punti dati (15 minuti) prima dell'avvio del processo di scalabilità per il gruppo Auto Scaling. Dopo l'avvio del processo di riduzione orizzontale, fino a quando Amazon ECS non ha bisogno di ridurre il numero di istanze di container registrate, il gruppo con scalabilità automatica imposta il valore `DesireCapacity` di modo che sia superiore a un'istanza e inferiore al 50% ogni minuto.

Quando Amazon ECS richiede un aumento orizzontale (quando `CapacityProviderReservation` è maggiore di 100) mentre è in corso un processo di riduzione orizzontale, il processo di riduzione orizzontale viene interrotto e ricomincerà daccapo, se necessario.

Di seguito viene descritto il comportamento del ridimensionamento in modo più dettagliato:

1. Amazon ECS calcola il numero di istanze di container vuote. Un'istanza di container è considerata vuota quando non sono in esecuzione attività daemon.
2. Amazon ECS imposta il valore `CapacityProviderReservation` su un numero compreso tra 0 e 100 che utilizza la seguente formula per rappresentare il rapporto tra la dimensione prevista del gruppo con dimensionamento automatico e la sua dimensione effettiva, espresso in percentuale. Quindi, Amazon ECS pubblica la metrica su. CloudWatch Per maggiori informazioni su come viene calcolato il parametro, consulta Approfondimento sul dimensionamento automatico del cluster Amazon ECS

   ```
   CapacityProviderReservation = (number of instances needed) / (number of running
    instances) x 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 utilizzi la cessazione gestita dal provider di capacità, il gruppo con scalabilità automatica seleziona le istanze EC2 utilizzando la policy di cessazione del gruppo con scalabilità automatica e termina le istanze fino a che il numero di istanze EC2 raggiunge la `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.

Attivazione della scalabilità automatica del cluster Amazon ECS

È possibile utilizzare il AWS CLI per attivare la scalabilità automatica del cluster.

Prima di iniziare, crea un gruppo con scalabilità automatica e un provider di capacità. Per ulteriori informazioni, consulta the section called “Fornitori di capacità per il tipo di lancio EC2”.

Per attivare la scalabilità automatica del cluster, si associa il provider di capacità al cluster, quindi si attiva la scalabilità automatica del cluster.

1. Esegui il comando `put-cluster-capacity-providers` per associare uno o più provider di capacità al cluster.

Per aggiungere i provider AWS Fargate di capacità, includi `FARGATE` i fornitori `FARGATE_SPOT` di capacità nella richiesta. Per ulteriori informazioni, consulta la sezione `put-cluster-capacity-providers` nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \
  --cluster ClusterName \
  --capacity-providers CapacityProviderName FARGATE FARGATE_SPOT \
  --default-capacity-provider-strategy capacityProvider=CapacityProvider,weight=1
```

Per aggiungere un gruppo Auto Scaling per il tipo di avvio EC2, includi il nome del gruppo Auto Scaling nella richiesta. Per ulteriori informazioni, consulta la sezione `put-cluster-capacity-providers` nella Documentazione di riferimento della AWS CLI .

```
aws ecs put-cluster-capacity-providers \
  --cluster ClusterName \
  --capacity-providers CapacityProviderName \
  --default-capacity-provider-strategy capacityProvider=CapacityProvider,weight=1
```

2. Utilizza il comando `describe-clusters` per verificare che l'associazione abbia avuto successo. Per ulteriori informazioni, consulta la sezione `describe-clusters` nella Documentazione di riferimento della AWS CLI .

```
aws ecs describe-clusters \
  --cluster ClusterName \
  --include ATTACHMENTS
```

3. Esegui il comando `update-capacity-provider` per attivare la scalabilità automatica gestita per il provider di capacità. Per ulteriori informazioni, consulta la sezione `update-capacity-provider` nella Documentazione di riferimento della AWS CLI .

```
aws ecs update-capacity-provider \
  --capacity-providers CapacityProviderName \
  --auto-scaling-group-provider managedScaling=ENABLED
```

Disattivazione della scalabilità automatica del cluster Amazon ECS

È possibile utilizzare il AWS CLI per disattivare la scalabilità automatica del cluster.

Per disattivare la scalabilità automatica del cluster per un cluster, è possibile disassociare il provider di capacità con scalabilità gestita attivato dal cluster o aggiornare il provider di capacità per disattivare la scalabilità gestita.

Dissociare il provider di capacità

Utilizza la procedura seguente per annullare l'associazione di un provider di capacità con un cluster.

1. Utilizza il comando `put-cluster-capacity-providers` per annullare l'associazione del provider di capacità del gruppo con scalabilità automatica con il cluster. Il cluster può mantenere l'associazione con i fornitori AWS Fargate di capacità. Per ulteriori informazioni, consulta la sezione `put-cluster-capacity-providers` nella Documentazione di riferimento della AWS CLI .

   ```
   aws ecs put-cluster-capacity-providers \
     --cluster ClusterName \
     --capacity-providers FARGATE FARGATE_SPOT \
     --default-capacity-provider-strategy '[]'
   ```

   Utilizza il comando `put-cluster-capacity-providers` per annullare l'associazione del provider di capacità del gruppo con scalabilità automatica con il cluster. Per ulteriori informazioni, consulta la sezione `put-cluster-capacity-providers` nella Documentazione di riferimento della AWS CLI .

   ```
   aws ecs put-cluster-capacity-providers \
     --cluster ClusterName \
     --capacity-providers [] \
     --default-capacity-provider-strategy '[]'
   ```

2. Utilizza il comando `describe-clusters` per verificare che l'annullamento dell'associazione abbia avuto successo. Per ulteriori informazioni, consulta la sezione `describe-clusters` nella Documentazione di riferimento della AWS CLI .

   ```
   aws ecs describe-clusters \
     --cluster ClusterName \
     --include ATTACHMENTS
   ```

Disattivare la scalabilità gestita per il provider di capacità

Utilizza la procedura seguente per disattivare la scalabilità gestita per il provider di capacità.

- Utilizza il comando `update-capacity-provider` per attivare la scalabilità automatica gestita per il provider di capacità. Per ulteriori informazioni, consulta la sezione `update-capacity-provider` nella Documentazione di riferimento della AWS CLI .

```
aws ecs update-capacity-provider \
  --capacity-providers CapacityProviderName \
  --auto-scaling-group-provider managedScaling=DISABLED
```

Blocca in sicurezza i carichi di lavoro Amazon ECS in esecuzione su istanze EC2

Il drenaggio gestito delle istanze facilita la chiusura graduale delle istanze Amazon EC2. Ciò consente ai carichi di lavoro di interrompersi in sicurezza e di essere riprogrammati su istanze non terminanti. La manutenzione e gli aggiornamenti dell'infrastruttura vengono eseguiti senza preoccuparsi dell'interruzione dei carichi di lavoro. Utilizzando il drenaggio gestito delle istanze, semplifichi i flussi di lavoro di gestione dell'infrastruttura che richiedono la sostituzione delle istanze Amazon EC2, garantendo al contempo la resilienza e la disponibilità delle applicazioni.

Il drenaggio gestito delle istanze di Amazon ECS funziona con le istanze di gruppo Auto Scaling sostitutive. In base all'aggiornamento dell'istanza e alla durata massima dell'istanza, i clienti possono garantire la conformità ai più recenti requisiti di sistema operativo e di sicurezza relativi alla loro capacità.

Il drenaggio gestito delle istanze può essere utilizzato solo con i provider di capacità di Amazon ECS. Puoi attivare il drenaggio gestito delle istanze quando crei o aggiorni i fornitori di capacità del gruppo Auto Scaling utilizzando la console AWS CLI Amazon ECS o l'SDK.

I seguenti eventi sono coperti dal drenaggio delle istanze gestite di Amazon ECS.

- Aggiornamento delle istanze di gruppo Auto Scaling - Utilizza l'aggiornamento dell'istanza per eseguire la sostituzione progressiva delle istanze Amazon EC2 nel gruppo Auto Scaling anziché eseguirla manualmente in batch. Ciò è utile quando è necessario sostituire un gran numero di istanze. L'aggiornamento dell'istanza viene avviato tramite la console Amazon EC2 o l'API. `StartInstanceRefresh` Assicurati di selezionare `Replace` la protezione Scale-in quando chiami `StartInstanceRefresh` se utilizzi la protezione gestita dalla terminazione.

- Durata massima dell'istanza - È possibile definire una durata massima quando si tratta di sostituire le istanze del gruppo Auto Scaling. Ciò è utile per pianificare le istanze sostitutive in base a politiche di sicurezza interne o alla conformità.
- Scale-in del gruppo Auto Scaling - Basato su politiche di scalabilità e azioni di scalabilità pianificate, il gruppo Auto Scaling supporta il ridimensionamento automatico delle istanze. Utilizzando un gruppo Auto Scaling come fornitore di capacità Amazon ECS, puoi scalare le istanze di gruppo Auto Scaling quando non sono in esecuzione alcuna attività.
- Controlli di integrità del gruppo Auto Scaling - Il gruppo Auto Scaling supporta molti controlli di integrità per gestire la chiusura delle istanze non integre.
- AWS CloudFormation aggiornamenti dello stack - È possibile aggiungere un `UpdatePolicy` attributo allo AWS CloudFormation stack per eseguire aggiornamenti continui quando il gruppo cambia.
- Ribilanciamento della capacità Spot - Il gruppo Auto Scaling cerca di sostituire in modo proattivo le istanze Spot che presentano un rischio di interruzione più elevato sulla base dell'avviso di ribilanciamento della capacità di Amazon EC2. Il gruppo Auto Scaling termina la vecchia istanza quando la variante sostitutiva viene lanciata ed è funzionante. Il drenaggio dell'istanza gestita di Amazon ECS prosciuga l'istanza Spot nello stesso modo in cui drena un'istanza non Spot.
- Interruzione in loco - Le istanze Spot vengono chiuse con un preavviso di due minuti. Il drenaggio delle istanze gestito da Amazon ECS mette l'istanza in stato di drenaggio in risposta.

Agganci al ciclo di vita di Amazon EC2 Auto Scaling con drenaggio gestito delle istanze

Gli hook del ciclo di vita del gruppo Auto Scaling consentono al cliente di creare soluzioni attivate da determinati eventi nel ciclo di vita dell'istanza ed eseguire un'azione personalizzata quando si verifica quel determinato evento. Un gruppo Auto Scaling consente fino a 50 hook. Possono esistere più hook di terminazione che vengono eseguiti in parallelo e il gruppo Auto Scaling attende il completamento di tutti gli hook prima di terminare un'istanza.

Oltre alla terminazione degli hook gestita da Amazon ECS, puoi anche configurare i tuoi hook di terminazione del ciclo di vita. Gli hook del ciclo di vita dispongono di un `default action` hook e consigliamo di impostarlo `continue` come predefinito per garantire che altri hook, come l'hook gestito di Amazon ECS, non siano influenzati da errori causati dagli hook personalizzati.

Se hai già configurato un hook del ciclo di vita di terminazione del gruppo Auto Scaling e hai anche abilitato il drenaggio delle istanze gestite di Amazon ECS, vengono eseguiti entrambi gli hook del ciclo di vita. Le tempistiche relative, tuttavia, non sono garantite. I Lifecycle Hook hanno un'`default`

`action`impostazione per specificare l'azione da intraprendere allo scadere del timeout. In caso di errori, consigliamo di utilizzare `continue` come risultato predefinito l'hook personalizzato. Ciò garantisce che altri hook, in particolare gli hook gestiti di Amazon ECS, non siano influenzati da errori nel tuo hook personalizzato del ciclo di vita. Il risultato alternativo di `abandon` fa sì che tutti gli altri hook vengano ignorati e dovrebbe essere evitato. Per ulteriori informazioni sugli hook del ciclo di vita del gruppo Auto Scaling, consulta gli hook del ciclo di vita di Amazon EC2 Auto Scaling nella Amazon EC2 Auto Scaling User Guide.

Attività e drenaggio gestito delle istanze

Il drenaggio gestito delle istanze di Amazon ECS utilizza la funzionalità di drenaggio esistente presente nelle istanze di container. La funzionalità di drenaggio delle istanze del contenitore esegue la sostituzione e l'interruzione delle attività di replica che appartengono a un servizio Amazon ECS. Un'attività autonoma, come quella richiamata da`RunTask`, che si trova nello stato or rimane inalterata`PENDING`. `RUNNING` È necessario attendere che vengano completate o interromperle manualmente. L'istanza del contenitore rimane nello `DRAINING` stato finché tutte le attività non vengono interrotte o non sono trascorse 48 ore. Le attività daemon sono le ultime a interrompersi dopo l'interruzione di tutte le attività di replica.

Drenaggio gestito delle istanze e protezione gestita dalla terminazione

Il drenaggio gestito delle istanze funziona anche se la terminazione gestita è disabilitata. Per informazioni sulla protezione gestita dalla terminazione, vedere. Controlla le istanze terminate da Amazon ECS

La tabella seguente riassume il comportamento delle diverse combinazioni di terminazione gestita e drenaggio gestito.

Terminazione gestita	Drenaggio gestito	Outcome
Abilitato	Abilitato	Amazon ECS protegge le istanze Amazon EC2 che eseguono attività dall'inte

Terminazione gestita	Drenaggio gestito	Outcome
		rruzione a causa di eventi di scalabilità. Tutte le istanze in fase di chiusura, ad esempio quelle per cui non è impostata la protezione dalla terminazione, che hanno subito un'interruzione Spot o sono state forzate dall'aggiornamento dell'istanza, vengono esaurite senza problemi.

Terminazione gestita	Drenaggio gestito	Outcome
Disabilitato	Abilitato	Amazon ECS non protegge le istanze Amazon EC2 che eseguono attività dalla scalabilità. Tuttavia, tutte le istanze che vengono terminate vengono svuotate correttamente.

Terminazione gestita	Drenaggio gestito	Outcome
Abilitato	Disabilitato	Amazon ECS protegge le istanze Amazon EC2 che eseguono attività dall'interruzione a causa di eventi di scalabilità. Tuttavia, le istanze possono comunque essere interrotte dall'interruzione di Spot o dall'aggiornamento forzato dell'istanza o se non eseguono alcuna attività. Amazon ECS non esegue operazion

Terminazione gestita	Drenaggio gestito	Outcome
		i di drenaggio regolari per queste istanze e avvia attività di assistenza sostitutive dopo che queste si sono interrotte.

Terminazione gestita	Drenaggio gestito	Outcome
Disabilitato	Disabilitato	Le istanze Amazon EC2 possono essere scalate o terminate in qualsiasi momento, anche se eseguono attività di Amazon ECS. Amazon ECS avvierà attività di assistenza sostitutive una volta interrotte.

Drenaggio gestito delle istanze e drenaggio delle istanze Spot

Con il drenaggio delle istanze Spot, puoi impostare una variabile di ambiente `ECS_ENABLE_SPOT_INSTANCE_DRAINING` sull'agente Amazon ECS che consente ad Amazon ECS di mettere un'istanza nello stato di drenaggio in risposta all'interruzione Spot di due minuti. Il drenaggio gestito delle istanze di Amazon ECS facilita l'arresto graduale delle istanze Amazon EC2 in fase di chiusura per molte ragioni, non solo per un'interruzione Spot. Ad esempio, puoi utilizzare il ribilanciamento della capacità di Amazon EC2 Auto Scaling per sostituire in modo proattivo l'istanza Spot con un rischio elevato di interruzione, mentre il drenaggio gestito delle istanze esegue l'arresto graduale dell'istanza Spot in fase di sostituzione. Quando si utilizza il drenaggio gestito delle istanze, non è necessario abilitare il drenaggio delle istanze Spot separatamente,

quindi `ECS_ENABLE_SPOT_INSTANCE_DRAINING` nel gruppo Auto Scaling i dati degli utenti sono ridondanti. Per ulteriori informazioni sul drenaggio delle istanze Spot, consulta. Spot Instances

Come funziona il drenaggio gestito delle istanze con EventBridge

Gli eventi di drenaggio delle istanze gestite di Amazon ECS vengono pubblicati su Amazon EventBridge e Amazon ECS crea una regola EventBridge gestita nel bus predefinito del tuo account per supportare il drenaggio gestito delle istanze. Puoi filtrare questi eventi su altri AWS servizi come Lambda, Amazon SNS e Amazon SQS per monitorare e risolvere i problemi.

- Amazon EC2 Auto Scaling invia un evento quando viene richiamato un hook EventBridge 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 e le API di Amazon ECS.
- 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 fornitori di capacità del gruppo Auto Scaling utilizzando la console Amazon ECS e. AWS CLI

> **Note**
>
> Il drenaggio gestito delle istanze è attivo per impostazione predefinita quando crei un provider di capacità.

Di seguito sono riportati alcuni esempi di utilizzo di AWS CLI per creare un provider di capacità con il drenaggio gestito delle istanze abilitate e di abilitazione del drenaggio gestito delle istanze per il provider di capacità esistente di un cluster.

Crea un provider di capacità con il drenaggio gestito delle istanze abilitato

Per creare un provider di capacità con il drenaggio gestito delle istanze abilitate, usa il `create-capacity-provider` comando. Imposta il parametro `managedDraining` su `ENABLED`.

```
aws ecs create-capacity-provider \
--name capacity-provider \
```

```
--auto-scaling-group-provider '{
  "autoScalingGroupArn": "asg-arn",
  "managedScaling": {
    "status": "ENABLED",
    "targetCapacity": 100,
    "minimumScalingStepSize": 1,
    "maximumScalingStepSize": 1
  },
  "managedDraining": "ENABLED",
  "managedTerminationProtection": "ENABLED",
}'
```

Risposta:

```
{
    "capacityProvider": {
        "capacityProviderArn": "capacity-provider-arn",
        "name": "capacity-provider",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "asg-arn",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 1
            },
            "managedTerminationProtection": "ENABLED"
            "managedDraining": "ENABLED"
        }
    }
}
```

Abilita il drenaggio gestito delle istanze per il provider di capacità esistente di un cluster

Abilita il drenaggio gestito delle istanze per il provider di capacità esistente di un cluster utilizza il `update-capacity-provider` comando. Vedete che `managedDraining` attualmente dice `DISABLED` e `updateStatus` dice`UPDATE_IN_PROGRESS`.

```
aws ecs update-capacity-provider \
--name cp-draining \
--auto-scaling-group-provider '{
  "managedDraining": "ENABLED"
```

```
}
```

Risposta:

```
{
    "capacityProvider": {
        "capacityProviderArn": "cp-draining-arn",
        "name": "cp-draining",
        "status": "ACTIVE",
        "autoScalingGroupProvider": {
            "autoScalingGroupArn": "asg-draining-arn",
            "managedScaling": {
                "status": "ENABLED",
                "targetCapacity": 100,
                "minimumScalingStepSize": 1,
                "maximumScalingStepSize": 1,
                "instanceWarmupPeriod": 300
            },
            "managedTerminationProtection": "DISABLED",
            "managedDraining": "DISABLED" // before update
        },
        "updateStatus": "UPDATE_IN_PROGRESS", // in progress and need describe again to
 find out the result
        "tags": [
        ]
    }
}
```

Usa il `describe-clusters` comando e includi`ATTACHMENTS`. L'allegato `status` di drenaggio dell'istanza gestita è`PRECREATED`, e il complesso `attachmentsStatus` è`UPDATING`.

```
aws ecs describe-clusters --clusters cluster-name --include ATTACHMENTS
```

Risposta:

```
{
    "clusters": [
        {
            ...

            "capacityProviders": [
```

```
                "cp-draining"
            ],
            "defaultCapacityProviderStrategy": [],
            "attachments": [
                # new precreated managed draining attachment
                {
                    "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                    "type": "managed_draining",
                    "status": "PRECREATED",
                    "details": [
                        {
                            "name": "capacityProviderName",
                            "value": "cp-draining"
                        },
                        {
                            "name": "autoScalingLifecycleHookName",
                            "value": "ecs-managed-draining-termination-hook"
                        }
                    ]
                },

                ...

            ],
            "attachmentsStatus": "UPDATING"
        }
    ],
    "failures": []
}
```

Quando l'aggiornamento è terminato`describe-capacity-providers`, usa e lo vedi `managedDraining` ora`ENABLED`.

```
aws ecs describe-capacity-providers --capacity-providers cp-draining
```

Risposta:

```
{
    "capacityProviders": [
        {
            "capacityProviderArn": "cp-draining-arn",
            "name": "cp-draining",
            "status": "ACTIVE",
```

```
            "autoScalingGroupProvider": {
                "autoScalingGroupArn": "asg-draning-arn",
                "managedScaling": {
                    "status": "ENABLED",
                    "targetCapacity": 100,
                    "minimumScalingStepSize": 1,
                    "maximumScalingStepSize": 1,
                    "instanceWarmupPeriod": 300
                },
                "managedTerminationProtection": "DISABLED",
                "managedDraining": "ENABLED" // successfully update
            },
            "updateStatus": "UPDATE_COMPLETE",
            "tags": []
        }
    ]
}
```

Risoluzione dei problemi relativi al drenaggio delle istanze di Amazon ECS Managed

Potrebbe essere necessario risolvere i problemi relativi al drenaggio gestito delle istanze. Di seguito è riportato un esempio di un problema e di una risoluzione che potresti riscontrare durante l'utilizzo.

Le istanze non terminano dopo aver superato la durata massima dell'istanza quando si utilizza la scalabilità automatica.

Se le istanze non terminano nemmeno dopo aver raggiunto e superato la durata massima dell'istanza durante l'utilizzo di un gruppo con scalabilità automatica, è possibile che siano protette dalla scalabilità in entrata. Puoi disattivare la terminazione gestita e consentire il drenaggio gestito per gestire il riciclo delle istanze.

Creazione di risorse per la scalabilità automatica del cluster Amazon ECS utilizzando AWS Management Console

Scopri come creare le risorse per la scalabilità automatica del AWS Management Console cluster utilizzando. Laddove le risorse richiedono un nome, utilizziamo il prefisso `ConsoleTutorial` per assicurarci che tutte abbiano un nome univoco e per renderle facili da individuare.

Argomenti

- Prerequisiti
- Fase 1: Creazione di un cluster Amazon ECS

- Fase 2: Registrazione di una definizione di attività
- Fase 3: esecuzione di un'attività
- Fase 4: verifica
- Fase 5: rimozione

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.
- Il tuo AWS utente dispone delle autorizzazioni richieste specificate nell'esempio di policy AmazonECS_ FullAccess IAM.
- Viene creato il ruolo IAM dell'istanza di container Amazon ECS. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.
- Viene creato il ruolo IAM collegato ai servizi Amazon ECS. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.
- Viene creato il ruolo IAM collegato al servizio Auto Scaling. Per ulteriori informazioni, consulta Ruoli collegati ai servizi per Amazon EC2 Auto Scaling nella Guida per l'utente di Amazon EC2 Auto Scaling.
- 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 lancio di Amazon EC2 Auto Scaling e un gruppo Auto Scaling per tuo conto come parte dello stack. AWS CloudFormation

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel riquadro di navigazione scegli Cluster, quindi Crea cluster.
3. In Configurazione del cluster, per Nome cluster, inserisci `ConsoleTutorial-cluster`.
4. In Infrastruttura, AWS deseleziona 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.
  - Per Tipo di istanza EC2, seleziona 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. Seleziona 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 all'indirizzo 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 Editor JSON, incolla i seguenti contenuti.

```
{
    "family": "ConsoleTutorial-taskdef",
    "containerDefinitions": [
        {
```

```
            "name": "sleep",
            "image": "amazonlinux:2",
            "memory": 20,
            "essential": true,
            "command": [
                "sh",
                "-c",
                "sleep infinity"
            ]
        }
    ],
    "requiresCompatibilities": [
        "EC2"
    ]
}
```

5. Scegli 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella pagina Clusters, scegli -cluster. ConsoleTutorial
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 Crea.

Fase 4: verifica

A questo punto del tutorial dovresti disporre di un cluster con cinque attività in esecuzione e un gruppo con dimensionamento automatico con un provider di capacità. Il provider di capacità ha il dimensionamento gestito di Amazon ECS abilitato.

Possiamo verificare che tutto funzioni correttamente visualizzando le CloudWatch metriche, le impostazioni del gruppo Auto Scaling e infine il conteggio delle attività del cluster Amazon ECS.

Per visualizzare le CloudWatch metriche per il tuo cluster

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
3. Nel pannello di navigazione, in Parametri, scegli Tutti i parametri.
4. Nella pagina Tutti i parametri, all'interno della scheda Sfoglia, scegli `AWS/ECS/ManagedScaling`.
5. Scegli CapacityProvideril nome, 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 console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
2. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
3. Nel pannello di navigazione, nella sezione Dimensionamento automatico, seleziona Gruppi con dimensionamento automatico.
4. Scegli il gruppo con dimensionamento automatico `ConsoleTutorial-cluster` creato in questo tutorial. Visualizza il valore in Capacità desiderata e visualizza le istanze nella scheda Gestione delle istanze per confermare che il gruppo è stato dimensionato a due istanze.

Utilizza la procedura seguente per visualizzare il cluster Amazon ECS per verificare che le istanze Amazon EC2 siano state registrate con il cluster e che i processi siano passati allo stato `RUNNING`.

Per verificare le istanze nel gruppo Auto Scaling

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster `ConsoleTutorial-cluster`.
4. Nella scheda Attività verifica che siano visualizzate cinque attività nello stato `RUNNING`.

Fase 5: rimozione

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi. L'eliminazione dei provider di capacità e delle definizioni di attività non è supportata, ma non sono associati costi a queste risorse.

Per eliminare le risorse del tutorial

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters, scegli ConsoleTutorial-cluster.
4. Nella pagina ConsoleTutorial-cluster, scegli la scheda Attività, quindi scegli Stop, Arresta tutto.
5. Nel pannello di navigazione scegliere Clusters (Cluster).
6. Nella pagina Clusters, scegli ConsoleTutorial -cluster.
7. In alto a destra della pagina, scegli Elimina cluster.
8. Nella casella di conferma, inserisci delete ConsoleTutorial-cluster e scegli Elimina.
9. Elimina i gruppi Auto Scaling completando la seguente procedura.
    a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
    b. Nella barra di navigazione nella parte superiore della schermata selezionare la regione .
    c. Nel pannello di navigazione, nella sezione Dimensionamento automatico, seleziona Gruppi con dimensionamento automatico.
    d. Seleziona il gruppo con dimensionamento automatico `ConsoleTutorial-cluster`, quindi scegli Operazioni.
    e. Nel menu Actions (Operazioni) selezionare Delete (Elimina). Nella casella di conferma, immetti elimina e scegli Elimina.

Istanze di container Amazon EC2 per Amazon ECS

Un'istanza di container Amazon ECS è un'istanza Amazon EC2 che esegue l'agente container Amazon ECS ed è registrata in un cluster. Quando esegui le attività con Amazon ECS utilizzando il tipo di avvio EC2, esterno o un provider di capacità del gruppo con dimensionamento automatico, le attività vengono posizionate nelle istanze di container attive. Sei responsabile della gestione e della manutenzione delle istanze di container.

Sebbene tu possa creare la tua AMI di istanza Amazon EC2 che soddisfi le specifiche di base necessarie per eseguire i carichi di lavoro containerizzati su Amazon ECS, le AMI ottimizzate per Amazon ECS sono preconfigurate e testate su Amazon ECS da ingegneri. AWS È il modo più semplice per iniziare ed eseguire i container su AWS rapidamente.

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando si utilizza AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di avvio di Amazon EC2 per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisci lo stack, venga verificato se è disponibile un aggiornamento da applicare alle istanze EC2. Per un esempio di come utilizzare il parametro Systems Manager, consulta Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS nella Guida per AWS CloudFormation l'utente.

- Recupero di metadati AMI Linux ottimizzati per Amazon ECS
- Recupero di metadati AMI ottimizzati per Amazon ECS Bottlerocket
- Recupero di metadati AMI Windows ottimizzati per Amazon ECS

Puoi scegliere tra i tipi di istanza compatibili con la tua applicazione. Con istanze più grandi, puoi avviare più attività contemporaneamente. Con istanze più piccole, puoi scalare orizzontalmente in modo più dettagliato per risparmiare sui costi. Non è necessario scegliere un unico tipo di istanza Amazon EC2 che si adatti a tutte le applicazioni del cluster. È invece possibile creare più gruppi di Auto Scaling in cui ogni gruppo ha un tipo di istanza diverso. Quindi, puoi creare un provider di capacità Amazon EC2 per ognuno di questi gruppi.

Utilizza le seguenti linee guida per determinare i tipi di famiglie di istanze e il tipo di istanza da utilizzare:

- Elimina i tipi o le famiglie di istanze che non soddisfano i requisiti specifici della tua applicazione. Ad esempio, se l'applicazione richiede una GPU, puoi escludere tutti i tipi di istanza che non dispongono di una GPU.
- Prendi in considerazione i requisiti, tra cui la velocità di trasmissione e lo storage di rete.
- Considerate la CPU e la memoria. Come regola generale, la CPU e la memoria devono essere sufficientemente grandi da contenere almeno una replica dell'attività che si desidera eseguire.

Spot Instances

La capacità spot può offrire risparmi significativi sui costi rispetto alle istanze on demand. La capacità spot è una capacità in eccesso a un prezzo notevolmente inferiore rispetto a quella on demand o riservata. La capacità spot è adatta per carichi di lavoro di elaborazione in batch e machine learning ambienti di sviluppo e staging. Più in generale, è adatto a qualsiasi carico di lavoro che tollera tempi di inattività temporanei.

Tieni presente le seguenti conseguenze, poiché la capacità spot potrebbe non essere sempre disponibile.

- Nei periodi con domanda estremamente elevata, la capacità spot potrebbe non essere disponibile. Ciò può causare ritardi nell'avvio delle istanze spot di Amazon EC2. In questi eventi, i servizi Amazon ECS riprovano ad avviare le attività e anche i gruppi con dimensionamento automatico Amazon EC2 riprovano ad avviare le istanze, finché la capacità richiesta non diventa disponibile. Amazon EC2 non sostituisce la capacità spot con quella 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.

- Utilizza più tipi di istanze Amazon EC2: quando utilizzi policy a istanze miste con Dimensionamento automatico Amazon EC2, nel tuo gruppo con dimensionamento automatico vengono avviati 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, consultare la sezione relativa ai Gruppi con dimensionamento automatico con più tipi di istanze e opzioni di acquisto nella Guida per l'utente di Dimensionamento automatico Amazon EC2.
- 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 ottimizzata per la capacità al momento dell'avvio di una nuova istanza, Amazon EC2 Spot seleziona il tipo di istanza con la massima disponibilità nella zona di disponibilità selezionata. Ciò aiuta a ridurre la possibilità che l'istanza venga terminata subito dopo l'avvio.

Per informazioni su come configurare gli avvisi di terminazione spot sulle istanze di container, consulta:

- Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot
- Configurazione delle istanze di container Amazon ECS Windows per ricevere avvisi sulle istanze Spot

AMI Linux ottimizzate per Amazon ECS

Amazon ECS fornisce le AMI ottimizzate per Amazon ECS preconfigurate con i requisiti e i consigli per eseguire i carichi di lavoro dei container. Consigliamo di utilizzare l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS per le istanze di Amazon EC2, a meno che l'applicazione non richieda istanze Amazon EC2 basate sulla GPU, un determinato sistema operativo o una determinata versione di Docker non ancora disponibile in quell'AMI. Per informazioni sulle istanze Amazon Linux 2 e Amazon Linux 2023, consulta la sezione Confronto tra Amazon Linux 2 e Amazon Linux 2023 nella Guida per l'utente di Amazon Linux 2023. L'avvio delle istanze di container dall'AMI ottimizzata per Amazon ECS più recente ti assicura di ricevere gli aggiornamenti di sicurezza e la versione dell'agente del container corrente. Per ulteriori informazioni sull'avvio di un'istanza, consulta Avvio di un'istanza di container Linux di Amazon ECS.

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando si utilizza AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di avvio di Amazon EC2 per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisci lo stack, venga verificato se è disponibile un aggiornamento da applicare alle istanze EC2. Per un esempio di come utilizzare il parametro Systems Manager, consulta Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS nella Guida per AWS CloudFormation l'utente.

Se devi personalizzare l'AMI ottimizzata per Amazon ECS, consulta Amazon ECS Optimized AMI Build Recipies on. GitHub

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 come base. Sono disponibili anche le note di rilascio di AMI di Amazon Linux 2. Per ulteriori informazioni, consulta Note di rilascio di Amazon Linux 2.

Si consiglia di utilizzare un'AMI con Linux kernel 5.10 perché il kernel Linux 4.14 è stato raggiunto il 10 end-of-life gennaio 2024.

Le seguenti varianti dell'AMI ottimizzata per Amazon ECS sono disponibili per le istanze Amazon EC2.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2023	AMI Amazon Linux 2023 ottimizzata per Amazon ECS	Amazon Linux 2023 è la nuova generazione di Amazon Linux di AWS. Nella maggior parte dei casi, è consigliata per l'avvio delle istanze Amazon EC2 per i carichi di lavoro Amazon ECS. Per ulteriori informazioni, consulta la sezione Che cos'è	Di default, l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS è fornita con un singolo volume root di 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio per aumentare lo storage disponibi

Sistema operativo	AMI	Descrizione	Configurazione dello storage
		Amazon Linux 2023 nella Guida per l'utente di Amazon Linux 2023.	le nell'istanza di container. Questo storage è utilizzat o per il sistema operativo e per le immagini Docker e i metadati. Il file system di default per l'AMI Amazon Linux 2023 ottimizza ta per Amazon ECS è `xfs` e Docker utilizza il driver di archiviaz ione `overlay2`. Per ulteriori informazi oni, consulta Use the OverlayFS storage driver nella documentazione Docker.
Amazon Linux 2023 (arm64)	AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS	L'uso di questa AMI basata su Amazon Linux 2023 è consiglia to quando si avviano istanze Amazon EC2, che sono alimentate da processori AWS Graviton/Graviton 2 basati su Arm, per i carichi di lavoro Amazon ECS. Per ulteriori informazi oni, consulta General Purpose Instances nella Guida per l'utente di Amazon EC2. L'AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2023 (Neuron)	Amazon Linux 2023 (Neuron)	Basata su Amazon Linux 2023, questa AMI è per istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfigurato con i driver AWS Inferenti a e AWS Trainium e il runtime AWS Neuron per Docker, che semplifica l'esecuzi one di carichi di lavoro di inferenza di machine learning su Amazon ECS. Per ulteriori informazioni, consulta Definizio ni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron. L'AMI Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2	AMI Amazon Linux 2 kernel 5.10 ottimizzata per Amazon ECS	Quest'AMI basata su Amazon Linux 2 deve essere utilizzata quando si avviano istanze Amazon EC2 e si desidera utilizzare Linux kernel 5.10 anziché kernel 4.14 per i 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.	Di default, le AMI ottimizzate per Amazon ECS basate su Amazon Linux 2 (AMI, Amazon Linux 2 ottimizzate per Amazon ECS, AMI Amazon Linux 2 (arm64) ottimizzate per Amazon ECS e AMI Amazon ECS ottimizzate per GPU) vengono fornite con un singolo volume radice da 30 GiB. Puoi modificare le dimensioni del volume root di 30 GiB al momento dell'avvio 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 `xfs` e Docker utilizza
	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.	
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 istanze Amazon EC2, che sono alimentate da processori AWS Graviton/Graviton 2 basati su ARM, e	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
		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 General Purpose Instances nella Guida per l'utente di Amazon EC2. L'AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	il driver di archiviazione `overlay2`. Per ulteriori informazioni, consulta Use the OverlayFS storage driver nella documentazione Docker.

Sistema operativo	AMI	Descrizione	Configurazione dello storage
	AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS	Basata su Amazon Linux 2, questa AMI è destinata all'avvio di istanze Amazon EC2, alimentate da processori Graviton/ Graviton 2 AWS basati su ARM, per i carichi di lavoro Amazon ECS. L'AMI Amazon Linux 2 (arm64) ottimizza ta per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (GPU)	AMI kernel 5.10 ottimizzata per GPU Amazon ECS	Basata su Amazon Linux 2, questa AMI è consigliata per l'avvio di istanze basate su GPU Amazon EC2 con kernel Linux 5.10 per i carichi di lavoro Amazon ECS. Viene fornita pre-configurato con driver del kernel NVIDIA e un runtime della GPU Docker che costituisce i carichi di lavoro in esecuzion e che sfruttano i vantaggi delle GPU su Amazon ECS. Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU.	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
	AMI Amazon ECS ottimizzata per GPU	Basata su Amazon Linux 2, questa AMI è consigliata per l'avvio di istanze basate su GPU Amazon EC2 con kernel Linux 4.14 per i carichi di lavoro Amazon ECS. Viene fornita pre-configurato con driver del kernel NVIDIA e un runtime della GPU Docker che costituisce i carichi di lavoro in esecuzion e che sfruttano i vantaggi delle GPU su Amazon ECS. Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU.	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
Amazon Linux 2 (Neuron)	AMI Amazon ECS per il kernel 5.10 di Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS	Quest'AMI basata su Amazon Linux 2 deve essere utilizzata per le istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfigurato con AWS Inferentia con kernel Linux 5.10 e driver AWS Train ium 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 Definizio ni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron. L'AMI Amazon Linux 2 (Neuron) ottimizza ta per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	

Sistema operativo	AMI	Descrizione	Configurazione dello storage
	AMI Amazon Linux 2 (Neuron) ottimizzata per Amazon ECS	Quest'AMI basata su Amazon Linux 2 deve essere utilizzata per le istanze Amazon EC2 Inf1, Trn1 o Inf2. Viene preconfig urato con i driver AWS Inferentia e AWS Trainium e il runtime AWS Neuron per Docker, che semplifica l'esecuzi one di carichi di lavoro di inferenza di machine learning su Amazon ECS. Per ulteriori informazioni, consulta Definizio ni delle attività di Amazon ECS per i carichi di lavoro di machine learning di AWS Neuron. L'AMI Amazon Linux 2 (Neuron) ottimizza ta per Amazon ECS non viene fornita con quella preinstallata. AWS CLI	

Amazon ECS fornisce un changelog per la variante Linux dell'AMI ottimizzata per Amazon ECS su. GitHub Per ulteriori informazioni, consulta Changelog.

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 o Amazon Linux 2023 come base. Puoi recuperare il nome dell'AMI di origine di Amazon Linux 2 o Amazon Linux 2023 per ogni variante eseguendo query sull'API Archivio dei parametri Systems Manager. Per ulteriori informazioni, consulta Recupero di metadati AMI Linux ottimizzati per Amazon ECS. Sono disponibili anche le note di rilascio di AMI di Amazon Linux 2. Per ulteriori informazioni, consulta Note di rilascio di Amazon Linux 2. Sono disponibili anche le note di rilascio di Amazon Linux 2023. Per ulteriori informazioni, consulta Note di rilascio di Amazon Linux 2023.

Nelle pagine seguenti vengono fornite ulteriori informazioni sulle modifiche:

- Note di rilascio di Source AMI su GitHub
- Note di rilascio di Docker Engine nella documentazione Docker
- Documentazione dei driver NVIDIA nella documentazione NVIDIA
- Registro delle modifiche dell'agente Amazon ECS attivo GitHub

  Il codice sorgente dell'applicazione `ecs-init`, gli script e la configurazione per il pacchetto dell'agente fanno ora parte del repository dell'agente. Per le versioni `ecs-init` e i pacchetti precedenti di Amazon ecs-init, consulta il changelog di Amazon ecs-init su GitHub

Applicazione degli aggiornamenti di sicurezza all'AMI ottimizzata per Amazon ECS

Le AMI ottimizzate per Amazon ECS basate su Amazon Linux contengono una versione personalizzata di. cloud-init Cloud-initè 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, tutte le AMI ottimizzate per Amazon ECS basate su Amazon Linux rilasciate prima del 12 giugno 2024 hanno tutti gli aggiornamenti di sicurezza «critici» e «importanti» applicati all'avvio dell'istanza.

A partire dalle versioni del 12 giugno 2024 delle AMI ottimizzate per Amazon ECS basate su Amazon Linux 2, il comportamento predefinito non includerà più l'aggiornamento dei pacchetti al momento del lancio. Ti consigliamo invece di eseguire l'aggiornamento a una nuova AMI ottimizzata per Amazon ECS non appena le versioni vengono rese disponibili. Le AMI ottimizzate per Amazon ECS vengono rilasciate quando sono disponibili aggiornamenti di sicurezza o modifiche alle AMI di base. In questo modo avrai la certezza di ricevere le versioni più recenti dei pacchetti e gli aggiornamenti di sicurezza e che le versioni dei pacchetti siano immutabili al momento del lancio delle istanze. Per ulteriori informazioni sul recupero dell'ultima AMI ottimizzata per Amazon ECS, consulta. Recupero di metadati AMI Linux ottimizzati per Amazon ECS

Ti consigliamo di automatizzare l'ambiente per l'aggiornamento a una nuova AMI non appena viene resa disponibile. Per informazioni sulle opzioni disponibili, consulta Amazon ECS consente una gestione più semplice della capacità EC2, 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 istanza Amazon EC2.

```
yum update --security
```

Se desideri riattivare gli aggiornamenti di sicurezza all'avvio, puoi aggiungere la seguente riga alla `#cloud-config` sezione dei dati utente cloud-init all'avvio dell'istanza Amazon EC2. Per ulteriori informazioni, consulta Using cloud-init su Amazon Linux 2 nella Amazon Linux User Guide.

```
#cloud-config
repo_upgrade: security
```

Recupero di metadati AMI Linux ottimizzati per Amazon ECS

Puoi recuperare in modo programmatico i metadati AMI ottimizzati per Amazon ECS. I metadati includono il nome AMI, la versione dell'agente container Amazon ECS e la versione runtime di Amazon ECS che include la versione Docker.

Quando crei un cluster utilizzando la console, Amazon ECS crea un modello di avvio per le tue istanze con l'AMI più recente associata al sistema operativo selezionato.

Quando si utilizza AWS CloudFormation per creare un cluster, il parametro SSM fa parte del modello di avvio di Amazon EC2 per le istanze del gruppo Auto Scaling. Puoi configurare il modello per utilizzare un parametro dinamico di Systems Manager per determinare quale AMI Amazon ECS Optimized distribuire. Questo parametro garantisce che ogni volta che distribuisci lo stack, venga verificato se è disponibile un aggiornamento da applicare alle istanze EC2. Per un esempio di come utilizzare il parametro Systems Manager, consulta Creare un cluster Amazon ECS con l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS nella Guida per AWS CloudFormation l'utente.

L'ID dell'AMI, il nome dell'immagine, il sistema operativo, la versione dell'agente del container, il nome dell'immagine di origine e la versione di runtime per ogni variante delle AMI ottimizzate per Amazon ECS possono essere recuperati a livello di programmazione eseguendo una query sull'API dell'archivio parametri di Systems Manager. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere GetParameterse GetParametersByPath.

> **Note**
>
> Per recuperare i metadati dell'AMI ottimizzata per Amazon ECS, l'utente di amministrazione deve disporre delle seguenti autorizzazioni IAM. Queste autorizzazioni sono state aggiunte alla policy IAM `AmazonECS_FullAccess`.
>
> - ssm: GetParameters
> - ssm: GetParameter
> - ssm: GetParameters ByPath

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.

AMI Amazon Linux ottimizzate per Amazon ECS

- 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 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/amazal2023neuronamion-linux-2/arm64/<version>
```

- Metadati AMI Amazon Linux 2 (arm64) kernel 5.10:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/arm64/<version>
```

- Metadati AMI del kernel 5.10 ottimizzati per GPU Amazon ECS:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/gpu/<version>
```

- Metadati AMI Amazon Linux 2 (GPU):

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/<version>
```

- Metadati AMI Amazon Linux 2 (Neuron) del kernel 5.10 ottimizzati per Amazon ECS:

```
/aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/inf/<version>
```

- Metadati dell'AMI Amazon Linux 2 (Neuron):

```
/aws/service/ecs/optimized-ami/amazon-linux-2/inf/<version>
```

Il formato dei nomi di parametro seguente recupera l'ID immagine dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS tramite 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.l-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'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.

AMI Amazon Linux ottimizzate per Amazon ECS

- Per le AMI Amazon Linux 2023 ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/
recommended --region us-east-1
```

- Per le AMI Amazon Linux 2023 (arm64) ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/
arm64/recommended --region us-east-1
```

- Per le AMI Amazon Linux 2023 (Neuron) ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2023/
neuron/recommended --region us-east-1
```

- Per le AMI Amazon Linux 2 kernel 5.10 ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/recommended --region us-east-1
```

- Per le AMI Amazon Linux 2 ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
recommended --region us-east-1
```

- Per le AMI Amazon Linux 2 (arm64) kernel 5.10 ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/arm64/recommended --region us-east-1
```

- Per le AMI Amazon Linux 2 (arm64) ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazal2023neuronamion-
linux-2/arm64/recommended --region us-east-1
```

- Per le AMI del kernel 5.10 ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/gpu/recommended --region us-east-1
```

- Per le AMI Amazon ECS ottimizzate per GPU:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended --region us-east-1
```

- Per le AMI del kernel 5.10 Amazon Linux 2 (Neuron) ottimizzate per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/inf/recommended --region us-east-1
```

- Per le AMI Amazon Linux 2 (Neuron) ottimizzate per Amazon ECS:

```
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 GetParametersByPath Manager utilizzando AWS CLI il comando seguente.

```
aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/ --region us-east-1
```

Recupero dell'ID immagine dell'ultima AMI Amazon Linux 2 kernel 5.10 consigliata e ottimizzata per Amazon ECS

Puoi recuperare l'ID immagine dell'ultimo ID AMI Amazon Linux Linux 2 kernel 5.10 consigliato per Amazon ECS utilizzando il sottoparametro. `image_id`

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
kernel-5.10/recommended/image_id --region us-east-1
```

Per recuperare solo il valore `image_id`, è possibile eseguire query sul valore di parametro specifico, ad esempio:

```
aws ssm get-parameters --names /aws/service/ecs/optimized-ami/amazon-linux-2/
recommended/image_id --region us-east-1 --query "Parameters[0].Value"
```

Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello AWS CloudFormation facendo riferimento al nome dell'archivio parametri di Systems Manager.

Esempio per Linux

```
Parameters:kernel-5.10
  LatestECSOptimizedAMI:
    Description: AMI ID
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ecs/optimized-ami/amazon-linux-2/kernel-5.10/recommended/
image_id
```

Script di build per AMI Linux ottimizzata per Amazon ECS

Amazon ECS ha reso open-source gli script della build che vengono utilizzati per creare le varianti Linux dell'AMI ottimizzata per Amazon ECS. Questi script di build sono ora disponibili su. GitHub Per ulteriori informazioni, consulta amazon-ecs-ami su. GitHub

Se devi personalizzare l'AMI ottimizzata per Amazon ECS, consulta Amazon ECS Optimized AMI Build Recipies 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 alle nostre AMI. 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 aws/amazon-ecs-ami su. GitHub

Come creare un'AMI Linux ottimizzata per Amazon ECS

1. `aws/amazon-ecs-ami` GitHub Clona il repository.

   ```
   git clone https://github.com/aws/amazon-ecs-ami.git
   ```

2. Aggiungi una variabile di ambiente per la AWS regione da utilizzare durante la creazione dell'AMI. Sostituisci il valore `us-west-2` con la regione da utilizzare.

```
export REGION=us-west-2
```

3. Viene fornito un Makefile per creare l'AMI. Dalla directory principale del repository clonato, utilizza uno dei seguenti comandi, corrispondente alla variante Linux dell'AMI ottimizzata per Amazon ECS che desideri creare.

   - AMI Amazon Linux 2 ottimizzata per Amazon ECS

     ```
     make al2
     ```

   - AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS

     ```
     make al2arm
     ```

   - AMI Amazon ECS ottimizzata per GPU

     ```
     make al2gpu
     ```

   - 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 Amazon Linux 2023 (Neuron) ottimizzata per Amazon ECS

     ```
     make al2023neu
     ```

AMI Bottlerocket ottimizzate per Amazon ECS

Bottlerocketè 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 varianti dell'BottlerocketAMI ottimizzata per Amazon ECS per kernel 6.1 e kernel 5.10.

Le seguenti varianti utilizzano il kernel 6.1:

- `aws-ecs-2`
- `aws-ecs-2-nvidia`

Le seguenti varianti utilizzano il kernel 5.1.10:

- `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 le istanze Amazon EC2 con processori `x86_64` e `arm64`. L'AMI Bottlerocket non è consigliata per l'uso con istanze Amazon EC2 dotate di chip Inferentia.
- Le immagini Bottlerocket non includono un server SSH o uno shell (interprete di comandi). Tuttavia, è possibile utilizzare gli 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 container gestisce l'agente AWS Systems Manager che è possibile utilizzare per eseguire comandi o avviare sessioni di shell sulle istanze Bottlerocket di Amazon EC2. Per ulteriori informazioni, consulta Configurazione di Session Manager nella Guida per l'utente di AWS Systems Manager .

- Bottlerocket è ottimizzato per i carichi di lavoro dei container e si concentra sulla sicurezza. Bottlerocket non include un gestore di pacchetti ed è immutabile. Per informazioni sulle funzionalità e sulle linee guida di sicurezza, vedere Funzionalità di sicurezza e Linee guida sulla sicurezza su. GitHub
- La modalità di rete `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'`initProcessEnabled`attività è supportato per la versione AMI Bottlerocket `1.19.0` o successiva.
- Le AMI Bottlerocket inoltre non supportano i servizi e le funzionalità seguenti:
  - ECS Anywhere
  - Service Connect
  - Amazon EFS in modalità crittografata e modalità di rete `awsvpc`
  - Acceleratore di inferenze elastiche

Recupero di metadati AMI ottimizzati per Amazon ECS Bottlerocket

Puoi recuperare l'ID Amazon Machine Image (AMI) per le AMI ottimizzate per Amazon ECS interrogando l'API Parameter Store. AWS Systems Manager Con questo parametro, non è necessario eseguire una ricerca manuale degli ID dell'AMI ottimizzata per Amazon ECS. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere GetParameter. Il principale IAM che utilizzi deve disporre dell'autorizzazione IAM `ssm:GetParameter` per recuperare i metadati dell'AMI ottimizzata per Amazon ECS.

`aws-ecs-2`Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2` Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima 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
    ```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro indicato. Nel seguente URL, sostituisci *`region`* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta Ricerca di un'AMI attiva GitHub.
  - Per un'architettura a 64 bit (`x86_64`):

    ```
    https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/aws-ecs-2/x86_64/latest/image_id/description?region=region#
    ```

- Per l'architettura a 64 bit Arm (`arm64`):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/
aws-ecs-2/arm64/latest/image_id/description?region=region#
```

`aws-ecs-2-nvidia`Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-2-nvdia` Bottlerocket per regione e architettura con AWS CLI o. AWS Management Console

- 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
```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro indicato. Nel seguente URL, sostituisci *`region`* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta Ricerca di un'AMI attiva GitHub.
  - Per l'architettura a 64 bit (`x86_64`):

```
https://regionconsole.aws.amazon.com/systems-manager/parameters/aws/service/
bottlerocket/aws-ecs-2-nvidia/x86_64/latest/image_id/description?region=region#
```

  - Per l'architettura a 64 bit Arm (`arm64`):

```
https://regionconsole.aws.amazon.com/systems-manager/parameters/aws/service/
bottlerocket/aws-ecs-2-nvidia/arm64/latest/image_id/description?region=region#
```

`aws-ecs-1`Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-1` Bottlerocket Regione AWS tramite un'architettura con o AWS CLI il. AWS Management Console

- AWS CLI— Puoi recuperare l'ID immagine dell'ultima 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
    ```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro indicato. Nel seguente URL, sostituisci *`region`* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta Ricerca di un'AMI attiva GitHub.
  - Per un'architettura a 64 bit (`x86_64`):

    ```
    https://region.console.aws.amazon.com/systems-manager/parameters/aws/service/
    bottlerocket/aws-ecs-1/x86_64/latest/image_id/description
    ```

  - Per l'architettura a 64 bit Arm (`arm64`):

```
https://region.console.aws.amazon.com/systems-manager/parameters/aws/service/
bottlerocket/aws-ecs-1/arm64/latest/image_id/description
```

`aws-ecs-1-nvidia`Variante AMI Bottlerocket

Puoi recuperare l'ultima variante stabile dell'AMI `aws-ecs-1-nvdia` Bottlerocket per regione e architettura con AWS CLI o. AWS Management Console

- 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-nvidia/x86_64/latest/image_id" --query Parameter.Value --output text
    ```

  - Per l'architettura a 64 bit Arm (`arm64`):

    ```
    aws ssm get-parameter --region us-east-1 --name "/aws/service/bottlerocket/aws-
    ecs-1-nvidia/arm64/latest/image_id" --query Parameter.Value --output text
    ```

- AWS Management Console: puoi eseguire una query per l'ID dell'AMI ottimizzata consigliata per Amazon ECS utilizzando un URL nella AWS Management Console. L'URL apre la console Amazon EC2 Systems Manager con il valore dell'ID per il parametro indicato. Nel seguente URL, sostituisci *region* con il codice della regione per cui desideri ottenere l'ID AMI. Per informazioni sulle funzionalità supportate Regioni AWS, consulta Ricerca di un'AMI attiva GitHub.
  - Per l'architettura a 64 bit (`x86_64`):

    ```
    https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/
    aws-ecs-1-nvidia/x86_64/latest/image_id/description?region=region#
    ```

  - Per l'architettura a 64 bit Arm (`arm64`):

```
https://console.aws.amazon.com/systems-manager/parameters/aws/service/bottlerocket/
aws-ecs-1-nvidia/arm64/latest/image_id/description?region=region#
```

Passaggi successivi

Per un tutorial dettagliato su come iniziare a usare 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 informazioni su come avviare un'istanza Bottlerocket, consulta Avvio di un'Bottlerocketistanza per Amazon ECS

Avvio di un'Bottlerocketistanza per Amazon ECS

Puoi avviare un'Bottlerocketistanza in modo da poter eseguire i carichi di lavoro dei container.

Puoi usare il AWS CLI per avviare l'Bottlerocketistanza.

1. Crea un file denominato `userdata.toml`. Questo file viene utilizzato per i dati utente dell'istanza. Sostituire *`cluster-name`* con il nome del cluster.

   ```
   [settings.ecs]
   cluster = "cluster-name"
   ```

2. Usa uno dei comandi inclusi in the section called “Recupero di metadati AMI ottimizzati per Amazon ECS Bottlerocket” 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:

   - Sostituisci la *`sottorete`* con l'ID della sottorete privata o pubblica in cui verrà avviata l'istanza.
   - Sostituisci *`bottlerocket_ami`* con l'ID dell'AMI ottenuto nella fase precedente.
   - Sostituisci *`t3.large`* con il tipo di istanza da utilizzare.
   - Sostituisci *`regione`* con il codice della regione.

   ```
   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 *my-cluster* con il nome del cluster.
   - Sostituisci *regione* con il codice della Regione.

```
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 AMI 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 istanze EC2 per i tuoi carichi di lavoro Amazon ECS, sei responsabile della manutenzione delle istanze

Procedure di gestione

- Avvio di un'istanza di container Linux di Amazon ECS
- Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati
- Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot
- Esecuzione di uno script all'avvio di un'istanza di container Amazon ECS Linux
- Aumento delle interfacce di rete di istanze di container Amazon ECS Linux
- Riservare la memoria delle istanze del contenitore Amazon ECS Linux
- Gestione remota delle istanze di container Amazon ECS tramite AWS Systems Manager
- Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux
- Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling

- Aggiornamento dell'agente del container Amazon ECS

Ogni versione dell'agente del container di Amazon ECS supporta una serie di funzioni differenti e assicura le correzioni dei bug delle versioni precedenti. Quando possibile, consigliamo sempre di utilizzare la versione più recente dell'agente del container di Amazon ECS. Per passare all'ultima versione dell'agente del container, consulta Aggiornamento dell'agente del container Amazon ECS.

Per vedere le caratteristiche e i miglioramenti inclusi in ogni versione dell'agente, consulta https://github.com/aws/amazon-ecs-agent/releases.

> **Important**
>
> La versione Docker minima per parametri affidabili è `v20.10.13` e successive, inclusa nell'AMI `20220607` ottimizzata per Amazon ECS e versioni successive.
> Le versioni dell'agente Amazon ECS `1.20.0` e successive non supportano più le versioni di Docker precedenti alla `1.9.0`.

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 console Amazon EC2 e l' AWS CLI 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.

È possibile utilizzare la nuova procedura guidata Amazon EC2 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 console Amazon EC2 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. Dal pannello di controllo della console Amazon EC2, scegli Launch Instance (Avvia istanza).

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 AMI ottimizzate per Amazon ECS più recenti e i relativi valori, consulta AMI Amazon Linux ottimizzata per Amazon ECS.
2. Premere Invio.
3. Nella pagina Scegliere un'Amazon Machine Image (AMI), selezionare la categoria AWS AMI del 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 indicati in Istanze Linux.

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 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

Configurare le impostazioni di rete, se necessario.

- Networking platform Piattaforma di rete): scegli Virtual Private Cloud (VPC), quindi specifica la sottorete nella sezione Network interfaces (Interfacce di rete).
- VPC: selezionare un VPC esistente in cui creare il gruppo di sicurezza.
- Sottorete: è possibile avviare un'istanza in una sottorete associata a una zona di disponibilità, una Local Zone, una zona Wavelength o un Outpost.

  Per avviare l'istanza in una zona di disponibilità, selezionare la sottorete in cui avviare l'istanza. Per creare una nuova sottorete, scegliere Create new subnet (Crea nuova sottorete) per passare alla console Amazon VPC. Al termine, tornare alla procedura guidata di avvio istanza e scegliere Refresh (Aggiorna) per caricare la sottorete nell'elenco.

  Per avviare l'istanza in una Local Zone, selezionare una sottorete creata nella Local Zone.

  Per avviare un'istanza in un Outpost, selezionare una sottorete in un VPC associato a un Outpost.
- Auto-assign Public IP (IP pubblico di assegnazione automatica): se l'istanza deve essere accessibile da Internet, verifica che il campo Auto-assign Public IP (Assegna automaticamente IP pubblico) sia impostato su Enable (Abilita). In caso contrario, imposta il campo su Disable (Disabilita).

> **Note**
>
> Le istanze di container richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite istanze di container con indirizzi IP pubblici.
> Per ulteriori informazioni sugli endpoint VPC di interfaccia, vedi Endpoint VPC dell'interfaccia di Amazon ECS (AWS PrivateLink)
> Se non disponi di un endpoint VPC di interfaccia configurato e le istanze di container non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta NAT gateways (Gateway NAT) nella Guida per l'utente di Amazon VPC e Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux in questa guida.

- Firewall (security groups) (Firewall [gruppi di sicurezza]): utilizza un gruppo di sicurezza per definire le regole del firewall per l'istanza di container. Tali regole specificano quale traffico di rete in entrata viene distribuito sull'istanza di container. Tutto il traffico rimanente verrà ignorato.
  - Per selezionare un gruppo di sicurezza esistente, scegli Select an existing security group (Seleziona un gruppo di sicurezza esistente), quindi seleziona il gruppo di sicurezza creato in Configurazione per l'uso di Amazon ECS.

Per configurare l'archiviazione

L'AMI selezionata include uno o più volumi di archiviazione, compreso il volume dispositivo root. È possibile specificare altri volumi da collegare all'istanza.

Puoi utilizzare la vista Semplice.

- Storage Type (Tipo di storage): configura lo storage per l'istanza di container.

  Se stai utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS, l'istanza dispone di un singolo volume di 30 GiB configurato che è condiviso tra il sistema operativo e Docker.

  Se stai utilizzando l'AMI ottimizzata per Amazon ECS, l'istanza dispone di due volumi configurati. Il volume Root è per il sistema operativo, mentre il secondo volume di Amazon EBS (collegato a `/dev/xvdcz`) è per l'utilizzo di Docker.

  Puoi aumentare o diminuire le dimensioni del volume per l'istanza in modo che soddisfi le esigenze applicative.

Dettagli avanzati

Per Advanced Details (Dettagli avanzati), espandi la sezione per visualizzare i campi e specifica eventuali parametri aggiuntivi per l'istanza.

- Purchasing option (Opzioni di acquisto): seleziona Request Spot Instances (Richiedi istanze Spot) per avviare un'istanza Spot. Dovrai anche impostare altri campi correlati alle istanze Spot. Per ulteriori informazioni, consulta Richiesta Istanza Spot.

> **Note**
>
> Se utilizzi istanze Spot e visualizzi un messaggio `Not available`, potresti dover scegliere un altro tipo di istanza.

.

- Profilo istanza IAM: seleziona il ruolo IAM dell'istanza di container. Questo di solito è chiamato `ecsInstanceRole`.

> **Important**
>
> Se non avvii l'istanza di container con le autorizzazioni IAM corrette, l'agente Amazon ECS non potrà connettersi al cluster. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

- (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 di dati utente di Amazon EC2 vengono eseguiti solo una volta, al primo avvio dell'istanza. Di seguito sono elencati esempi comuni dei dati utente utilizzati per:
  - Di default, l'istanza di container si avvia nel cluster predefinito. Per avviarla in un cluster non predefinito, scegli l'elenco Advanced Details (Dettagli avanzati). Quindi, incolla lo script seguente nel campo User data (Dati utente), sostituendo *`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, sostituendo *`your_bucket_name con il nome del bucket`* per installarlo AWS CLI e scrivi il file di configurazione al momento del lancio.

> **Note**
>
> Per ulteriori informazioni su questa configurazione, consulta Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3.

```
#!/bin/bash
yum install -y aws-cli
aws s3 cp s3://your_bucket_name/ecs.config /etc/ecs/ecs.config
```

- Specificare i tag per l'istanza di container utilizzando il parametro di configurazione `ECS_CONTAINER_INSTANCE_TAGS`. Ciò crea i tag associati solo a Amazon ECS, non possono essere elencati utilizzando l'API Amazon EC2.

  > **Important**
  >
  > Se avvii le istanze del container utilizzando un gruppo Auto Scaling di Amazon EC2, devi utilizzare 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 istanze Amazon EC2 avviate utilizzando i gruppi Auto Scaling.

  ```
  #!/bin/bash
  cat <<'EOF' >> /etc/ecs/ecs.config
  ECS_CLUSTER=your_cluster_name
  ECS_CONTAINER_INSTANCE_TAGS={"tag_key": "tag_value"}
  EOF
  ```

- Specifica i tag per l'istanza di container e quindi utilizza il parametro di configurazione `ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM` per propagarli da Amazon EC2 ad Amazon ECS.

  Di seguito è riportato un esempio di script di dati utente che propaga i tag associati a un'istanza di container, nonché registra l'istanza di container con un cluster denominato `your_cluster_name`:

  ```
  #!/bin/bash
  cat <<'EOF' >> /etc/ecs/ecs.config
  ECS_CLUSTER=your_cluster_name
  ECS_CONTAINER_INSTANCE_PROPAGATE_TAGS_FROM=ec2_instance
  EOF
  ```

Per ulteriori informazioni, consulta Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati.

Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati

Quando avvii un'istanza Amazon EC2, puoi passare i dati utente all'istanza EC2. 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, ad esempio hook di avvio del 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 quando si utilizza la procedura guidata di avvio di Amazon EC2, consulta. Avvio di un'istanza di container Linux di Amazon ECS

Puoi configurare l'istanza del contenitore per passare i dati nella configurazione dell'agente del contenitore o nella configurazione del demone Docker.

Agente del container Amazon ECS

Le varianti Linux dell'AMI ottimizzata per Amazon ECS cercano i dati di configurazione dell'agente nel file `/etc/ecs/ecs.config` all'avvio dell'agente del container. Puoi specificare questi dati di configurazione al momento dell'avvio con i dati utente di Amazon EC2. 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

È possibile specificare le informazioni di configurazione del daemon Docker con i dati utente Amazon EC2. Per ulteriori informazioni sulle opzioni di configurazione, consulta la documentazione del daemon Docker.

Nell'esempio riportato di seguito, le opzioni personalizzate vengono aggiunte al file di configurazione del daemon Docker, `/etc/docker/daemon.json`, che viene quindi specificato nei dati utente all'avvio dell'istanza.

```
#!/bin/bash
cat <<EOF >/etc/docker/daemon.json
{"debug": true}
EOF
systemctl restart docker --no-block
```

Nell'esempio riportato di seguito, le opzioni personalizzate vengono aggiunte al file di configurazione del daemon Docker, `/etc/docker/daemon.json`, che viene quindi specificato nei dati utente all'avvio dell'istanza. Questo esempio mostra come disattivare il docker-proxy nel file di configurazione del daemon Docker.

```
#!/bin/bash
cat <<EOF >/etc/docker/daemon.json
{"userland-proxy": false}
EOF
systemctl restart docker --no-block
```

Configurazione delle istanze di container Amazon ECS Linux per ricevere avvisi sulle istanze Spot

Amazon EC2 termina, arresta 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 preavviso di due minuti di interruzione dell'istanza Spot per le operazioni di terminazione e interruzione. Non fornisce l'avviso di due minuti per l'operazione di ibernazione. Se il drenaggio delle istanze Spot di Amazon ECS è attivato sull'istanza, Amazon ECS riceve l'avviso di interruzione dell'istanza Spot e imposta lo stato dell'istanza. `DRAINING`

> **Important**
>
> Amazon ECS non riceve alcun avviso da Amazon EC2 quando le istanze vengono rimosse da Ribilanciamento della capacità di Auto Scaling. Per ulteriori informazioni, consulta Ribilanciamento della capacità di Amazon EC2 Auto Scaling.

Quando un'istanza di container è impostata su `DRAINING`, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container. Le attività di servizio nell'istanza di container di esaurimento che sono in stato `PENDING` vengono interrotte immediatamente. Se nel cluster sono disponibili istanze di container, le attività del servizio di sostituzione vengono avviate su di esse.

Il drenaggio delle istanze Spot è disattivato per impostazione predefinita.

Puoi attivare il drenaggio delle istanze Spot all'avvio di un'istanza. Aggiungi il seguente script nel campo Dati utente. Sostituisci *`MyCluster`*con il nome del cluster in cui registrare l'istanza del contenitore.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_ENABLE_SPOT_INSTANCE_DRAINING=true
EOF
```

Per ulteriori informazioni, consulta Avvio di un'istanza di container Linux di Amazon ECS.

Per attivare lo svuotamento dell'istanza Spot per un'istanza di container esistente

1. Connettiti all'istanza Spot su SSH.

2. Modifica il file `/etc/ecs/ecs.config` e aggiungi quanto segue:

```
ECS_ENABLE_SPOT_INSTANCE_DRAINING=true
```

3. Riavvia il servizio `ecs`.

   - Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart ecs
```

4. (Facoltativo) Puoi verificare se l'agente è in esecuzione e visualizzare alcune informazioni sulla nuova istanza di container interrogando l'operazione API di introspezione dell'agente. Per ulteriori informazioni, consulta the section called “Introspezione dei contenitori”.

```
curl http://localhost:51678/v1/metadata
```

Esecuzione di uno script all'avvio di un'istanza di container Amazon ECS Linux

Potrebbe essere necessario eseguire un contenitore specifico su ogni istanza del contenitore per gestire problemi operativi o di sicurezza come il monitoraggio, la sicurezza, le metriche, l'individuazione dei servizi o la registrazione.

Per eseguire questa operazione, puoi configurare le istanze di container per chiamare il comando docker run con lo script di dati utente all'avvio o in un sistema di inizializzazione come Upstart o systemd. Anche se questo metodo funziona, presenta di alcuni svantaggi perché Amazon ECS non conosce i container e non può monitorare CPU, memoria, porte o le altre risorse utilizzate. Per garantire che Amazon ECS tenga correttamente conto di tutte le risorse delle attività, crea una definizione di attività per il container per l'esecuzione nelle istanze di container. Quindi, utilizza Amazon ECS per posizionare il processo al momento dell'avvio con i dati utente di Amazon EC2.

Lo script di dati utente di Amazon EC2 nella procedura seguente utilizza l'API di introspezione di Amazon ECS per identificare l'istanza di container. Quindi, utilizza il AWS CLI start-task comando and per eseguire un'attività specificata 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 Modifica di un ruolo nella AWS Identity and Access Management Guida per l'utente di IAM.

2. Avvia una o più istanze di container utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Avvia nuove istanze di container e usa il seguente script di esempio nei dati utente EC2. Sostituisci *your_cluster_name con il* cluster in cui l'istanza del contenitore deve registrarsi e *my_task_def* con la definizione dell'attività da eseguire sull'istanza al momento del lancio.

   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 all'indirizzo https://console.aws.amazon.com/ecs/v2.

b. Dalla barra di navigazione, scegli la regione in cui si trova il cluster.

c. Nel pannello di navigazione, scegli Clusters (Cluster) e seleziona il cluster che ospita le istanze di container.

d. Nella pagina Cluster, seleziona Attività e quindi le tue attività.

   Su ogni istanza di container che hai avviato dovrebbe essere in esecuzione la tua attività.

   Se non visualizzi le attività, puoi accedere alle istanze di container con SSH e controllare le informazioni di debug nel file `/var/log/ecs/ecs-start-task.log`.

Aumento delle interfacce di rete di istanze di container Amazon ECS Linux

> **Note**
>
> Questa funzione non è disponibile su Fargate.

Ogni attività Amazon ECS che utilizza la modalità di `awsvpc` rete riceve la propria elastic network interface (ENI), collegata all'istanza del contenitore che la ospita. Esiste un limite di default al numero di interfacce di rete che possono essere collegate a un'istanza di Amazon EC2 e l'interfaccia di rete primaria viene conteggiata come una di queste. Ad esempio, di default, un'istanza `c5.large` può avere fino a tre ENI collegate a essa. L'interfaccia di rete primaria per l'istanza viene conteggiata come una, pertanto puoi collegare altre due ENI all'istanza. Poiché ogni attività che utilizza la modalità `awsvpc` di rete richiede unaENI, in genere è possibile eseguire solo due di queste attività su questo tipo di istanza.

Amazon ECS supporta il lancio di istanze di container con maggiore ENI densità utilizzando tipi di istanze Amazon EC2 supportati. Quando usi questi tipi di istanze e attivi l'impostazione dell'`awsvpcTrunking`account, sono disponibili ENI aggiuntivi sulle istanze di container appena lanciate. Questa configurazione consente di posizionare più attività su ciascuna istanza di container. Per informazioni sull'impostazione dell'`awsvpcTrunking`account, consulta. Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account

Ad esempio, un'`c5.large`istanza con `awsvpcTrunking` ha un ENI limite 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.

L'interfaccia di rete trunk è completamente gestita da Amazon ECS e viene eliminata quando termini o annulli la registrazione dell'istanza di container dal cluster. Per ulteriori informazioni, consulta Opzioni di task networking di Amazon ECS per il tipo di lancio EC2.

Considerazioni

Considerate quanto segue quando utilizzate la funzione di ENI trunking.

- Solo le varianti Linux dell'AMI ottimizzata per Amazon ECS o altre varianti di Amazon Linux con versione `1.28.1` o successiva dell'agente contenitore e versione `1.28.1-2` o successiva del pacchetto ecs-init supportano i limiti aumentati. ENI 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 istanze Amazon EC2 lanciate dopo l'attivazione `awsvpcTrunking` ricevono ENI i limiti aumentati e l'interfaccia di rete trunk. In precedenza, le istanze avviate non ricevevano queste caratteristiche, a prescindere dalle operazioni eseguite.
- Le richieste DNS IPv4 basate su risorse delle istanze Amazon EC2 devono essere disattivate. Per disabilitare questa opzione, assicurati che l'opzione Enable resource-based IPV4 (A record) DNS requests (Abilita richieste DNS IPV4 (registro A) basate sulle risorse) 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
  ```

- Le istanze Amazon EC2 nelle sottoreti condivise non sono supportate. Se vengono utilizzate, non riescono a eseguire la registrazione a un cluster.
- I processi Amazon ECS devono utilizzare la modalità di rete `awsvpc` e il tipo di avvio EC2. Le attività che utilizzano il tipo di lancio Fargate hanno sempre ricevuto un messaggio dedicato ENI indipendentemente da quante ne sono state avviate, quindi questa funzionalità non è necessaria.
- I processi di Amazon ECS devono essere avviati 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 la chiusura dell'istanza del contenitore, Amazon ECS eliminerà il trunk. ENI

> **Note**
>
> Amazon ECS emette eventi di modifica dello stato dell'istanza di container che è possibile monitorare per le istanze che passano a uno stato `REGISTRATION_FAILED`. Per ulteriori informazioni, consulta Eventi di modifica dello stato dell'istanza del contenitore Amazon ECS.

- Al termine dell'istanza di container, l'istanza esegue la transizione a uno stato `DEREGISTERING` mentre viene annullato il provisioning dell'interfaccia di rete elastica trunk. L'istanza quindi esegue la transizione a uno stato `INACTIVE`.
- Se un'istanza di contenitore in una sottorete pubblica con ENI limiti elevati viene interrotta e quindi riavviata, l'istanza perde il suo indirizzo IP pubblico e l'agente del contenitore perde la connessione.
- Quando abilitia`wsvpcTrunking`, le istanze di container ne ricevono un'altra ENI che utilizza il gruppo di sicurezza predefinito del VPC ed è gestita da Amazon ECS.

Prerequisiti

Prima di avviare un'istanza di container con i ENI limiti aumentati, è necessario completare i seguenti prerequisiti.

- Il ruolo collegato ai servizi per Amazon ECS deve essere creato. Il ruolo collegato al servizio Amazon ECS fornisce ad Amazon ECS le autorizzazioni per effettuare chiamate ad altri AWS servizi per tuo conto. Questo ruolo viene creato automaticamente quando crei un cluster oppure quando crei o aggiorni un servizio nella AWS Management Console. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS. Puoi anche creare il ruolo collegato al servizio con il seguente comando. AWS CLI

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

- Il ruolo IAM dell'account o dell'istanza di container deve abilitare le impostazioni dell'account `awsvpcTrunking`. Ti consigliamo di creare 2 ruoli di istanza del contenitore ()`ecsInstanceRole`. È quindi possibile abilitare l'impostazione dell'`awsvpcTrunking`account per un ruolo e utilizzare quel ruolo per attività che richiedono il trunking ENI. Per informazioni sul ruolo dell'istanza del contenitore, vedere. Ruolo IAM delle istanze di container Amazon ECS

Una volta soddisfatti i prerequisiti, puoi avviare una nuova istanza di container utilizzando uno dei tipi di istanza Amazon EC2 supportati e l'istanza avrà i ENI limiti maggiori. Per una lista di tipi di istanze supportate, consulta Istanze supportate per interfacce di rete di container Amazon ECS potenziate. La versione dell'istanza di container per l'agente del container deve essere `1.28.1` o successiva e la versione del pacchetto ecs-init deve essere `1.28.1-2` o successiva. Se utilizzi la variante Linux più recente dell'AMI ottimizzata per Amazon ECS, questi requisiti saranno soddisfatti. Per ulteriori informazioni, consulta Avvio di un'istanza di container Linux di Amazon ECS.

> **Important**
>
> Le richieste DNS IPv4 basate su risorse delle istanze Amazon EC2 devono essere disattivate. Per disabilitare questa opzione, assicurati che l'opzione Enable resource-based IPV4 (A record) DNS requests (Abilita richieste DNS IPV4 (registro A) basate sulle risorse) 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. Usa il comando seguente per elencare le istanze del contenitore con ENI limiti maggiori eseguendo una query sull'`ecs.awsvpc-trunk-id`attributo, 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.awsvpc-trunk-id \
```

```
    --cluster cluster_name \
    --region us-east-1
```

- AttributeListget-ECS ()AWS Tools for Windows PowerShell

```
Get-ECSAttributeList -TargetType container-instance -AttributeName ecs.awsvpc-trunk-id -Region us-east-1
```

Istanze supportate per interfacce di rete di container Amazon ECS potenziate

Di seguito vengono visualizzati i tipi di istanze Amazon EC2 supportati e il numero di attività che usano la modalità di rete `awsvpc` che è possibile avviare su ogni tipo di istanza prima e dopo aver abilitato l'impostazione dell'account `awsvpcTrunking`. Per i limiti di elastic network interface (ENI) per ogni tipo di istanza, aggiungine uno al limite di attività corrente, poiché l'interfaccia di rete principale conta rispetto al limite, e aggiungine due al nuovo limite di attività, poiché sia l'interfaccia di rete principale che l'interfaccia di rete trunk contano nuovamente il limite.

> **Important**
>
> Sebbene altri tipi di istanza siano supportati nella stessa famiglia di istanze, i tipi di istanza `a1.metal`, `c5.metal`, `c5a.8xlarge`, `c5ad.8xlarge`, `c5d.metal`, `m5.metal`, `p3dn.24xlarge`, `r5.metal`, `r5.8xlarge` e `r5d.metal` non sono supportati.
> Le famiglie di istanze `c5n`, `d3`, `d3en`, `g3`, `g3s`, `g4dn`, `i3`, `i3en`, `inf1`, `m5dn`, `m5n`, `m5zn`, `mac1`, `r5b`, `r5n`, `r5dn`, `u-12tb1`, `u-6tb1`, `u-9tb1` e `z1d` non sono supportate.

Argomenti

- Uso generale
- Calcolo ottimizzato
- Memoria ottimizzata
- Archiviazione ottimizzata
- Elaborazione accelerata
- High Performance Computing

Uso generale

Tipo di istanza	Limite di attività senza ENI trunking	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 ENI trunking	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
m5n.large	2	10

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
m5n.xlarge	3	20
m5n.2xlarge	3	40
m5n.4xlarge	7	60
m5n.8xlarge	7	60
m5n.12xlarge	7	60
m5n.16xlarge	14	120
m5zn.large	2	14
m5zn.xlarge	3	31
m5zn.2xlarge	3	64
m5zn.3xlarge	7	98
m5zn.6xlarge	7	120
m6a.large	2	10
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

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
m6a.48xlarge	14	120
m6a.metal	14	120
m6g.medium	1	4
m6g.large	2	10
m6g.xlarge	3	20
m6g.2xlarge	3	40
m6g.4xlarge	7	60
m6g.8xlarge	7	60
m6g.12xlarge	7	60
m6g.16xlarge	14	120
m6g.metal	14	120
m6gd.medium	1	4
m6gd.large	2	10
m6gd.xlarge	3	20
m6gd.2xlarge	3	40
m6gd.4xlarge	7	60
m6gd.8xlarge	7	60
m6gd.12xlarge	7	60
m6gd.16xlarge	14	120
m6gd.metal	14	120

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
m6i.large	2	10
m6i.xlarge	3	20
m6i.2xlarge	3	40
m6i.4xlarge	7	60
m6i.8xlarge	7	90
m6i.12xlarge	7	120
m6i.16xlarge	14	120
m6i.24xlarge	14	120
m6i.32xlarge	14	120
m6i.metal	14	120
m6id.large	2	10
m6id.xlarge	3	20
m6id.2xlarge	3	40
m6id.4xlarge	7	60
m6id.8xlarge	7	90
m6id.12xlarge	7	120
m6id.16xlarge	14	120
m6id.24xlarge	14	120
m6id.32xlarge	14	120
m6id.metal	14	120

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
m6idn.large	2	10
m6idn.xlarge	3	20
m6idn.2xlarge	3	40
m6idn.4xlarge	7	60
m6idn.8xlarge	7	90
m6idn.12xlarge	7	120
m6idn.16xlarge	14	120
m6idn.24xlarge	14	120
m6idn.32xlarge	15	120
m6idn.metal	15	120
m6in.large	2	10
m6in.xlarge	3	20
m6in.2xlarge	3	40
m6in.4xlarge	7	60
m6in.8xlarge	7	90
m6in.12xlarge	7	120
m6in.16xlarge	14	120
m6in.24xlarge	14	120
m6in.32xlarge	15	120
m6in.metal	15	120

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
m7a.medium	1	4
m7a.large	2	10
m7a.xlarge	3	20
m7a.2xlarge	3	40
m7a.4xlarge	7	60
m7a.8xlarge	7	90
m7a.12xlarge	7	120
m7a.16xlarge	14	120
m7a.24xlarge	14	120
m7a.32xlarge	14	120
m7a.48xlarge	14	120
m7a.metal-48xl	14	120
m7g.medium	1	4
m7g.large	2	10
m7g.xlarge	3	20
m7g.2xlarge	3	40
m7g.4xlarge	7	60
m7g.8xlarge	7	60
m7g.12xlarge	7	60
m7g.16xlarge	14	120

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
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
m7 g.d., metallo	14	120
m7i.large	2	10
m7i.xlarge	3	20
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

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
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
mac2.metal	7	12
mac2-m2.metal	7	12
mac2-m2pro.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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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.metallo	14	120
c7i.large	2	10
c7i.xlarge	3	20
c7i.2xlarge	3	40
c7i.4xlarge	7	60
c7i.8xlarge	7	90
c7i.12xlarge	7	120
c7i.16xlarge	14	120
c7i.24xlarge	14	120
c7i.48xlarge	14	120
c7i.metal-24xl	14	120
c7i.metal-48xl	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
c7i-flex.large	2	4
c7i-flex.xlarge	3	10
c7i-flex.2xgrande	3	20
c7i-flex.4xgrande	7	40
c7i-flex.8xgrande	7	60

Memoria ottimizzata

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r5.large	2	10
r5.xlarge	3	20
r5.2xlarge	3	40
r5.4xlarge	7	60
r5.12xlarge	7	60
r5.16xlarge	14	120
r5.24xlarge	14	120
r5a.large	2	10
r5a.xlarge	3	20
r5a.2xlarge	3	40
r5a.4xlarge	7	60

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r6i.metal	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
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
r6id. Metallo	14	120
r7a. medio	1	4
r7a.large	2	10
r7a.xlarge	3	20
r7a.2xlarge	3	40
r7a.4xlarge	7	60
r7a.8xlarge	7	90
r7a.12xlarge	7	120
r7a.16xlarge	14	120
r7a.24xlarge	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
r7iz.metal-16xl	14	120
r7iz.metal-32xl	14	120
u-3tb1.56xlarge	7	12
u-6tb1.56xlarge	14	12
u-18tb1.112xlarge	14	12
u-18tb1.metal	14	12
u-24tb1.112xlarge	14	12
u-24tb1.metal	14	12
u7i-12 tb.224 x grande	14	120
u7in-16 tb.224xlarge	15	120
u7in-24 tb.224xlarge	15	120
u7in-32 tb.224xlarge	15	120
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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
x2gd.metal	14	120
x2idn.16xlarge	14	120
x2idn.24xlarge	14	120
x2idn.32xlarge	14	120
x2idn.metal	14	120
x2iedn.xlarge	3	13
x2iedn.2xlarge	3	29
x2iedn.4xlarge	7	60
x2iedn.8xlarge	7	120
x2iedn.16xlarge	14	120
x2iedn.24xlarge	14	120
x2iedn.32xlarge	14	120
x2iedn.metal	14	120
x2iezn.2xlarge	3	64
x2iezn.4xlarge	7	120
x2iezn.6xlarge	7	120
x2iezn.8xlarge	7	120
x2iezn.12xlarge	14	120
x2iezn.metal	14	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
i4i.12xlarge	7	118
i4i.16xlarge	14	248
i4i.24xlarge	14	118
i4i.32xlarge	14	498
i4i.metal	14	498
im4gn.large	2	10
im4gn.xlarge	3	20
im4gn.2xlarge	3	40
im4gn.4xlarge	7	60

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
dl1.24xlarge	59	120
dl2q.24xlarge	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

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
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
g5g.2xlarge	3	40
g5g.4xlarge	7	60
g5g.8xlarge	7	60
g5g.16xlarge	14	120
g5g.metal	14	120
g 6, x, grande	3	20
g6,2 x grande	3	40
g 6,4 x grande	7	60
g 6,8 x grande	7	90
g 6,12 x grande	7	120
g6,16 x grande	14	120
g6,24 x grande	14	120

Tipo di istanza	Limite di attività senza trunking ENI	Limite di attività con trunking ENI
g6,48 x grande	14	120
gr 6,4 x grande	7	60
gr 6,8 x grande	7	90
inf2.xlarge	3	20
inf2.8xlarge	7	90
inf2.24xlarge	14	120
inf2.48xlarge	14	120
p4d.24xlarge	59	120
p4de.24xlarge	59	120
p5.48xlarge	63	242
trn1.2xlarge	3	19
trn1.32xlarge	39	120
trn1n.32xlarge	79	242
vt1.3xlarge	3	40
vt1.6xlarge	7	60
vt1.24xlarge	14	120

High Performance Computing

Tipo di istanza	Limite di attività senza ENI trunking	Limite di attività con trunking ENI
hpc6a.48xlarge	1	120
hpc6id.32xlarge	1	120
hpc7g.4xlarge	3	120
hpc7g.8xlarge	3	120
hpc7g.16xlarge	3	120

Riservare la memoria delle istanze del contenitore Amazon ECS Linux

Quando l'agente container Amazon ECS registra un'istanza di contenitore in un cluster, deve determinare la quantità di memoria disponibile nell'istanza del contenitore da riservare per le tue attività. A causa dei costi del sovraccarico di memoria della piattaforma e della memoria occupata dal kernel del sistema, questo numero è diverso rispetto alla quantità di memoria installata pubblicizzata per le istanze Amazon EC2. Ad esempio, un'istanza `m4.large` dispone di 8 GiB di memoria installata. Tuttavia, ciò non sempre si traduce esattamente in 8192 MiB di memoria disponibili per le attività quando l'istanza del contenitore si registra.

L'agente container Amazon ECS fornisce una variabile di configurazione chiamata`ECS_RESERVED_MEMORY`, che puoi usare per rimuovere un numero specifico di MiB di memoria dal pool assegnato alle tue attività. In questo modo si riserva la memoria per i processi di sistema critici.

Se occupi tutta la memoria di un'istanza del contenitore per le tue attività, è possibile che queste abbiano a che fare con processi di sistema critici per la memoria e che possano causare un errore di sistema.

Ad esempio, se si specifica `ECS_RESERVED_MEMORY=256` nel file di configurazione dell'agente del container, l'agente registra la memoria totale -256 MiB per quell'istanza e 256 MiB di memoria non possono essere allocati da task ECS. Per ulteriori informazioni sulle variabili dell'agente e su come impostarle, vedere Configurazione dell'agente del container Amazon ECS e Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati.

Se specifichi 8192 MiB per l'attività e nessuna delle istanze del contenitore dispone di 8192 MiB o più di memoria disponibile per soddisfare questo requisito, l'attività non può essere inserita nel cluster. Se si utilizza un ambiente di elaborazione gestito, è AWS Batch necessario avviare un tipo di istanza più grande per soddisfare la richiesta.

È inoltre necessario riservare parte della memoria per l'agente del container di Amazon ECS e per altri processi di sistema critici nelle istanze di container, in modo che i container dell'attività non si contendano la stessa memoria e finiscano con l'attivare un errore di sistema.

L'agente del container di Amazon ECS utilizza la funzione Docker `ReadMemInfo()` per eseguire una query sulla memoria totale disponibile per il sistema operativo. Sia Linux che Windows forniscono utilità da riga di comando per determinare la memoria totale.

Example - Determinare la memoria totale in Linux

Il comando free restituisce la memoria totale riconosciuta dal sistema operativo.

```
$ free -b
```

Esempio di output per un'istanza `m4.large` che esegue l'AMI Amazon Linux ottimizzata per Amazon ECS.

```
             total       used       free     shared    buffers     cached
Mem:    8373026816  348180480 8024846336      90112   25534464  205418496
-/+ buffers/cache:  117227520 8255799296
```

Questa istanza ha 8373026816 byte di memoria totale, che si traducono in 7985 MiB disponibili per le attività.

Example - Determinare la memoria totale in Windows

Il comando wmic restituisce la memoria totale riconosciuta dal sistema operativo.

```
C:\> wmic ComputerSystem get TotalPhysicalMemory
```

Esempio di output per un'`m4.large`istanza che esegue l'AMI Windows Server ottimizzata per Amazon ECS.

```
TotalPhysicalMemory
8589524992
```

Questa istanza ha 8589524992 byte di memoria totale, che si traducono in 8191 MiB disponibili per le attività.

Visualizzazione della memoria dell'istanza del contenitore

Puoi visualizzare la quantità di memoria con cui registra un'istanza di contenitore nella console Amazon ECS (o con l'operazione dell'API DescribeContainerInstances). Se stai cercando di massimizzare l'utilizzo delle risorse fornendo alle tue attività quanta più memoria possibile per un particolare tipo di istanza, puoi osservare la memoria disponibile per quell'istanza del contenitore e quindi assegnare alle tue attività quella quantità di memoria.

Per visualizzare la memoria dell'istanza del contenitore

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel riquadro di navigazione, scegli Cluster, quindi scegli il cluster che ospita l'istanza del contenitore.
3. Scegli Infrastruttura, quindi in Istanze di container, scegli un'istanza di contenitore.
4. La sezione Risorse mostra la memoria registrata e disponibile per l'istanza del contenitore.

   Il valore di memoria registrata è quello dell'istanza del contenitore, registrato con Amazon ECS al primo avvio, e il valore di memoria disponibile è quello che non è già stato assegnato alle attività.

Gestione remota delle istanze di container Amazon ECS tramite AWS Systems Manager

Puoi utilizzare la funzionalità Run Command in AWS Systems Manager (Systems Manager) per gestire in modo sicuro e remoto la configurazione delle tue istanze di container Amazon ECS. Run Command fornisce un modo semplice per eseguire le attività di amministrazione comuni senza accedere localmente all'istanza. Puoi gestire le modifiche della configurazione dei tuoi cluster tramite l'esecuzione simultanea di comandi su più istanze di container. Run Command tiene traccia dello stato e dei risultati di ciascun comando.

Di seguito sono elencati alcuni esempi dei tipi di attività che puoi eseguire mediante Run Command:

- Installare o disinstallare pacchetti.
- Eseguire aggiornamenti di sicurezza.
- Eliminare immagini Docker.
- Interrompere o avviare i servizi.

- Visualizzare le risorse di sistema.
- Visualizzare file di log.
- Eseguire operazioni sui file.

Per ulteriori informazioni su Run Command, consulta AWS Systems Manager Run Command nella Guida per l'utente di AWS Systems Manager .

Di seguito sono riportati i prerequisiti per l'utilizzo di Systems Manager con Amazon ECS.

1. È necessario concedere al ruolo dell'istanza del contenitore (ecs InstanceRole) le autorizzazioni per accedere alle API Systems Manager. Puoi farlo assegnando AmazonSSM Core al ruolo ManagedInstance. `ecsInstanceRole` Per informazioni su come allegare una politica a un ruolo, consulta Modifica della politica di autorizzazione di un ruolo (console) nella Guida per l'utente AWS Identity and Access Management
2. Verifica che SSM Agent sia installato nelle istanze di container. Per ulteriori informazioni, consulta Installazione manuale di SSM Agent su istanze EC2 per Linux.

Dopo aver collegato le policy gestite di Systems Manager alle istanze del contenitore `ecsInstanceRole` e aver verificato che AWS Systems Manager l'agente (agente SSM) sia installato sulle istanze del contenitore, è possibile iniziare a utilizzare Run Command per inviare comandi alle istanze del contenitore. Per ulteriori informazioni sull'esecuzione di comandi e script di shell sulle istanze e sulla visualizzazione dell'output risultante, consulta Esecuzione di comandi mediante Run Command di Systems Manager e Spiegazione passo per passo di Run Command nella Guida per l'utente di AWS Systems Manager .

Un caso d'uso comune consiste nell'aggiornare il software delle istanze del contenitore con Run Command. È possibile seguire le procedure riportate nella Guida per l' AWS Systems Manager utente con i seguenti parametri.

Parametro	Valore
Documento di comando	`AWS-RunShellScript`
Comando	`$ yum update -y`
Istanze target	Le tue istanze di container

Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux

Puoi configurare le tue istanze di container Amazon ECS per l'utilizzo di un proxy HTTP sia per l'agente del container Amazon ECS che per il daemon Docker. Ciò è utile se le tue istanze di container non hanno accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

Per configurare l'istanza di container Linux di Amazon ECS per l'utilizzo di un proxy HTTP, imposta le seguenti variabili nei file pertinenti al momento dell'avvio (con dati utente di Amazon EC2). È inoltre possibile modificare manualmente il file di configurazione e quindi riavviare l'agente.

`/etc/ecs/ecs.config`(Amazon Linux 2e AmazonLinux AMI)

`HTTP_PROXY=10.0.0.131:3128`

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per l'agente Amazon ECS per la connessione a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

`NO_PROXY=169.254.169.254,169.254.170.2,/var/run/docker.sock`

Imposta questo valore su `169.254.169.254,169.254.170.2,/var/run/docker.sock` per filtrare i metadati dell'istanza EC2, i ruoli IAM per i processi e il traffico del daemon Docker 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 `169.254.169.254,169.254.170.2,/var/run/docker.sock` per filtrare i metadati dell'istanza EC2, i ruoli IAM per i processi e il traffico del daemon Docker dal proxy.

`/etc/init/ecs.override` (solo AMI Amazon Linux)

`env HTTP_PROXY=10.0.0.131:3128`

Imposta questo valore sul nome host (o indirizzo IP) e sul numero di porta di un proxy HTTP da utilizzare per connettere `ecs-init` a Internet. Ad esempio, le tue istanze di container potrebbero non avere accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

`env NO_PROXY=169.254.169.254,169.254.170.2,/var/run/docker.sock`

Imposta questo valore su `169.254.169.254,169.254.170.2,/var/run/docker.sock` per filtrare i metadati dell'istanza EC2, i ruoli IAM per i processi e il traffico del daemon Docker 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"`

Imposta questo valore su `169.254.169.254` per filtrare i metadati dell'istanza EC2 dal proxy.

`/etc/sysconfig/docker` (solo AMI 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 `169.254.169.254` per filtrare i metadati dell'istanza EC2 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 AL2023.

Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling

Amazon ECS supporta i warm pool Amazon EC2 Auto Scaling. Un warm pool è un gruppo di istanze Amazon EC2 pre-inizializzate pronte per essere messe in servizio. Ogni volta che l'applicazione ha bisogno di aumentare orizzontalmente, Amazon EC2 Auto Scaling utilizza le istanze pre-inizializzate dal warm pool anziché avviare istanze cold, consente l'esecuzione di qualsiasi processo di inizializzazione finale e quindi mette in servizio l'istanza.

Per ulteriori informazioni sui warm pool e su come aggiungere un warm pool al gruppo con scalabilità automatica, consulta Warm pool per Dimensionamento automatico Amazon EC2 nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

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 in Riferimenti della AWS Command Line Interface .

Per utilizzare i warm pool con il cluster Amazon ECS, imposta la variabile di configurazione dell'agente `ECS_WARM_POOLS_CHECK` su `true` nel campo User data (Dati utente) del modello di avvio del gruppo Amazon EC2 Auto Scaling.

Di seguito è illustrato un esempio di come la variabile di configurazione dell'agente può essere specificata nel campo User data (Dati utente) di un modello di avvio Amazon EC2. Sostituisci *`MyCluster`*con il nome del tuo cluster.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=MyCluster
ECS_WARM_POOLS_CHECK=true
EOF
```

La variabile `ECS_WARM_POOLS_CHECK` è supportata sull'agente solo a partire dalla versione `1.59.0`. Per ulteriori informazioni sulle variabili, consulta Configurazione dell'agente del container Amazon ECS.

Aggiornamento dell'agente del container Amazon ECS

Occasionalmente, potrebbe essere necessario aggiornare l'agente container Amazon ECS per rilevare correzioni di bug e nuove funzionalità. L'aggiornamento dell'agente del container di Amazon ECS non interrompe i processi o i servizi in esecuzione nell'istanza di container. Il processo per l'aggiornamento dell'agente differisce a seconda se la tua istanza di container è stata avviata con l'AMI ottimizzata per Amazon ECS o con un altro sistema operativo.

> **Note**
>
> Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Verifica della versione dell'agente del container Amazon ECS

Puoi verificare la versione dell'agente del container che è in esecuzione nelle tue istanze di container per vedere se è necessario aggiornarla. La vista dell'istanza di container nella console di Amazon ECS fornisce la versione dell'agente. Utilizza la procedura seguente per verificare la versione del tuo agente.

Amazon ECS console

1. Apri la console all'indirizzo 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. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (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 del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS).

> **Important**
>
> Per aggiornare la versione dell'agente di Amazon ECS da versioni precedenti alla v1.0.0 sull'AMI ottimizzata per Amazon ECS, ti consigliamo di terminare la tua attuale istanza di container e avviare una nuova istanza con la versione dell'AMI più recente. Qualsiasi istanza di container che utilizza una versione di anteprima deve essere ritirata e sostituita con l'AMI più recente. Per ulteriori informazioni, consulta Avvio di un'istanza di container Linux di Amazon ECS.

Amazon ECS container agent introspection API

Puoi inoltre utilizzare l'API di introspezione dell'agente del container di Amazon ECS per verificare la versione dell'agente dall'istanza di container stessa. Per ulteriori informazioni, consulta Introspezione dei container Amazon ECS.

Come verificare se il tuo agente del container di Amazon ECS sta eseguendo l'ultima versione con l'API di introspezione

1. Accedi alla tua istanza di container con SSH.
2. Interroga l'API di introspezione.

```
[ec2-user ~]$ curl -s 127.0.0.1:51678/v1/metadata | python3 -mjson.tool
```

> **Note**
>
> L'API di introspezione ha aggiunto delle informazioni di `Version` nella versione v1.0.0 dell'agente del container di Amazon ECS. Se `Version` non è presente quando si interroga l'API di introspezione oppure se quest'ultima è completamente assente nel tuo agente, la versione in esecuzione è v0.0.3 o precedente. È necessario aggiornare la versione.

Come aggiornare l'agente del container Amazon ECS su un'AMI ottimizzata per Amazon ECS

Se utilizzi l'AMI ottimizzata per Amazon ECS, hai diverse opzioni per ottenere la versione più recente dell'agente del container di Amazon ECS (mostrato in ordine di raccomandazione):

- Termina l'istanza di container e avvia la versione più recente dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS (manualmente o tramite l'aggiornamento della configurazione di avvio di Auto Scaling con l'AMI più recente). 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 AMI Linux ottimizzate per Amazon ECS.
- 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 gli AWS SDK AWS CLI o. 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 del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS). 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 gli AWS CLI AWS SDK. Amazon ECS verifica la versione corrente dell'agente confrontandola con l'ultima versione disponibile dell'agente e verifica se è possibile effettuare un aggiornamento. Se non è disponibile alcun aggiornamento, ad esempio se l'agente sta già eseguendo la versione più recente, allora viene restituito `NoUpdateAvailableException`.

Le fasi del processo di aggiornamento riportate in precedenza sono le seguenti:

`PENDING`

È disponibile un aggiornamento dell'agente e il processo di aggiornamento è stato avviato.

`STAGING`

L'agente ha iniziato a scaricare il relativo aggiornamento. Se l'agente non è in grado di scaricare l'aggiornamento, oppure se i contenuti dell'aggiornamento non sono corretti o sono danneggiati, l'agente invia una notifica dell'errore e l'aggiornamento passa allo stato `FAILED`.

`STAGED`

Il download dell'agente è stato completato e i contenuti dell'agente sono stati verificati.

`UPDATING`

Il servizio `ecs-init` è stato riavviato e ottiene la nuova versione dell'agente. Se l'agente non è in grado di riavviarsi per qualsiasi motivo, l'aggiornamento passa allo stato `FAILED`. In caso contrario, l'agente segnala ad Amazon ECS che l'aggiornamento è completo.

> **Note**
>
> Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Come aggiornare l'agente del container Amazon ECS su un'AMI ottimizzata per Amazon ECS nella console

1. Apri la console all'indirizzo 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. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (Infrastruttura).
5. In Istanze di container, seleziona le istanze da aggiornare, quindi scegli Operazioni, Aggiorna agente.

Aggiornamento manuale dell'agente del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS)

Come aggiornare manualmente l'agente del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS)

> **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**
>
> Puoi facoltativamente archiviare le variabili di ambiente dell'agente in Amazon S3 (che possono essere scaricate nelle tue istanze di container all'avvio utilizzando i dati utente di Amazon EC2). Ciò è consigliato per le informazioni sensibili, come le credenziali di autenticazione per i repository privati. Per ulteriori informazioni, consulta Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3 e Utilizzo di immagini non AWS containerizzate in Amazon ECS.

7. Estrai l'ultima immagine dell'agente del container di Amazon ECS da Amazon Elastic Container Registry Public.

```
ubuntu:~$ docker pull public.ecr.aws/ecs/amazon-ecs-agent:latest
```

Output:

```
Pulling repository amazon/amazon-ecs-agent
a5a56a5e13dc: Download complete
511136ea3c5a: Download complete
```

```
9950b5d678a1: Download complete
c48ddcf21b63: Download complete
Status: Image is up to date for amazon/amazon-ecs-agent:latest
```

8. Esegui l'agente del container di Amazon ECS più recente sulla tua istanza di container.

> **Note**
>
> Utilizza le policy di riavvio di Docker o un gestore di processo (come upstart o systemd) per trattare l'agente del container come un servizio o un daemon e assicurarne il riavvio dopo l'uscita. Per ulteriori informazioni, consulta l'articolo sull'avvio automatico dei container e sulle policy di riavvio nella documentazione di Docker. 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 su cui è abilitato SELinux richiedono l'opzione `--privileged` nel tuo comando docker run. Inoltre, per le istanze di container abilitate per SELinux, consigliamo di aggiungere l'opzione `:Z` ai montaggi dei volumi `/log` e `/data`. 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`. Esegui l'operazione seguente se riscontri difficoltà nell'esecuzione dell'agente Amazon ECS in un'istanza di container abilitata per SELinux:
>
> - 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.

AMI Windows ottimizzate per Amazon ECS

Le AMI ottimizzate per Amazon ECS sono preconfigurate con i componenti necessari per eseguire i carichi di lavoro Amazon ECS. Sebbene sia possibile creare un'AMI personalizzata per istanze 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 preconfigurate e testate su Amazon ECS da ingegneri. AWS È il modo più semplice per iniziare ed eseguire i container su AWS rapidamente.

I metadati dell'AMI ottimizzata per Amazon ECS, incluso il nome dell'AMI, la versione dell'agente del container di Amazon ECS e la versione runtime di Amazon ECS, che include la versione Docker, possono essere recuperati a livello di codice per ciascuna variante. Per ulteriori informazioni, consulta the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”.

Puoi sottoscrivere gli argomenti Amazon SNS dell'AMI di Windows in modo da ricevere una notifica quando viene rilasciata una nuova AMI o una versione dell'AMI è contrassegnata come privata. Per ulteriori informazioni, consulta Iscrizione alle notifiche di aggiornamento dell'AMI Windows ottimizzate per Amazon ECS.

> **Important**
>
> Tutte le varianti AMI ottimizzate per ECS prodotte dopo agosto migreranno da Docker EE (Mirantis) a Docker CE (progetto Moby).
> Per garantire gli aggiornamenti di sicurezza più recenti ai clienti di default, Amazon ECS mantiene almeno le ultime tre AMI Windows ottimizzate per Amazon ECS. Dopo aver rilasciato le nuove AMI Windows ottimizzate per Amazon ECS, Amazon ECS rende private le AMI Windows ottimizzate per Amazon ECS più vecchie. Se devi accedere a un'AMI privata, comunicacelo compilando un ticket con Cloud Support.

Varianti AMI ottimizzate per Amazon ECS

Le seguenti varianti Windows Server dell'AMI ottimizzata per Amazon ECS sono disponibili per le istanze Amazon EC2.

> **Important**
>
> Tutte le varianti AMI ottimizzate per ECS prodotte dopo agosto migreranno da Docker EE (Mirantis) a Docker CE (progetto Moby).

- AMI Windows Server 2022 Full ottimizzata per Amazon ECS
- AMI Windows Server 2022 Core ottimizzata per Amazon ECS
- AMI Windows Server 2019 Full ottimizzata per Amazon ECS
- AMI Windows Server 2019 Core ottimizzata per Amazon ECS
- AMI Windows Server 2016 Full ottimizzata per Amazon ECS

> **Important**
>
> Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, le AMI complete di Windows Server 2016 non riceveranno patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:
>
> - Windows Server 2022 Full
> - Windows Server 2022 Core

- Windows Server 2019 Full
- Windows Server 2019 Core

Il 9 agosto 2022, l'AMI Windows Server 20H2 Core ottimizzata per Amazon ECS ha raggiunto la data di fine del supporto. Non verranno rilasciate nuove versioni di questa AMI. Per ulteriori informazioni, consulta Informazioni sulle versioni di Windows Server.

Windows Server 2022, Windows Server 2019 e Windows Server 2016 sono versioni LTSC (Long-Term Servicing Channel). Windows Server 20H2 è una versione SAC (Semi-Annual Channel). Per ulteriori informazioni, consulta Informazioni sulle versioni di Windows Server.

Considerazioni

Di seguito sono riportate alcune informazioni utili sui container Windows di Amazon EC2 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 container Windows sono supportati per i processi che utilizzano il tipo di avvio EC2 e Fargate.
- I container Windows e le istanze di container non supportano tutti i parametri di definizione di attività disponibili per le istanze di container e i container Linux. Alcuni parametri non sono supportati, mentre altri si comportano in modo diverso su Windows e su Linux. Per ulteriori informazioni, consulta Differenze nella definizione delle attività di Amazon ECS per le istanze EC2 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 Windows di Amazon EC2.
- La funzionalità dei ruoli IAM per i processi utilizza un proxy di credenziali per fornire le credenziali ai container. Questo proxy di credenziali occupa la porta 80 nell'istanza di container, perciò se utilizzi i ruoli IAM per i processi, la porta 80 non è disponibile per i processi. Per i container di servizi Web, puoi utilizzare un Application Load Balancer e la mappatura di porte dinamiche per

fornire connessioni di porta 80 HTTP standard ai container. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS.

- Le immagini Docker del server Windows sono di grandi dimensioni (9 GiB). Pertanto, le istanze di container di Windows richiedono più spazio di archiviazione rispetto alle istanze di container Linux.
- Per eseguire un container Windows su Windows Server, la versione del sistema operativo dell'immagine di base del container deve corrispondere a quella dell'host. Per ulteriori informazioni, consulta Compatibilità con la versione del container Windows sul sito Web della documentazione Microsoft. Se il cluster esegue più versioni di Windows, puoi assicurarti che un'attività venga posizionata su un'istanza EC2 in esecuzione sulla stessa versione utilizzando il vincolo di posizionamento: `memberOf(attribute:ecs.os-family == WINDOWS_SERVER_<OS_Release>_<FULL or CORE>)`. Per ulteriori informazioni, consulta the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”.

Recupero di metadati AMI Windows ottimizzati per Amazon ECS

L'ID dell'AMI, il nome dell'immagine, il sistema operativo, la versione dell'agente del container e la versione di runtime per ogni variante delle AMI ottimizzate per Amazon ECS possono essere recuperati a livello di programmazione eseguendo una query sull'API dell'archivio parametri di Systems Manager. Per ulteriori informazioni sull'API Systems Manager Parameter Store, vedere GetParameterse GetParametersByPath.

> **Note**
>
> Per recuperare i metadati dell'AMI ottimizzata per Amazon ECS, l'utente di amministrazione deve disporre delle seguenti autorizzazioni IAM. Queste autorizzazioni sono state aggiunte alla policy IAM `AmazonECS_FullAccess`.
>
> - ssm: GetParameters
> - ssm: GetParameter
> - ssm: GetParameters ByPath

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 AMI Windows più recenti:
>
> - `/aws/service/ecs/optimized-ami/windows_server/2016/english/full/recommended/image_id`
> - `/aws/service/ecs/optimized-ami/windows_server/2019/english/full/recommended/image_id`

Di seguito è riportato il formato del nome del parametro per ogni variante AMI ottimizzata per Amazon ECS.

- Metadati dell'AMI di Windows Server 2022 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2022 Core:

```
/aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2019 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2019 Core:

```
/aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized
```

- Metadati dell'AMI di Windows Server 2016 Full:

```
/aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized
```

Il formato dei nomi di parametro seguente recupera i metadati dell'AMI stabile Windows Server 2019 completa più recente.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-
English-Full-ECS_Optimized
```

Di seguito viene mostrato un esempio dell'oggetto JSON restituito per il valore del parametro.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ami-windows-latest/Windows_Server-2019-English-Full-
ECS_Optimized",
            "Type": "String",
            "Value": "{\"image_name\":\"Windows_Server-2019-English-Full-
ECS_Optimized-2023.06.13\",\"image_id\":\"ami-0debc1fb48e4aee16\",\"ecs_runtime_version
\":\"Docker (CE) version 20.10.21\",\"ecs_agent_version\":\"1.72.0\"}",
            "Version": 58,
            "LastModifiedDate": "2023-06-22T19:37:37.841000-04:00",
            "ARN": "arn:aws:ssm:us-east-1::parameter/aws/service/ami-windows-latest/
Windows_Server-2019-English-Full-ECS_Optimized",
            "DataType": "text"
        }
    ],
    "InvalidParameters": []
}
```

Ognuno dei campi nell'output riportato sopra sono disponibili per essere interrogati come parametri secondari. Costruisci il percorso di parametro per un parametro secondario aggiungendo il nome del parametro secondario al percorso per l'AMI selezionata. Sono disponibili i seguenti parametri secondari:

- `schema_version`
- `image_id`
- `image_name`
- `os`
- `ecs_agent_version`
- `ecs_runtime_version`

Esempi

Negli esempi seguenti vengono illustrati i modi in cui è possibile recuperare i metadati per ogni variante dell'AMI ottimizzata per Amazon ECS.

Recupero dei metadati dell'AMI ottimizzata per Amazon ECS stabile più recente

Puoi recuperare l'ultima AMI stabile ottimizzata per Amazon ECS utilizzando AWS CLI i AWS CLI seguenti comandi.

- Per l'AMI Windows Server 2022 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2022 Core ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2019 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2019 Core ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized --region us-east-1
```

- Per l'AMI Windows Server 2016 Full ottimizzata per Amazon ECS:

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized --region us-east-1
```

Utilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation

Puoi consultare l'AMI ottimizzata per Amazon ECS più recente in un modello AWS CloudFormation facendo riferimento al nome dell'archivio parametri di Systems Manager.

```
Parameters:
```

```
  LatestECSOptimizedAMI:
    Description: AMI ID
    Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
    Default: /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-
ECS_Optimized/image_id
```

Iscrizione alle notifiche di aggiornamento dell'AMI Windows ottimizzate per Amazon ECS

AWS fornisce due ARN tematici Amazon SNS per le notifiche relative alle AMI di Windows Server. Un argomento invia notifiche di aggiornamento quando vengono rilasciate nuove AMI di Windows Server. L'altro argomento invia notifiche quando le AMI di Windows Server rilasciate in precedenza vengono rese private. Sebbene questi argomenti non siano specifici delle AMI Windows ottimizzate per Amazon ECS, poiché le AMI Windows ottimizzate per Amazon ECS seguono lo stesso programma di rilascio, puoi utilizzare queste notifiche come indicazione di quando le nuove AMI Windows ottimizzate per Amazon ECS vengono aggiornate. Per ulteriori informazioni sulla sottoscrizione alle notifiche Windows AMI, consulta Sottoscrizione alle notifiche Windows AMI nella Amazon EC2 User Guide.

> **Note**
>
> Per sottoscrivere un argomento Amazon SNS, l'utente, o il ruolo ad esso collegato, deve disporre dell'autorizzazione IAM `sns::subscribe`.

Versioni AMI Windows ottimizzate per Amazon ECS

Visualizza le versioni attuali e precedenti delle AMI ottimizzate per Amazon ECS e le versioni corrispondenti dell'agente container Amazon ECS, Docker e del pacchetto. `ecs-init`

I metadati dell'AMI ottimizzata per Amazon ECS, incluso l'ID AMI per ogni variante, possono essere recuperati a livello di programmazione. Per ulteriori informazioni, consulta the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”.

Nelle seguenti schede è riportato un elenco delle versioni delle AMI Windows ottimizzate per Amazon ECS. Per informazioni dettagliate su come fare riferimento al parametro Systems Manager Parameter Store in un AWS CloudFormation modello, vedereUtilizzo dell'ultima AMI ottimizzata per Amazon ECS consigliata in un modello AWS CloudFormation.

Important

Per garantire gli aggiornamenti di sicurezza più recenti ai clienti di default, Amazon ECS mantiene almeno le ultime tre AMI Windows ottimizzate per Amazon ECS. Dopo aver rilasciato le nuove AMI Windows ottimizzate per Amazon ECS, Amazon ECS rende private le AMI Windows ottimizzate per Amazon ECS più vecchie. Se devi accedere a un'AMI privata, comunicacelo compilando un ticket con Cloud Support.
Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, le AMI complete di Windows Server 2016 non riceveranno patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:

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

Windows Server 2022 Full AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2022 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-Inglese-Full-ECS_Optimized-2024.05.14	`1.82.3`	`25.0.3 (Docker CE)`	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.04.09	`1.82.2`	`25.0.3 (Docker CE)`	Pubblica

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.03.12	1.82.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.02.13	1.81.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2024.01.09	1.79.2	20.10.23 (Docker CE)	Pubblica
Windows_Server-2022-inglese-Full-ECS_Optimized-2023.12.12	1.79.1	20.10.23 (Docker CE)	Privata
Windows_Server-2022-inglese-Full-ECS_Optimized-2023.11.14	1.79.0	20.10.23 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.10.11	1.77.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.09.15	1.75.3	20.10.21 (Docker CE)	Privata

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-English-Full-ECS_Optimized-2023.08.09	1.74.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.07.11	1.73.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.06.13	1.72.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.05.18	1.71.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.04.18	1.70.2	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.03.21	1.69.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2023.02.21	1.68.2	20.10.21 (Docker CE)	Privata

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-English-Full-ECS_Optimized-2023.01.11	1.68.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.12.14	1.67.2	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.11.09	1.65.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.10.12	1.64.0	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.09.22	1.63.1	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.09.14	1.62.2	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.08.15	1.62.1	20.10.9	Privata

AMI Windows Server 2022 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-English-Full-ECS_Optimized-2022.07.13	1.61.3	20.10.9	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.06.15	1.61.2	20.10.9	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2022.01.18	1.57.1	20.10.9	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2021.12.16	1.57.1	20.10.7	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2021.11.11	1.57.0	20.10.7	Privata
Windows_Server-2022-English-Full-ECS_Optimized-2021.009.23	1.55.3	20.10.7	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2022 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Full-ECS_Optimized
```

Windows Server 2022 Core AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2022 Core ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-Inglese-Core-ECS_Optimized-2024.05.14	1.82.3	25.0.3 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2024.04.09	1.82.2	25.0.3 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2024.03.12	1.82.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2024.02.13	1.81.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2022-inglese-core-ecs_optimized-2024.01.09	1.79.2	20.10.23 (Docker CE)	Pubblica

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-inglese-core-ecs_optimized-2023.12.12	1.79.1	20.10.23 (Docker CE)	Privata
Windows_Server-2022-inglese-core-ecs_optimized-2023.11.14	1.79.0	20.10.23 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.10.11	1.77.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.09.15	1.75.3	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.08.09	1.74.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.07.11	1.73.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.06.13	1.72.0	20.10.21 (Docker CE)	Privata

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-English-Core-ECS_Optimized-2023.05.18	1.71.1	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.04.18	1.70.2	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.03.21	1.69.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.02.21	1.68.2	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2023.01.11	1.68.0	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.12.14	1.67.2	20.10.21 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.11.09	1.65.1	20.10.21 (Docker CE)	Privata

AMI Windows Server 2022 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2022-English-Core-ECS_Optimized-2022.10.12	1.64.0	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.09.22	1.63.1	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.09.14	1.62.2	20.10.17 (Docker CE)	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.08.15	1.62.1	20.10.9	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.07.13	1.61.3	20.10.9	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2022.06.15	1.61.2	20.10.9	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2021.12.16	1.57.1	20.10.7	Privata

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-2021.11.11	1.57.0	20.10.7	Privata
Windows_Server-2022-English-Core-ECS_Optimized-2021.009.23	1.55.3	20.10.7	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2022 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2022-English-Core-ECS_Optimized
```

Windows Server 2019 Full AMI versions

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2019 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-Inglese-Full-ECS_Optimized-2024.05.14	1.82.3	25.0.3 (Docker CE)	Pubblica
Windows_Server-2019-Inglese-Full-ECS	1.82.2	25.0.3 (Docker CE)	Pubblica

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
_Optimized-2024.04.09			
Windows_Server-2019-Inglese-Full-ECS_Optimized-2024.03.12	1.82.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2019-Inglese-Full-ECS_Optimized-2024.02.13	1.81.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2019-Inglese-Full-ECS_Optimized-2024.01.09	1.79.2	20.10.23 (Docker CE)	Pubblica
Windows_Server-2019-inglese-Full-ECS_Optimized-2023.12.12	1.79.1	20.10.23 (Docker CE)	Privata
Windows_Server-2019-Inglese-Full-ECS_Optimized-2023.11.14	1.79.0	20.10.23 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.10.11	1.77.0	20.10.21 (Docker CE)	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2023.09.15	1.75.3	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.08.09	1.74.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.07.11	1.73.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.06.13	1.72.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.05.18	1.71.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.04.18	1.70.2	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.03.21	1.69.0	20.10.21 (Docker CE)	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2023.02.21	1.68.2	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2023.01.11	1.68.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.12.14	1.67.2	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.11.09	1.65.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.10.12	1.64.0	20.10.17 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.09.22	1.63.1	20.10.17 (Docker CE)	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.09.14	1.62.2	20.10.17 (Docker CE)	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2022.08.15	1.62.1	20.10.9	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.07.13	1.61.3	20.10.9	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.06.15	1.61.2	20.10.9	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2022.01.18	1.57.1	20.10.9	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.12.16	1.57.1	20.10.7	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.11.11	1.57.0	20.10.7	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.009.23	1.55.3	20.10.7	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2021.08.12	1.55.0	20.10.6	Pubblica
Windows_Server-2019-English-Full-ECS_Optimized-2021.07.13	1.54.02	20.10.6	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.07.08	1.54.0	20.10.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.06.11	1.53.0	20.10.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.05.21	1.52.2	20.10.4	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.04.14	1.51.0	20.10.0	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.03.11	1.50.2	19.03.14	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2021.02.10	1.50.0	19.03.14	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2021.01.13	1.49.0	19.03.14	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.11.18	1.48.0	19.03.13	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.11.06	1.47.0	19.03.11	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.10.14	1.45.0	19.03.11	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.08.12	1.43.0	19.03.11	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.07.15	1.41.1	19.03.5	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2020.06.11	1.40.0	19.03.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.05.14	1.39.0	19.03.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2020.01.15	1.35.0	19.03.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.12.16	1.34.0	19.03.5	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.11.25	1.34.0	19.03.4	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.11.13	1.32.1	19.03.4	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.10.09	1.32.0	19.03.2	Privata

AMI Windows Server 2019 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Full-ECS_Optimized-2019.09.11	1.30.0	19.03.1	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.08.16	1.29.1	19.03.1	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.07.19	1.29.0	18.09.8	Privata
Windows_Server-2019-English-Full-ECS_Optimized-2019.05.10	1.27.0	18.09.4	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2019 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Full-ECS_Optimized
```

Windows Server 2019 Core AMI versions

> **Important**
>
> 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-2024.05.14	`1.82.3`	`25.0.3 (Docker CE)`	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.04.09	`1.82.2`	`25.0.3 (Docker CE)`	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.03.12	`1.82.0`	`20.10.23 (Docker CE)`	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.02.13	`1.81.0`	`20.10.23 (Docker CE)`	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2024.01.09	`1.79.2`	`20.10.23 (Docker CE)`	Pubblica
Windows_Server-2019-inglese-core-ecs_optimized-2023.12.12	`1.79.1`	`20.10.23 (Docker CE)`	Privata

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-inglese-core-ecs_optimized-2023.11.14	1.79.0	20.10.23 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.10.11	1.77.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.09.15	1.75.3	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.08.09	1.74.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.07.11	1.73.1	20.10.21 (Docker CE)	Privata

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Core-ECS_Optimized-2023.06.13	1.72.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.05.18	1.71.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.04.18	1.70.2	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.03.21	1.69.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2023.02.21	1.68.2	20.10.21 (Docker CE)	Privata

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Core-ECS_Optimized-2023.01.11	1.68.0	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.12.14	1.67.2	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.11.09	1.65.1	20.10.21 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.10.12	1.64.0	20.10.17 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.09.22	1.63.1	20.10.17 (Docker CE)	Privata

AMI Windows Server 2019 Core ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2019-English-Core-ECS_Optimized-2022.09.14	1.62.2	20.10.17 (Docker CE)	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.08.15	1.62.1	20.10.9	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.07.13	1.61.3	20.10.9	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.06.15	1.61.2	20.10.9	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2022.01.18	1.57.1	20.10.9	Privata

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-2021.12.16	1.57.1	20.10.7	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.11.11	1.57.0	20.10.7	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.09.23	1.55.3	20.10.7	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.08.12	1.55.0	20.10.6	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.07.13	1.54.02	20.10.6	Privata

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-2021.07.08	1.54.0	20.10.6	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.06.11	1.53.0	20.10.5	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.05.21	1.52.2	20.10.4	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.04.14	1.51.0	20.10.0	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.03.11	1.50.2	19.03.14	Privata

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-2021.02.10	1.50.0	19.03.14	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2021.01.13	1.49.0	19.03.14	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.11.18	1.48.0	19.03.13	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.11.06	1.47.0	19.03.11	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.10.14	1.45.0	19.03.11	Privata

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-2020.09.09	1.44.3	19.03.11	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.08.12	1.43.0	19.03.11	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.07.15	1.41.1	19.03.5	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.06.11	1.40.0	19.03.5	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2020.05.14	1.39.0	19.03.5	Privata

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-2020.01.15	1.35.0	19.03.5	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2019.12.16	1.34.0	19.03.5	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2019.11.25	1.34.0	19.03.4	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2019.11.13	1.32.1	19.03.4	Privata
Windows_Server-2019-English-Core-ECS_Optimized-2019.10.09	1.32.0	19.03.2	Privata

Usa il seguente AWS CLI comando per recuperare l'attuale AMI completa di Windows Server 2019 ottimizzata per Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2019-English-Core-ECS_Optimized
```

Windows Server 2016 Full AMI versions

Important

Windows Server 2016 non supporta la versione Docker più recente, ad esempio 25.x.x. Pertanto, le AMI complete di Windows Server 2016 non riceveranno patch di sicurezza o di bug nel runtime Docker. Ti consigliamo di passare a una delle seguenti piattaforme Windows:

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

La tabella seguente riporta le versioni correnti e precedenti dell'AMI Windows Server 2016 Full ottimizzata per Amazon ECS e le corrispondenti versioni di Docker e dell'agente del container di Amazon ECS.

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-Inglese-Full-ECS_Optimized-2024.03.12	`1.82.0`	`20.10.23 (Docker CE)`	Pubblica

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-Inglese-Full-ECS_Optimized-2024.02.13	1.81.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-Inglese-Full-ECS_Optimized-2024.01.09	1.79.2	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2023.12.12	1.79.1	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-inglese-Full-ECS_Optimized-2023.11.14	1.79.0	20.10.23 (Docker CE)	Pubblica
Windows_Server-2016-English-Full-ECS_Optimized-2023.10.11	1.77.0	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.09.15	1.75.3	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.08.09	1.74.1	20.10.21 (Docker CE)	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2023.07.11	1.73.1	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.06.13	1.72.0	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.05.18	1.71.1	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.04.18	1.70.2	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.03.21	1.69.0	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.02.21	1.68.2	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2023.01.11	1.68.0	20.10.21 (Docker CE)	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2022.12.14	1.67.2	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.11.09	1.65.1	20.10.21 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.10.12	1.64.0	20.10.17 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.09.22	1.63.1	20.10.17 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.09.14	1.62.2	20.10.17 (Docker CE)	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.08.15	1.62.1	20.10.9	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.07.13	1.61.3	20.10.9	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2022.06.15	1.61.2	20.10.9	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2022.01.18	1.57.1	20.10.9	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.12.16	1.57.1	20.10.7	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.11.11	1.57.0	20.10.7	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.09.23	1.55.3	20.10.7	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.08.12	1.55.0	20.10.6	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.07.13	1.54.02	20.10.6	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2021.07.08	1.54.0	20.10.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.06.11	1.53.0	20.10.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.05.21	1.52.2	20.10.4	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.04.14	1.51.0	20.10.0	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.03.11	1.50.2	19.03.14	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.02.10	1.50.0	19.03.14	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2021.01.13	1.49.0	19.03.14	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2020.11.18	1.48.0	19.03.13	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.11.06	1.47.0	19.03.11	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.10.14	1.45.0	19.03.12	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.09.09	1.44.3	19.03.11	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.08.12	1.43.0	19.03.11	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.07.15	1.41.1	19.03.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.06.11	1.40.0	19.03.5	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2020.05.14	1.39.0	19.03.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2020.01.15	1.35.0	19.03.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.12.16	1.34.0	19.03.5	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.11.25	1.34.0	19.03.4	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.11.13	1.32.1	19.03.4	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.10.09	1.32.0	19.03.2	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.09.11	1.30.0	19.03.1	Privata

AMI Windows Server 2016 Full ottimizzata per Amazon ECS	Versione dell'agente del container Amazon ECS	Versione Docker	Visibilità
Windows_Server-2016-English-Full-ECS_Optimized-2019.08.16	1.29.1	19.03.1	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.07.19	1.29.0	18.09.8	Privata
Windows_Server-2016-English-Full-ECS_Optimized-2019.03.07	1.26.0	18.03.1	Privata

Utilizza la seguente AMI completa per Windows Server 2016 ottimizzata per AWS CLI Amazon ECS.

```
aws ssm get-parameters --names /aws/service/ami-windows-latest/Windows_Server-2016-English-Full-ECS_Optimized
```

Creazione dell'AMI Windows ottimizzata per Amazon ECS

Usa EC2 Image Builder per creare la tua AMI Windows personalizzata ottimizzata per Amazon ECS. In questo modo è facile utilizzare un'AMI di Windows con la propria licenza su Amazon ECS. Amazon ECS fornisce un componente di Image Builder gestito che fornisce la configurazione del sistema necessaria per eseguire istanze di Windows per ospitare i container. Ogni componente gestito da Amazon ECS include un agente del container specifico e una versione Docker. Puoi personalizzare l'immagine in modo da utilizzare l'ultimo componente gestito da Amazon ECS oppure, se è necessario un agente del container o una versione Docker precedente, puoi specificare un componente diverso.

Per una spiegazione passo per passo dell'utilizzo di EC2 Image Builder, consulta Nozioni di base su EC2 Image Builder nella Guida per l'utente di EC2 Image Builder.

Quando crei la tua AMI Windows ottimizzata per Amazon ECS tramtie EC2 Image Builder, crei una ricetta per le immagini. La tua ricetta per le immagini deve soddisfare i seguenti requisiti:

- L'immagine sorgente deve essere basata su Windows Server 2019 Core, Windows Server 2019 Full, Windows Server 2022 Core o Windows Server 2022 Full. Altre eventuali versioni di Windows non sono supportate e potrebbero non essere compatibili con il componente.
- Quando si specifica l'opzione Crea componenti, il componente `ecs-optimized-ami-windows` è obbligatorio. Il componente `update-windows`, che garantisce che l'immagine contenga gli aggiornamenti della sicurezza più recenti, è consigliato.

  Per specificare una versione diversa del componente, espandi il menu Opzioni di controllo delle versioni e specifica la versione del componente da utilizzare. Per ulteriori informazioni, consulta Elenco delle versioni del componente `ecs-optimized-ami-windows`.

Elenco delle versioni del componente **`ecs-optimized-ami-windows`**

Quando crei una ricetta di EC2 Image Builder e specifichi il componente `ecs-optimized-ami-windows`, puoi utilizzare l'opzione di default oppure specificare una versione specifica del componente. Per determinare le versioni dei componenti che sono disponibili, insieme alle versioni dell'agente del container di Amazon ECS e del Docker contenute nel componente, puoi utilizzare la AWS Management Console.

Come elencare le versioni del componente **`ecs-optimized-ami-windows`** disponibili

1. Apri la console EC2 Image Builder all'indirizzo https://console.aws.amazon.com/imagebuilder/.
2. Nella barra di navigazione seleziona la regione in cui viene creata l'immagine.
3. Nel pannello di navigazione, sotto il menu Configurazioni salvate, scegli Componenti.
4. Sulla pagina Componenti, nella barra di ricerca digita `ecs-optimized-ami-windows` ed espandi verso il basso il menu di qualificazione, quindi seleziona Avvio rapido (gestito da Amazon).
5. Utilizza la colonna Descrizione per determinare la versione del componente con l'agente del container Amazon ECS e la versione Docker richiesta dall'immagine.

Gestione delle istanze di container Amazon ECS Windows

Quando utilizzi istanze EC2 per i tuoi carichi di lavoro Amazon ECS, sei responsabile della manutenzione delle istanze.

Gli aggiornamenti dell'agente non si applicano alle istanze di container di Windows. Consigliamo di avviare nuove istanze di container per aggiornare la versione dell'agente nei cluster Windows.

Procedure di gestione

- Avvio di un'istanza di container Windows di Amazon ECS
- Avvio delle istanze di container Amazon ECS Windows per il trasferimento di dati
- Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Windows
- Configurazione delle istanze di container Amazon ECS Windows per ricevere avvisi sulle istanze Spot

Avvio di un'istanza di container Windows di Amazon ECS

Le tue istanze di container di Amazon ECS sono 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.

È possibile utilizzare la nuova procedura guidata Amazon EC2 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 console Amazon EC2 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, a differenza di altre, alcune risorse Amazon EC2 possono essere condivise tra più regioni.
3. Dal pannello di controllo della console Amazon EC2, scegli Launch Instance (Avvia istanza).

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 AMI ottimizzate per Amazon ECS più recenti e i relativi valori, consulta AMI Amazon Windows 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 AMI seguenti nella barra Search (Cerca) 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 Scegli un'Amazon Machine Image (AMI), seleziona la categoria AMI della 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 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

Configurare le impostazioni di rete, se necessario.

- Networking platform Piattaforma di rete): scegli Virtual Private Cloud (VPC), quindi specifica la sottorete nella sezione Network interfaces (Interfacce di rete).
- VPC: selezionare un VPC esistente in cui creare il gruppo di sicurezza.
- Sottorete: è possibile avviare un'istanza in una sottorete associata a una zona di disponibilità, una Local Zone, una zona Wavelength o un Outpost.

  Per avviare l'istanza in una zona di disponibilità, selezionare la sottorete in cui avviare l'istanza. Per creare una nuova sottorete, scegliere Create new subnet (Crea nuova sottorete) per passare alla console Amazon VPC. Al termine, tornare alla procedura guidata di avvio istanza e scegliere Refresh (Aggiorna) per caricare la sottorete nell'elenco.

  Per avviare l'istanza in una Local Zone, selezionare una sottorete creata nella Local Zone.

  Per avviare un'istanza in un Outpost, selezionare una sottorete in un VPC associato a un Outpost.

- Auto-assign Public IP (IP pubblico di assegnazione automatica): se l'istanza deve essere accessibile da Internet, verifica che il campo Auto-assign Public IP (Assegna automaticamente IP pubblico) sia impostato su Enable (Abilita). In caso contrario, imposta il campo su Disable (Disabilita).

  > **Note**
  >
  > Le istanze di container richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite istanze di container con indirizzi IP pubblici.
  > Per ulteriori informazioni sugli endpoint VPC di interfaccia, vedi Endpoint VPC dell'interfaccia di Amazon ECS (AWS PrivateLink)
  > Se non disponi di un endpoint VPC di interfaccia configurato e le istanze di container non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta NAT gateways (Gateway NAT) nella Guida per l'utente di Amazon VPC e Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Linux in questa guida.

- Firewall (security groups) (Firewall [gruppi di sicurezza]): utilizza un gruppo di sicurezza per definire le regole del firewall per l'istanza di container. Tali regole specificano quale traffico di rete in entrata viene distribuito sull'istanza di container. Tutto il traffico rimanente verrà ignorato.
  - Per selezionare un gruppo di sicurezza esistente, scegli Select an existing security group (Seleziona un gruppo di sicurezza esistente), quindi seleziona il gruppo di sicurezza creato in Configurazione per l'uso di Amazon ECS

Per configurare l'archiviazione

L'AMI selezionata include uno o più volumi di archiviazione, compreso il volume dispositivo root. È possibile specificare altri volumi da collegare all'istanza.

Puoi utilizzare la vista Semplice.

- Storage Type (Tipo di storage): configura lo storage per l'istanza di container.

  Se stai utilizzando l'AMI Amazon Linux 2 ottimizzata per Amazon ECS, l'istanza dispone di un singolo volume di 30 GiB configurato che è condiviso tra il sistema operativo e Docker.

Se stai utilizzando l'AMI ottimizzata per Amazon ECS, l'istanza dispone di due volumi configurati. Il volume Root è per il sistema operativo, mentre il secondo volume di Amazon EBS (collegato a `/dev/xvdcz`) è per l'utilizzo di Docker.

Puoi aumentare o diminuire le dimensioni del volume per l'istanza in modo che soddisfi le esigenze applicative.

Dettagli avanzati

Per Advanced Details (Dettagli avanzati), espandi la sezione per visualizzare i campi e specifica eventuali parametri aggiuntivi per l'istanza.

- Purchasing option (Opzioni di acquisto): seleziona Request Spot Instances (Richiedi istanze Spot) per avviare un'istanza Spot. Dovrai anche impostare altri campi correlati alle istanze Spot. Per ulteriori informazioni, consulta Richiesta Istanza Spot.

  > **Note**
  >
  > Se utilizzi istanze Spot e visualizzi un messaggio `Not available`, potresti dover scegliere un altro tipo di istanza.

  .

- Profilo istanza IAM: seleziona il ruolo IAM dell'istanza di container. Questo di solito è chiamato `ecsInstanceRole`.

  > **Important**
  >
  > Se non avvii l'istanza di container con le autorizzazioni IAM corrette, l'agente Amazon ECS non potrà connettersi al cluster. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

- (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 di dati utente di Amazon EC2 vengono eseguiti solo una volta, al primo avvio dell'istanza. Di seguito sono elencati esempi comuni dei dati utente utilizzati per:

- Di default, l'istanza di container si avvia nel cluster predefinito. Per avviarla in un cluster non predefinito, scegli l'elenco Advanced Details (Dettagli avanzati). Quindi, incolla lo script seguente nel campo User data (Dati utente), sostituendo *your_cluster_name* con il nome del cluster.

  La `EnableTaskIAMRole` attiva la funzionalità ruoli IAM dei processi per i processi.

  Inoltre, le seguenti opzioni sono disponibili quando si utilizza la modalità di rete `awsvpc`.

  - `EnableTaskENI`: questo flag attiva la rete di processi ed è necessario quando si utilizza la modalità di rete `awsvpc`.
  - `AwsvpcBlockIMDS`: questo flag facoltativo blocca l'accesso IMDS per i contenitori di processi in esecuzione con la modalità di rete `awsvpc`.
  - `AwsvpcAdditionalLocalRoutes`: questo flag facoltativo consente di avere percorsi aggiuntivi nello spazio dei nomi dei processi.

    Sostituisci `ip-address` con l'indirizzo IP per le route aggiuntive, ad esempio 172.31.42.23/32.

  ```
  <powershell>
  Import-Module ECSTools
  Initialize-ECSAgent -Cluster your_cluster_name -EnableTaskIAMRole -EnableTaskENI -
  AwsvpcBlockIMDS -AwsvpcAdditionalLocalRoutes
  '["ip-address"]'
  </powershell>
  ```

Avvio delle istanze di container Amazon ECS Windows per il trasferimento di dati

Quando avvii un'istanza Amazon EC2, puoi passare i dati utente all'istanza EC2. 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, ad esempio hook di avvio del cloud, script di shell e direttive `cloud-init`. Per ulteriori informazioni su questi e altri tipi di formato, consulta la documentazione su cloud-init.

È possibile passare questi dati utente quando si utilizza la procedura guidata di avvio di Amazon EC2. 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 GitHubrepository Amazon ECS Container Agent.

> **Note**
>
> Questo script è condiviso a scopo esemplificativo. È molto più facile iniziare a utilizzare i container di Windows con l'AMI Windows Server ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta Creazione di un cluster Amazon ECS per il tipo di lancio Fargate.

Puoi usare questo script per le tue istanze di container (purché siano avviate con una versione dell'AMI Windows_Server-2016-English-Full-Containers). Assicurati di sostituire la riga *windows* per specificare il nome del tuo cluster (se non utilizzi un cluster denominato `windows`).

```
<powershell>
# Set up directories the agent uses
New-Item -Type directory -Path ${env:ProgramFiles}\Amazon\ECS -Force
New-Item -Type directory -Path ${env:ProgramData}\Amazon\ECS -Force
New-Item -Type directory -Path ${env:ProgramData}\Amazon\ECS\data -Force
# Set up configuration
$ecsExeDir = "${env:ProgramFiles}\Amazon\ECS"
[Environment]::SetEnvironmentVariable("ECS_CLUSTER", "windows", "Machine")
[Environment]::SetEnvironmentVariable("ECS_LOGFILE", "${env:ProgramData}\Amazon\ECS\log
\ecs-agent.log", "Machine")
[Environment]::SetEnvironmentVariable("ECS_DATADIR", "${env:ProgramData}\Amazon\ECS
\data", "Machine")
# Download the agent
$agentVersion = "latest"
$agentZipUri = "https://s3.amazonaws.com/amazon-ecs-agent/ecs-agent-windows-
$agentVersion.zip"
$zipFile = "${env:TEMP}\ecs-agent.zip"
Invoke-RestMethod -OutFile $zipFile -Uri $agentZipUri
```

```
# Put the executables in the executable directory.
Expand-Archive -Path $zipFile -DestinationPath $ecsExeDir -Force
Set-Location ${ecsExeDir}
# Set $EnableTaskIAMRoles to $true to enable task IAM roles
# Note that enabling IAM roles will make port 80 unavailable for tasks.
[bool]$EnableTaskIAMRoles = $false
if (${EnableTaskIAMRoles}) {
  $HostSetupScript = Invoke-WebRequest https://raw.githubusercontent.com/aws/amazon-
ecs-agent/master/misc/windows-deploy/hostsetup.ps1
  Invoke-Expression $($HostSetupScript.Content)
}
# Install the agent service
New-Service -Name "AmazonECS" `
        -BinaryPathName "$ecsExeDir\amazon-ecs-agent.exe -windows-service" `
        -DisplayName "Amazon ECS" `
        -Description "Amazon ECS service runs the Amazon ECS agent" `
        -DependsOn Docker `
        -StartupType Manual
sc.exe failure AmazonECS reset=300 actions=restart/5000/restart/30000/restart/60000
sc.exe failureflag AmazonECS 1
Start-Service AmazonECS
</powershell>
```

Utilizzo di un proxy HTTP per le istanze di container Amazon ECS Windows

Puoi configurare le tue istanze di container Amazon ECS per l'utilizzo di un proxy HTTP sia per l'agente del container Amazon ECS che per il daemon Docker. Ciò è utile se le tue istanze di container non hanno accesso alla rete esterna tramite un gateway Internet Amazon VPC, un gateway NAT o un'istanza.

Per configurare la tua istanza di container Windows di Amazon ECS per l'utilizzo di un proxy HTTP, imposta le seguenti variabili al momento dell'avvio (con i dati utente di Amazon EC2).

```
[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 `NO_PROXY` su `169.254.169.254,169.254.170.2,\\.\pipe\docker_engine` per filtrare i metadati dell'istanza EC2, i ruoli IAM per i processi e il traffico del daemon 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 riportato di seguito nel campo Dati utente (assicurati di sostituire i valori di esempio in rosso 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 Windows di Amazon EC2.

```
<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 termina, arresta 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 una notifica di interruzione dell'istanza spot, che dà all'istanza un preavviso di due minuti prima che venga interrotta. Se la funzione di svuotamento dell'istanza Spot di Amazon ECS è abilitata sull'istanza, ECS riceve l'avviso di interruzione dell'istanza Spot e posiziona l'istanza nello stato `DRAINING`.

> **Important**
>
> Amazon ECS monitora gli avvisi di interruzione dell'istanza Spot con le operazioni di istanza `terminate` e `stop`. Se hai specificato il comportamento di interruzione dell'istanza `hibernate` durante la richiesta di istanze Spot o del parco istanze Spot, la funzione di svuotamento delle istanze Spot di Amazon ECS non è supportata per tali istanze.

Quando un'istanza di container è impostata su `DRAINING`, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container. Le attività di servizio nell'istanza di container di esaurimento che sono in stato `PENDING` vengono interrotte immediatamente. Se nel cluster sono disponibili istanze di container, le attività del servizio di sostituzione vengono avviate su di esse.

Puoi attivare il drenaggio delle istanze Spot all'avvio di un'istanza. È necessario impostare il parametro `ECS_ENABLE_SPOT_INSTANCE_DRAINING` prima di avviare l'agente container. Sostituisci *`my-cluster`* con il nome del tuo cluster.

```
[Environment]::SetEnvironmentVariable("ECS_ENABLE_SPOT_INSTANCE_DRAINING", "true",
 "Machine")

# Initialize the agent
Initialize-ECSAgent -Cluster my-cluster
```

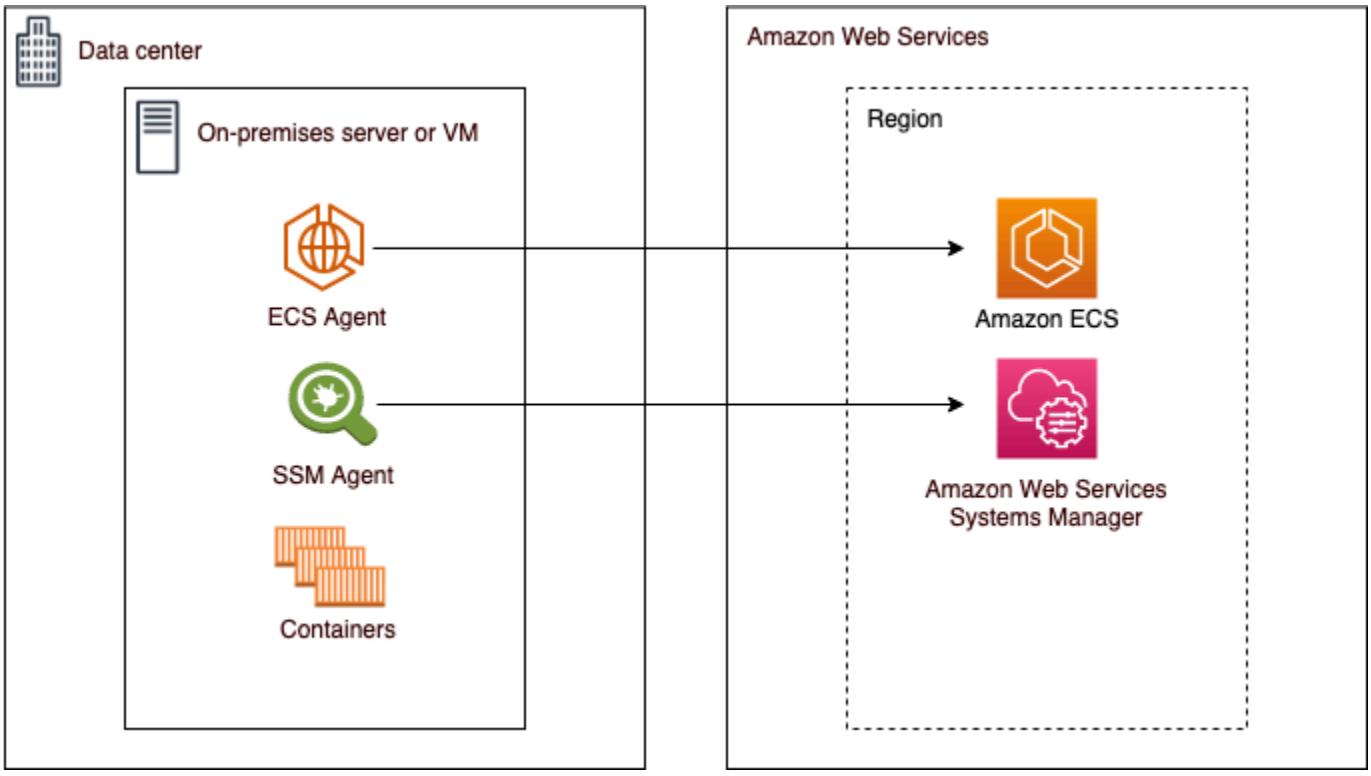
Per ulteriori informazioni, consulta the section called “Avvio di un'istanza di container”.

Cluster Amazon ECS per il tipo di lancio esterno

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

sono ottimizzate per l'esecuzione di applicazioni che generano traffico in uscita o dati di processo. Se l'applicazione richiede traffico in entrata, la mancanza di supporto Elastic Load Balancing rende l'esecuzione di questi carichi di lavoro meno efficiente. Amazon ECS ha aggiunto un nuovo tipo di avvio `EXTERNAL` che è possibile utilizzare per creare servizi o eseguire processi sulle istanze esterne.

Di seguito viene fornita una panoramica dell'architettura di sistema di alto livello di Amazon ECS Anywhere. Sul tuo server locale sono installati sia l'agente Amazon ECS che l'agente SSM.



Sistemi operativi e architetture di sistema supportati

Di seguito è riportato l'elenco dei sistemi operativi supportati e delle architetture di sistema.

- Amazon Linux 2
- CentOS 7
- CentOS Stream 8
- RHEL 7, RHEL 8: né i repository di pacchetti aperti di Docker né quelli di RHEL supportano l'installazione nativa di Docker su RHEL. Prima di eseguire lo script di installazione descritto in questo documento assicurati che Docker sia installato.
- Fedora 32, Fedora 33

- openSUSE Tumbleweed
- Ubuntu 18, Ubuntu 20, Ubuntu 22
- Debian 10

> **Important**
>
> Il supporto a lungo termine di Debian 9 (supporto LTS) è terminato il 30 giugno 2022 e non è più supportato da Amazon ECS Anywhere.

- Debian 11
- Debian 12 — Il NVIDIA Container Toolkit non è attualmente supportato su Debian 12. Non sarà possibile eseguire GPU su istanze di Debian 12.
- SUSE Enterprise Server 15
- Le architetture CPU `x86_64` e `ARM64` sono supportate.
- Sono supportate le seguenti versioni dei sistemi operativi Windows:
  - Windows Server 2022
  - Windows Server 2019
  - Windows Server 2016
  - Windows Server 20H2

Considerazioni

Prima di iniziare a utilizzare le istanze esterne, tieni presente le considerazioni riportate di seguito.

- Puoi registrare un'istanza esterna in un cluster alla volta. Per le istruzioni su come registrare un'istanza esterna con un cluster diverso, consulta Annullamento della registrazione di un'istanza esterna Amazon ECS.
- Le tue istanze esterne richiedono un ruolo IAM che consenta loro di comunicare con le API. AWS Per ulteriori informazioni, consulta Ruolo IAM di Amazon ECS Anywhere.
- Le istanze esterne non devono avere una catena di credenziali di istanze preconfigurata definita in locale in quanto ciò interferirà con lo script di registrazione.
- Per inviare i log dei container a CloudWatch Logs, assicurati di creare e specificare un ruolo IAM per l'esecuzione delle attività nella definizione dell'attività.

- Quando un'istanza esterna viene registrata in un cluster, l'attributo `ecs.capability.external` è associato all'istanza. Questo attributo identificherà l'istanza come istanza esterna. Puoi aggiungere attributi personalizzati alle istanze esterne da utilizzare come vincolo di posizionamento dei processi. Per ulteriori informazioni, consulta Attributi personalizzati.
- È possibile aggiungere tag di risorsa all'istanza esterna. Per ulteriori informazioni, consulta Istanze di container esterni.
- ECS Exec è supportato sulle istanze esterne. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS con ECS Exec.
- Di seguito sono riportate considerazioni aggiuntive specifiche per la rete con le istanze esterne. Per ulteriori informazioni, consulta Rete .
  - Il bilanciamento del carico dei servizi non è supportato.
  - L'individuazione dei servizi non è supportata.
  - I processi eseguiti su istanze esterne devono utilizzare le modalità di rete `bridge`, `host` oppure `none`. La modalità di rete `awsvpc` non è supportata.
  - Esistono domini di servizio Amazon ECS in ogni AWS regione. Questi domini di servizio devono essere autorizzati a inviare traffico alle istanze esterne.
  - SSM Agent installato nell'istanza esterna mantiene le credenziali IAM che vengono ruotate ogni 30 minuti utilizzando un'impronta digitale hardware. Se l'istanza esterna perde la connessione a AWS, l'agente SSM aggiorna automaticamente le credenziali dopo aver ristabilito la connessione. Per ulteriori informazioni, consulta Convalida di server on-premise e macchine virtuali con un'impronta digitale hardware nella Guida per l'utente di AWS Systems Manager .
- L'API `UpdateContainerAgent` non è supportata. Per istruzioni su come aggiornare SSM Agent o l'agente Amazon ECS nelle istanze esterne, consulta Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS su un'istanza esterna.
- I provider di capacità Amazon ECS non sono supportati. Per creare un servizio o eseguire un processo autonomo nelle istanze esterne, utilizza il tipo di avvio `EXTERNAL`.
- SELinux non è supportato.
- L'uso di volumi Amazon EFS o la specifica di un `EFSVolumeConfiguration`non sono supportati.
- L'integrazione con App Mesh non è supportata.
- Se utilizzi la console per creare una definizione delle attività di istanza esterna, devi creare la definizione delle attività con l'editor JSON della console.
- Quando esegui ECS Anywhere su Windows, devi utilizzare la tua licenza Windows sull'infrastruttura on-premise.

- Quando utilizzi un'AMI non ottimizzata per Amazon ECS, esegui i seguenti comandi sull'istanza di container esterna per configurare le regole per utilizzare i ruoli IAM per le attività. Per ulteriori informazioni, consulta Configurazione aggiuntiva dell'istanza esterna.

```
$ sysctl -w net.ipv4.conf.all.route_localnet=1
$ iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-
destination 127.0.0.1:51679
$ iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --
to-ports 51679
```

Rete

Le istanze esterne di Amazon ECS sono ottimizzate per l'esecuzione di applicazioni che generano traffico in uscita o dati di processo. Se l'applicazione richiede traffico in entrata, ad esempio un servizio Web, la mancanza di supporto per Elastic Load Balancing rende l'esecuzione di questi carichi di lavoro meno efficiente perché non è disponibile il supporto per il posizionamento di questi carichi di lavoro dietro un load balancer.

Di seguito sono riportate considerazioni aggiuntive specifiche per la rete con le istanze esterne.

- Il bilanciamento del carico dei servizi non è supportato.
- L'individuazione dei servizi non è supportata.
- I processi Linux eseguiti su istanze esterne devono utilizzare le modalità di rete `bridge`, `host` oppure `none`. La modalità di rete `awsvpc` non è supportata.

  Per ulteriori informazioni su ogni modalità di rete, consulta Scelta di una modalità di rete nella Guida alle best practice di Amazon ECS.
- I processi Windows eseguiti su istanze esterne devono utilizzare le modalità di rete `default`.
- Esistono domini del servizio Amazon ECS in ogni regione e devi disporre dell'autorizzazione per inviare traffico alle istanze esterne.
- SSM Agent installato nell'istanza esterna mantiene le credenziali IAM che vengono ruotate ogni 30 minuti utilizzando un'impronta digitale hardware. Se l'istanza esterna perde la connessione a AWS, l'agente SSM aggiorna automaticamente le credenziali dopo che la connessione è stata ristabilita. Per ulteriori informazioni, consulta Convalida di server on-premise e macchine virtuali con un'impronta digitale hardware nella Guida per l'utente di AWS Systems Manager .

I seguenti domini vengono utilizzati per la comunicazione tra il servizio Amazon ECS e l'agente Amazon ECS installato nell'istanza esterna. Assicurati che il traffico sia consentito e che la risoluzione DNS funzioni. Per ogni endpoint *region* rappresenta l'identificatore di regione per una regione AWS 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 di servizio per. AWS Systems Manager
- `ec2messages.`*region*`.amazonaws.com`— Questo è l'endpoint di servizio AWS Systems Manager utilizzato per comunicare tra l'agente Systems Manager e il servizio Systems Manager nel cloud.
- `ssmmessages.`*region*`.amazonaws.com`: questo endpoint del servizio necessario per creare ed eliminare canali di sessione con il servizio Session Manager nel cloud.
- Se le tue attività richiedono la comunicazione con altri AWS servizi, assicurati che tali endpoint di servizio siano consentiti. Le applicazioni di esempio includono l'uso di Amazon ECR per estrarre le immagini dei contenitori o l'utilizzo CloudWatch per i CloudWatch log. Per ulteriori informazioni, consulta Endpoint e quote in Riferimenti generali di AWS .

Amazon FSx for Windows File Server con ECS Anywhere

Per utilizzarlo Amazon FSx for Windows File Server con le istanze esterne di Amazon ECS, devi stabilire una connessione tra il tuo data center locale e il. Cloud AWS Per maggiori informazioni sulle opzioni per connettere la rete al VPC, consulta Opzioni di connettività di Amazon Virtual Private Cloud.

gMSA con ECS Anywhere

Sono supportati i seguenti casi d'uso per ECS Anywhere.

- Active Directory è in Cloud AWS : per questa configurazione, crei una connessione tra la rete locale e l' Cloud AWS utilizzo di una connessione. AWS Direct Connect Per informazioni su

come creare la connessione, consulta Opzioni di connettività di Amazon Virtual Private Cloud. Crei un Active Directory nel Cloud AWS. Per informazioni su come iniziare AWS Directory Service, consulta Configurazione AWS Directory Service nella Guida all'AWS Directory Service amministrazione. È quindi possibile aggiungere le istanze esterne al dominio utilizzando la AWS Direct Connect connessione. Per informazioni sull'utilizzo di GMSA con Amazon ECS, consulta the section called “Scopri come usare GMSAS per contenitori EC2 Windows”.

- Active Directory si trova nel data center on-premise. - Per questa configurazione, unisci le istanze esterne all'Active Directory on-premise. Quindi utilizzi le credenziali disponibili localmente quando esegui i processi di Amazon ECS.

Creazione di un cluster Amazon ECS per il tipo di avvio esterno

Puoi creare un cluster Amazon ECS utilizzando la console Amazon ECS. Prima di iniziare, accertati di aver completato le fasi in Configurazione per l'uso di Amazon ECS e assegna l'autorizzazione IAM corretta. Per ulteriori informazioni, consulta the section called “Esempi di cluster Amazon ECS”. La console Amazon ECS offre un modo semplice per creare le risorse necessarie a un cluster Amazon ECS creando uno AWS CloudFormation stack.

Per rendere il processo di creazione del cluster il più semplice possibile, la console dispone di selezioni predefinite per molte scelte che descriviamo di seguito. Ci sono anche pannelli di aiuto disponibili per la maggior parte delle sezioni della console che forniscono ulteriore contesto.

- Crea uno spazio dei nomi predefinito con lo stesso nome del cluster. AWS Cloud Map Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive.

  Per ulteriori informazioni, consulta Interconnetti i servizi Amazon ECS.

È possibile modificare le seguenti opzioni:

- Modifica lo spazio dei nomi predefinito associato al cluster.

  Uno spazio dei nomi consente ai servizi creati nel cluster di connettersi agli altri servizi nello spazio dei nomi senza configurazioni aggiuntive. Lo spazio dei nomi predefinito ha lo stesso nome del cluster. Per ulteriori informazioni, consulta Interconnetti i servizi Amazon ECS.
- Configurazione del cluster per le istanze esterne
- Attiva Container Insights.

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights”.

- Aggiungi tag per facilitare l'identificazione del cluster.

Creazione di un nuovo cluster (console Amazon ECS)

1. Apri la console all'indirizzo 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 Configurazione del cluster, configura gli elementi seguenti:
   - In Nome cluster, inserisci un nome univoco.

     Il nome può contenere fino a 255 lettere (maiuscole e minuscole), numeri e trattini.
   - (Facoltativo) Per fare in modo che lo spazio dei nomi utilizzato per Service Connect sia diverso dal nome del cluster, in Spazio dei nomi, inserisci un nome univoco.
6. Espandi Infrastruttura, seleziona AWS Fargate (serverless).
7. (Facoltativo) Per attivare Container Insights, espandi Monitoring (Monitoraggio) e poi attiva Use Container Insights (Usa Container Insights).
8. (Facoltativo) Per identificare il tuo cluster, espandi la sezione Tags (Tag), quindi configura i tag.

   [Aggiungere un tag] Scegliere Add tag (Aggiungi tag) e procedere come segue:
   - In Chiave, immetti il nome della chiave.
   - In Valore, immetti il valore della chiave.
9. Scegli Crea.

Passaggi successivi

È necessario registrare le istanze nel cluster. Per ulteriori informazioni, consulta Registrazione di un'istanza esterna in un cluster Amazon ECS.

Dopo aver creato il cluster, è possibile creare le definizioni delle attività per le applicazioni e quindi eseguirle come attività autonome o come parte di un servizio. Per ulteriori informazioni, consulta gli argomenti seguenti:

- Definizioni dei processi di Amazon ECS
- Esecuzione di un'applicazione come attività Amazon ECS
- Creazione di un servizio Amazon ECS utilizzando la console

Registrazione di un'istanza esterna in un cluster Amazon ECS

Per ogni istanza esterna che si registra con un cluster Amazon ECS, è necessario che sia installato SSM Agent, l'agente del container di Amazon ECS e Docker. Per registrare l'istanza esterna in un cluster Amazon ECS, deve prima essere registrata come istanza AWS Systems Manager gestita. Puoi creare lo script di installazione in pochi clic dalla console Amazon ECS. Lo script di installazione include una chiave di attivazione di Systems Manager e i comandi per installare ciascuno degli agenti richiesti e Docker. Per completare le fasi di installazione e registrazione, è necessario eseguire lo script di installazione sul server on-premise o sulla macchina virtuale.

> **Note**
>
> Prima di registrare l'istanza esterna Linux con il cluster, crea il file `/etc/ecs/ecs.config` sull'istanza esterna e aggiungi i parametri di configurazione dell'agente del container desiderati. Non è possibile eseguire questa operazione dopo aver registrato l'istanza esterna in un cluster. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

AWS Management Console

1. Apri la console all'indirizzo 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. Sulla pagina Cluster scegli un cluster in cui registrare l'istanza esterna.
5. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (Infrastruttura).
6. Alla pagina Register external instances (Registra istanze esterne), completa la procedura seguente.

a. Per Durata della chiave di attivazione (in giorni)specifica il numero di giorni per cui la chiave di attivazione rimane attiva. Una volta passati i giorni specificati, la chiave non funzionerà più quando si registra un'istanza esterna.

b. Per Numero di istanze specifica il numero di istanze esterne che si desidera registrare nel cluster con la chiave di attivazione.

c. Per Ruolo dell'istanza, scegli il ruolo IAM da associare alle istanze esterne. Se un ruolo non è già stato creato, scegli Crea nuovo ruolo perché Amazon ECS crei un ruolo per tuo conto. Per ulteriori informazioni sulle autorizzazioni IAM necessarie per le istanze esterne, consulta Ruolo IAM di Amazon ECS Anywhere.

d. Copia il comando di registrazione. Questo comando deve essere eseguito su ogni istanza esterna che si desidera registrare nel cluster.

> **Important**
>
> La parte bash dello script deve essere eseguita come root. Se il comando non viene eseguito come root, viene restituito un errore.

e. Scegli Chiudi.

AWS CLI for Linux operating systems

1. Crea una coppia di attivazione di Systems Manager. Viene utilizzato per l'attivazione dell'istanza gestita di Systems Manager. L'output include un `ActivationId` e un `ActivationCode`. Utilizzerai questa funzione in una fase successiva. Assicurati di specificare il ruolo IAM ECS Anywhere creato. Per ulteriori informazioni, consulta Ruolo IAM di Amazon ECS Anywhere.

```
aws ssm create-activation --iam-role ecsAnywhereRole | tee ssm-activation.json
```

2. Scarica lo script di installazione sul server on-premise o nella macchina virtuale (VM).

```
curl --proto "https" -o "/tmp/ecs-anywhere-install.sh" "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install-latest.sh"
```

3. (Facoltativo) Sul server on-premise o nella macchina virtuale (VM), completa la procedura seguente per verificare lo script di installazione utilizzando il file di firma dello script.

a. Scarica e installa GnuPG. Per ulteriori informazioni su GnuPG, consulta il sito Web 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-premise o nella macchina virtuale (VM). Specifica il nome del cluster, la regione, l'ID di attivazione e il codice di attivazione di Systems Manager dalla prima fase.

```
sudo bash /tmp/ecs-anywhere-install.sh \
    --region $REGION \
    --cluster $CLUSTER_NAME \
    --activation-id $ACTIVATION_ID \
    --activation-code $ACTIVATION_CODE
```

Per un server on-premise o una macchina virtuale (VM) con il driver NVIDIA installato per i carichi di lavoro della GPU, è necessario aggiungere il flag `--enable-gpu` allo script di installazione. Quando viene specificato questo flag, lo script di installazione verifica che il driver NVIDIA sia in esecuzione e quindi aggiunge le variabili di configurazione necessarie per eseguire i processi Amazon ECS. Per ulteriori informazioni sull'esecuzione dei carichi di lavoro della GPU e sulla definizione dei requisiti della GPU in una definizione di attività, consulta Specificazione delle GPU in una definizione di attività Amazon ECS.

```
sudo bash /tmp/ecs-anywhere-install.sh \
    --region $REGION \
    --cluster $CLUSTER_NAME \
    --activation-id $ACTIVATION_ID \
    --activation-code $ACTIVATION_CODE \
    --enable-gpu
```

Utilizza la seguente procedura per registrare un'istanza esterna esistente con un cluster diverso.

Come registrare un'istanza esterna esistente con un cluster diverso

1. Arresta l'agente del container di Amazon ECS.

```
sudo systemctl stop ecs.service
```

2. Modifica il file `/etc/ecs/ecs.config` e alla riga `ECS_CLUSTER`, assicurati che il nome del cluster corrisponda al nome del cluster con cui registrare l'istanza esterna.
3. Rimuovi i dati esistenti dell'agente Amazon ECS.

```
sudo rm /var/lib/ecs/data/agent.db
```

4. Avvia l'agente del container di Amazon ECS.

```
sudo systemctl start ecs.service
```

AWS CLI for Windows operating systems

1. Crea una coppia di attivazione di Systems Manager. Viene utilizzato per l'attivazione dell'istanza gestita di Systems Manager. L'output include un `ActivationId` e un

`ActivationCode`. Utilizzerai questa funzione in una fase successiva. Assicurati di specificare il ruolo IAM ECS Anywhere creato. Per ulteriori informazioni, consulta Ruolo IAM di Amazon ECS Anywhere.

```
aws ssm create-activation --iam-role ecsAnywhereRole | tee ssm-activation.json
```

2. Scarica lo script di installazione sul server on-premise o nella macchina virtuale (VM).

```
Invoke-RestMethod -URI "https://amazon-ecs-agent.s3.amazonaws.com/ecs-anywhere-install.ps1" -OutFile “ecs-anywhere-install.ps1”
```

3. (Facoltativo) Lo script Powershell è firmato da Amazon e pertanto Windows esegue automaticamente la convalida del certificato sullo stesso. Non è necessario eseguire alcuna convalida manuale.

   Per verificare manualmente il certificato, fai clic con il pulsante destro del mouse sul file, accedi alle proprietà e utilizza la scheda Digital Signatures (Firme digitali) per ottenere ulteriori dettagli.

   Questa opzione è disponibile solo quando l'host ha il certificato nell'archivio dei certificati.

   La verifica dovrebbe restituire informazioni simili alle seguenti:

```
# Verification (PowerShell)
Get-AuthenticodeSignature -FilePath .\ecs-anywhere-install.ps1

SignerCertificate                         Status      Path
-----------------                         ------      ----
EXAMPLECERTIFICATE                        Valid       ecs-anywhere-install.ps1

...

Subject              : CN="Amazon Web Services, Inc.",...

----
```

4. Esegui lo script di installazione sul server on-premise o nella macchina virtuale (VM). Specifica il nome del cluster, la regione, l'ID di attivazione e il codice di attivazione di Systems Manager dalla prima fase.

```
.\ecs-anywhere-install.ps1 -Region $Region -Cluster $Cluster -
ActivationID $ActivationID -ActivationCode $ActivationCode
```

5. Verifica che l'agente del container Amazon ECS sia in esecuzione.

```
Get-Service AmazonECS

Status   Name               DisplayName
------   ----               -----------
Running  AmazonECS          Amazon ECS
```

Utilizza la seguente procedura per registrare un'istanza esterna esistente con un cluster diverso.

Come registrare un'istanza esterna esistente con un cluster diverso

1. Arresta l'agente del container di Amazon ECS.

```
Stop-Service AmazonECS
```

2. Modifica il parametro `ECS_CLUSTER` in modo che il nome del cluster corrisponda al nome del cluster con cui registrare l'istanza esterna.

```
[Environment]::SetEnvironmentVariable("ECS_CLUSTER", $ECSCluster,
 [System.EnvironmentVariableTarget]::Machine)
```

3. Rimuovi i dati esistenti dell'agente Amazon ECS.

```
Remove-Item -Recurse -Force $env:ProgramData\Amazon\ECS\data\*
```

4. Avvia l'agente del container di Amazon ECS.

```
Start-Service AmazonECS
```

AWS CLI Può essere utilizzato per creare un'attivazione di Systems Manager prima di eseguire lo script di installazione per completare il processo di registrazione dell'istanza esterna.

Annullamento della registrazione di un'istanza esterna Amazon ECS

Ti consigliamo di annullare la registrazione dell'istanza sia da Amazon ECS che AWS Systems Manager dopo aver 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. È quindi possibile registrarla in un nuovo cluster.

Procedura

AWS Management Console

1. Apri la console all'indirizzo 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. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (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 pagina Istanza di container: ***id***, scegli Annulla registrazione.
7. Esamina il messaggio di annullamento della registrazione. Seleziona Annulla registrazione da AWS Systems Manager per annullare la registrazione dell'istanza esterna come istanza gestita da Systems Manager. Scegli Annulla registrazione.

> **Note**
>
> È possibile annullare la registrazione dell'istanza esterna come istanza gestita da Systems Manager nella console di Systems Manager. Per istruzioni, consulta Annullamento della registrazione di istanze gestite 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.
	`sudo systemctl stop ecs amazon-ssm-agent`
	b. Rimuovi i pacchetti Amazon ECS e Systems Manager.
	Per CentOS 7, CentOS 8 e RHEL 7
	`sudo yum remove -y amazon-ecs-init amazon-ssm-agent`
	Per SUSE Enterprise Server 15
	`sudo zypper remove -y amazon-ecs-init amazon-ssm-agent`
	Per Debian e Ubuntu
	`sudo apt remove -y amazon-ecs-init amazon-ssm-agent`
	c. Rimuovi le directory rimanenti.	

Sistema operativo	Fasi
	`sudo rm -rf /var/lib/ecs /etc/ecs /var/lib/amazon/ssm /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.
	`Stop-Service AmazonECS`
	`Stop-Service AmazonSSMAgent`
	b. Rimuovi il pacchetto Amazon ECS.
	`.\ecs-anywhere-install.ps1 -Uninstall`

AWS CLI

1. Per annullare la registrazione dell'istanza di container, sono necessari l'ID istanza e l'ARN dell'istanza di container. Se non disponi di questi valori, esegui i comandi seguenti

   Esegui il comando seguente per ottenere l'ID istanza.

   Utilizza l'ID dell'istanza (`instanceID`) per ottenere l'ARN dell'istanza di container (`containerInstanceARN`).

```
instanceId=$(aws ssm describe-instance-information --region "{{ region }}" |
 jq ".InstanceInformationList[] |select(.IPAddress==\"{{ IPv4 Address }}\")
 | .InstanceId" | tr -d'"'
```

Esegui i comandi seguenti.

Utilizza `containerInstanceArn`come parametro nel comando per annullare la registrazione dell'istanza (`deregister-container-instance`).

```
instances=$(aws ecs list-container-instances --cluster "{{ cluster }}" --region
 "{{ region }}" | jq -c '.containerInstanceArns')
containerInstanceArn=$(aws ecs describe-container-instances --cluster
 "{{ cluster }}" --region "{{ region }}" --container-instances $instances
 | jq ".containerInstances[] | select(.ec2InstanceId==\"{{ instanceId }}\")
 | .containerInstanceArn" | tr -d '"')
```

2. Esegui il seguente comando per svuotare l'istanza.

```
aws ecs update-container-instances-state --cluster "{{ cluster }}" --region
 "{{ region }}" --container-instances "{{ containerInstanceArn }}" --status
 DRAINING
```

3. Al termine dell'operazione, esegui il comando seguente per annullare la registrazione dell'istanza.

```
aws ecs deregister-container-instance --cluster "{{ cluster }}" --region
 "{{ region }}" --container-instance "{{ containerInstanceArn }}"
```

4. Esegui il comando seguente per rimuovere l'istanza di container da SSM.

```
aws ssm deregister-managed-instance --region "{{ region }}" --instance-id
 "{{ instanceId }}"
```

5. Dopo aver annullato la registrazione dell'istanza, pulisci le AWS risorse sul server o sulla macchina virtuale locale.

Sistema operativo	Fasi	
Linux	a. Arresta l'agente del container di Amazon	

Sistema operativo	Fasi	

	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/ssm /var/log/ecs /var/log/amazon/ssm`	

Sistema operativo	Fasi

Windows	a. Arresta l'agente del container di Amazon ECS e i servizi di SSM Agent sull'istanza.
	`Stop-Service AmazonECS`
	`Stop-Service AmazonSSMAgent`
	b. Rimuovi il pacchetto Amazon ECS.
	`.\ecs-anywhere-install.ps1 -Uninstall`

Aggiornamento dell' AWS Systems Manager agente e dell'agente container Amazon ECS su un'istanza esterna

Il server o la macchina virtuale locale deve eseguire sia l' AWS Systems Manager agente (agente SSM) che l'agente contenitore Amazon ECS durante l'esecuzione di carichi di lavoro Amazon ECS. AWS rilascia nuove versioni di questi agenti ogni volta che vengono aggiunte o aggiornate funzionalità. Se le istanze esterne utilizzano una versione precedente di entrambi gli agenti, è possibile aggiornarle utilizzando le procedure riportate di seguito.

Aggiornamento di SSM Agent su un'istanza esterna

AWS Systems Manager consiglia di automatizzare il processo di aggiornamento dell'agente SSM sulle istanze. Forniscono diversi metodi per automatizzare gli aggiornamenti. Per ulteriori informazioni, consulta Automazione degli aggiornamenti a SSM Agent nella Guida per l'utente di AWS Systems Manager .

Aggiornamento dell'agente di Amazon ECS su un'istanza esterna

Nelle istanze esterne, l'agente del container Amazon ECS viene aggiornato aggiornando il pacchetto `ecs-init`. L'aggiornamento dell'agente di Amazon ECS non interrompe i processi o i servizi in esecuzione. Amazon ECS fornisce il pacchetto `ecs-init` e il file di firma in un bucket Amazon S3 in ciascuna regione. A cominciare da `ecs-init` versione `1.52.1-1`, Amazon ECS fornisce pacchetti `ecs-init` da utilizzare in base al sistema operativo e all'architettura di sistema utilizzata dall'istanza esterna.

Utilizza la seguente tabella per determinare il pacchetto `ecs-init` da scaricare in base al sistema operativo e all'architettura di sistema utilizzata dall'istanza esterna.

> **Note**
>
> È possibile determinare il sistema operativo e l'architettura di sistema utilizzati dall'istanza esterna utilizzando i comandi riportati di seguito.
>
> ```
> cat /etc/os-release
> uname -m
> ```

Sistemi operativi (architettura)	Pacchetto ecs-init
CentOS 7 (x86_64) CentOS 8 (x86_64) SUSE Enterprise Server 15 (x86_64) RHEL 7 (x86_64) RHEL 8 (x86_64)	`amazon-ecs-init-latest.x86_64.rpm`
CentOS 7 (aarch64) CentOS 8 (aarch64) RHEL 7 (aarch64)	`amazon-ecs-init-latest.aarch64.rpm`
Debian 9 (x86_64)	`amazon-ecs-init-latest.amd64.deb`

Sistemi operativi (architettura)	Pacchetto ecs-init
Debian 10 (x86_64) Debian 11 (x86_64) Debian 12 (x86_64) Ubuntu 18 (x86_64) Ubuntu 20 (x86_64)	
Debian 9 (aarch64) Debian 10 (aarch64) Debian 11 (arch64) Debian 12 (aarch64) Ubuntu 18 (aarch64) Ubuntu 20 (aarch64)	`amazon-ecs-init-latest.arm64.deb`

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'identificatore *`<region>`* 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 ulteriori informazioni su GnuPG, consulta il sito Web 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'identificatore *<region>* nel comando con il nome della regione (ad esempio `us-west-2`) a cui sei geograficamente più vicino.

   amazon-ecs-init-latest.x86_64.rpm

   ```
   curl -o amazon-ecs-init.rpm.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.x86_64.rpm.asc
   ```

   amazon-ecs-init-latest.aarch64.rpm

```
curl -o amazon-ecs-init.rpm.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.aarch64.rpm.asc
```

amazon-ecs-init-latest.amd64.deb

```
curl -o amazon-ecs-init.deb.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.amd64.deb.asc
```

amazon-ecs-init-latest.arm64.deb

```
curl -o amazon-ecs-init.deb.asc https://s3.<region>.amazonaws.com/amazon-ecs-agent-<region>/amazon-ecs-init-latest.arm64.deb.asc
```

d. Verificare il file del pacchetto `ecs-init` usando la chiave.

Per i pacchetti **`rpm`**

```
gpg --verify amazon-ecs-init.rpm.asc ./amazon-ecs-init.rpm
```

Per i pacchetti **`deb`**

```
gpg --verify amazon-ecs-init.deb.asc ./amazon-ecs-init.deb
```

L'output previsto è il seguente:

```
gpg: Signature made Fri 14 May 2021 09:31:36 PM UTC
gpg:                using RSA key 50DECCC4710E61AF
gpg: Good signature from "Amazon ECS <ecs-security@amazon.com>" [unknown]
gpg: WARNING: This key is not certified with a trusted signature!
gpg:          There is no indication that the signature belongs to the owner.
Primary key fingerprint: F34C 3DDA E729 26B0 79BE  AEC6 BCE9 D9A4 2D51 784F
     Subkey fingerprint: D64B B6F9 0CF3 77E9 B5FB  346F 50DE CCC4 710E 61AF
```

4. Installare il pacchetto `ecs-init`.

Per il pacchetto **`rpm`** su CentOS 7, CentOS 8 e RHEL 7

```
sudo yum install -y ./amazon-ecs-init.rpm
```

Per il pacchetto **rpm** su SUSE Enterprise Server 15

```
sudo zypper install -y --allow-unsigned-rpm ./amazon-ecs-init.rpm
```

Per il pacchetto **deb**

```
sudo dpkg -i ./amazon-ecs-init.deb
```

5. Riavvia il servizio `ecs`.

```
sudo systemctl restart ecs
```

6. Verifica che la versione dell'agente Amazon ECS sia stata aggiornata.

```
curl -s 127.0.0.1:51678/v1/metadata | python3 -mjson.tool
```

Aggiornamento di un cluster Amazon ECS

Puoi modificare le seguenti proprietà del cluster:

- Imposta un provider di capacità di default

  Ogni cluster ha uno o più provider di capacità e una strategia di provider di capacità facoltativa. La strategia del provider di capacità determina il modo in cui le attività vengono distribuite tra i provider di capacità del cluster. Quando esegui un'attività autonoma o crei un servizio, puoi utilizzare la strategia del provider di capacità predefinita del cluster o specificare una strategia che sostituisce quella del cluster.
- Attiva Container Insights.

  CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati. Container Insights fornisce inoltre informazioni diagnostiche, ad esempio errori di riavvio del container, che puoi usare per isolare i problemi e risolverli in modo rapido. Per ulteriori informazioni, consulta the section called “Monitora i contenitori Amazon ECS utilizzando Container Insights”.
- Aggiungi tag per facilitare l'identificazione dei cluster.

Procedura

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella pagina Cluster: ***name***, scegli Aggiorna cluster.
5. Per impostare il provider di capacità predefinito, in Strategia predefinita del provider di capacità, scegli Aggiungi altro.

   a. Scegli il provider in Provider di capacità.
   b. (Facoltativo) Per Base, inserisci il numero minimo di attività eseguite sul provider di capacità.

      Puoi impostare solo un valore Base per ogni provider di capacità.
   c. (Facoltativo) Per Peso, inserisci la percentuale relativa al numero totale di attività avviate che utilizzano il provider di capacità specificato.
   d. (Facoltativo) Ripeti i passaggi per eventuali provider di capacità aggiuntivi.
6. Per attivare o disattivare Container Insights, espandi Monitoraggio e poi attiva Utilizza Container Insights.
7. Per identificare il tuo cluster, espandi la sezione Tag, quindi configura i tag.

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

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

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

Eliminazione di un cluster Amazon ECS

Al termine dell'utilizzo di un cluster, puoi eliminarlo. Dopo aver eliminato il cluster, lo stato di quest'ultimo diventa `INACTIVE`. I cluster con stato `INACTIVE` potrebbero rimanere individuabili nel tuo account per un periodo di tempo. Tuttavia, questo comportamento è soggetto a modifiche in futuro, quindi non dovresti fare affidamento sulla persistenza dei cluster `INACTIVE`.

Prima di eliminare un cluster, esegui le operazioni seguenti:

- Elimina tutti i servizi del cluster. Per ulteriori informazioni, consulta the section called “Eliminazione di un servizio”.
- Interrompi tutti i processi attualmente in esecuzione. Per ulteriori informazioni, consulta the section called “Interruzione di un'attività”.
- Annulla la registrazione di tutte le istanze del container registrate nel cluster. Per ulteriori informazioni, consulta the section called “Annullamento della registrazione di un'istanza di container”.
- Elimina lo spazio dei nomi . Per ulteriori informazioni, consulta Eliminazione degli spazi dei nomi nella Guida per gli sviluppatori di AWS Cloud Map .

Procedura

1. Apri la console all'indirizzo 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 Cluster, seleziona il cluster da eliminare.
5. In alto a destra della pagina, scegli Elimina Cluster.

   Se non sono state eliminate tutte le risorse associata al cluster, viene visualizzato un messaggio.
6. Nella finestra di conferma, inserisci delete ***cluster name*** (elimina nome cluster).

Creazione di un provider di capacità per Amazon ECS

Al termine della creazione del cluster, sarà possibile creare un nuovo provider di capacità (gruppo con scalabilità automatica) per il tipo di avvio EC2.

Prima di creare il provider di capacità, è necessario creare un gruppo con scalabilità automatica. Per ulteriori informazioni, consulta Gruppi con scalabilità automatica nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Creazione di un provider di capacità per il cluster (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.

4. Sulla pagina Cluster: ***name*** (Cluster: nome), scegli Infrastructure (Infrastruttura), quindi Create (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'istanza EC2 con attività Amazon ECS in esecuzione venga terminata, seleziona Turn on scaling protection (Attiva la protezione della scalabilità).
    - Per Set target capacity, inserisci il valore target per la CloudWatch metrica utilizzata nella policy di scalabilità di tracciamento degli obiettivi gestita da Amazon ECS.
6. Scegli Crea.

Aggiornamento di un provider di capacità Amazon ECS

Quando si utilizza un gruppo con scalabilità automatica come provider di capacità, è possibile modificare la policy di scalabilità del gruppo.

Per aggiornare un provider di capacità per il cluster (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella pagina Cluster: ***name*** (Cluster: nome), scegli Infrastructure (Infrastruttura), quindi Update (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 le istanze EC2 con attività Amazon ECS in esecuzione vengano terminate, seleziona Attiva protezione di dimensionamento.
- Per Set target capacity, inserisci il valore target per la CloudWatch metrica utilizzata nella policy di scalabilità di tracciamento degli obiettivi gestita da Amazon ECS.

6. Scegli Aggiorna.

Eliminazione di un provider di capacità Amazon ECS

Al termine dell'utilizzo di un provider di capacità del gruppo con dimensionamento automatico, puoi eliminarlo. Dopo l'eliminazione del gruppo, il provider di capacità del gruppo Auto Scaling passa allo stato. `INACTIVE` I provider di capacità con stato `INACTIVE` potrebbero rimanere rilevabili nell'account per un periodo di tempo. Tuttavia, questo comportamento è soggetto a modifiche in futuro, quindi non dovresti fare affidamento sulla persistenza dei provider di capacità `INACTIVE`. Prima di eliminare un provider di capacità del gruppo con scalabilità automatica, il provider di capacità deve essere rimosso dalla strategia del provider di capacità per tutti i servizi. L'API `UpdateService` o il flusso di lavoro del servizio di aggiornamento nella console Amazon ECS possono essere utilizzati per rimuovere un provider di capacità dalla strategia del provider di capacità di un servizio. Utilizza l'opzione Force new deployment per assicurarti che tutte le attività che utilizzano la capacità dell'istanza Amazon EC2 fornita dal provider di capacità vengano trasferite all'utilizzo della capacità dei restanti provider di capacità.

Per eliminare un provider di capacità per il cluster (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Sulla pagina Cluster: ***`name`*** (Cluster: nome), scegli Infrastructure (Infrastruttura), il gruppo con scalabilità automatica quindi Delete (Elimina).
5. Nella casella di conferma, inserisci delete ***`nome del gruppo con scalabilità automatica`***.
6. Scegli Elimina.

Annullamento della registrazione di un'istanza di container Amazon ECS

> **Important**
>
> Questo argomento riguarda solo le istanze del container create in Amazon EC2. Per ulteriori informazioni sull'annullamento delle registrazioni delle istanze gestite, consulta Annullamento della registrazione di un'istanza esterna Amazon ECS.

Una volta terminato di utilizzare un'istanza di container supportata da Amazon EC2, puoi annullarne la registrazione dal cluster. In seguito all'annullamento della registrazione, l'istanza di container non è più in grado di accettare nuove attività.

Se hai attività in esecuzione nell'istanza di container al momento dell'annullamento della registrazione, queste attività continueranno a essere in esecuzione fino al termine dell'istanza o fino a quando le attività non verranno interrotte in altri modi. Tuttavia, questi processi sono orfani (non più monitorati o tenuti in considerazione da Amazon ECS). Se un processo orfano nell'istanza di container è parte di un servizio Amazon ECS, il pianificatore di servizi avvia un'altra copia di tale attività in un'altra istanza di container, se possibile. La registrazione di eventuali container in attività di servizio orfana registrati con un gruppo di destinazione di Application Load Balancer viene annullata. Ha inizio lo svuotamento della connessione in base alle impostazioni nel sistema di bilanciamento del carico o nel gruppo di destinazione. Se un processo orfano utilizza la modalità di rete `awsvpc`, le sue interfacce di rete elastiche vengono eliminate.

Se hai intenzione di utilizzare l'istanza di container per altri motivi in seguito all'annullamento della registrazione, devi interrompere tutte le attività in esecuzione nell'istanza di container prima dell'annullamento. Questa operazione interrompe il consumo di risorse da parte delle attività orfane.

Quando annulli la registrazione di un'istanza di container, tieni presente le considerazioni riportate di seguito.

- Dato che ogni istanza di container presenta informazioni sullo stato univoche, non deve essere annullata da un cluster e registrata nuovamente in un altro. Per riposizionare le risorse dell'istanza di container, ti consigliamo di terminare le istanze di container da un cluster e avviare nuove istanze di container nel nuovo cluster. Per ulteriori informazioni, consulta Terminare l'istanza nella Guida per l'utente di Amazon EC2 e. Avvio di un'istanza di container Linux di Amazon ECS

- Se l'istanza del contenitore è gestita da un gruppo o da uno AWS CloudFormation stack di Auto Scaling, interrompi l'istanza aggiornando il gruppo o lo stack Auto Scaling. AWS CloudFormation In caso contrario, il gruppo con dimensionamento automatico o AWS CloudFormation creerà una nuova istanza dopo averla terminata.
- Se termini un'istanza di container in esecuzione con un agente del container Amazon ECS connesso, l'agente annulla automaticamente la registrazione dell'istanza dal cluster. Non viene automaticamente annullata la registrazione delle istanze di container interrotte o delle istanze con agenti disconnessi al loro termine.
- L'annullamento della registrazione di un'istanza di container rimuove l'istanza da un cluster, ma non termina l'istanza Amazon EC2. Al termine dell'utilizzo dell'istanza, assicurati di terminarla in modo da interrompere la fatturazione. Per ulteriori informazioni, consulta la sezione relativa alla terminazione dell'istanza nella Guida per l'utente di Amazon EC2.

Procedura

1. Apri la console all'indirizzo 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, scegli Clusters (Cluster) e seleziona il cluster che ospita l'istanza.
4. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (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 pagina Istanza di container: ***id***, scegli Annulla registrazione.
7. Nella schermata di conferma, seleziona Annulla registrazione.
8. Se un'istanza di container non è più necessaria, termina l'istanza Amazon EC2 sottostante. Per ulteriori informazioni, consulta Terminate Your Instance nella Amazon EC2 User Guide.

Drenaggio delle istanze di container Amazon ECS

In alcuni casi potrebbe essere necessario rimuovere un'istanza di container dal cluster, ad esempio per eseguire aggiornamenti di sistema o per ridurre la capacità del cluster. Amazon ECS offre la possibilità di passare un'istanza di container a uno stato `DRAINING`. Questa operazione è nota come svuotamento dell'istanza di container. Quando un'istanza di container è impostata su `DRAINING`, Amazon ECS impedisce che venga pianificato il posizionamento di nuovi processi nell'istanza di container.

Comportamento di svuotamento per i servizi

Qualsiasi processo che fa parte di un servizio che si trova in uno stato `PENDING` viene arrestato immediatamente. Se nel cluster è disponibile la capacità dell'istanza di container, il pianificatore di servizi avvierà i processi di sostituzione. Se la capacità dell'istanza di container non è sufficiente, verrà inviato un messaggio di evento del servizio che indica il problema.

I processi che fanno parte di un servizio nell'istanza di container che si trovano in uno stato `RUNNING` passano a uno stato `STOPPED`. Lo scheduler di servizi prova a sostituire le attività in base al tipo di implementazione e ai parametri di configurazione dell'implementazione del servizio, `minimumHealthyPercent` e `maximumPercent`. Per ulteriori informazioni, consulta Servizi Amazon ECS e Parametri di definizione del servizio Amazon ECS.

- Se `minimumHealthyPercent` è inferiore al 100%, il pianificatore può ignorare `desiredCount` temporaneamente durante la sostituzione delle attività. Ad esempio, `desiredCount` sono quattro attività, un minimo del 50% permette al pianificatore di interrompere due attività esistenti prima di avviare due nuove attività. Se il minimo è del 100%, il pianificatore del servizio non può rimuovere le attività esistenti fino a quando le attività di sostituzione non vengono considerate integre.
  Se le attività per i servizi che non utilizzano un load balancer sono in stato `RUNNING`, vengono considerate integre. Le attività per i servizi che utilizzano un load balancer vengono considerate integre se sono in stato `RUNNING` e se il load balancer considera integra l'istanza di container su cui sono ospitate.

  > **Important**
  >
  > Se utilizzi le istanze Spot e `minimumHealthyPercent` è maggiore o uguale al 100%, il servizio non avrà abbastanza tempo per sostituire l'attività prima della cessazione dell'istanza Spot.

- Il parametro `maximumPercent` rappresenta un limite superiore al numero di attività in esecuzione durante la sostituzione delle attività. Ciò permette di definire le dimensioni del batch di sostituzione. Ad esempio, se `desiredCount` di quattro attività, un massimo di 200% avvia quattro nuove attività prima di interrompere le quattro attività affinché vengano esaurite (a condizione che le risorse del cluster necessarie per questa operazione siano disponibili). Se il massimo è 100%, le attività di sostituzione non possono avviarsi fino all'interruzione delle attività di esaurimento.

> **Important**
>
> Se `minimumHealthyPercent` e `maximumPercent` sono entrambi al 100%, il servizio non può rimuovere i processi esistenti e non può inoltre avviare processi di sostituzione. Ciò impedisce il corretto svuotamento delle istanze del container e impedisce la creazione di nuove implementazioni.

Comportamento di svuotamento per processi autonomi

Qualsiasi processo autonomo nello stato `PENDING` o `RUNNING` non ne è influenzato; devi attenderne l'interruzione o interromperlo manualmente. L'istanza di container rimarrà nello `DRAINING` stato.

Un'istanza di container ha completato lo svuotamento quando tutti i processi in esecuzione sull'istanza passano a uno stato `STOPPED`. L'istanza di container rimane nello stato `DRAINING` finché non viene nuovamente attivata o eliminata. Puoi verificare lo stato delle attività sull'istanza del contenitore utilizzando l'ListTasksoperazione con il `containerInstance` parametro per ottenere un elenco di attività sull'istanza seguita da un'DescribeTasksoperazione con l'Amazon Resource Name (ARN) o l'ID di ciascuna attività per verificare lo stato dell'attività.

Quando desideri che l'istanza di container avvi nuovamente i processi di hosting, puoi modificare lo stato dell'istanza di container da `DRAINING` a `ACTIVE`. Lo scheduler del servizio Amazon ECS considererà nuovamente l'istanza di container per il posizionamento del processo.

Procedura

Le seguenti fasi possono essere utilizzate per impostare lo svuotamento di un'istanza di container utilizzando la nuova AWS Management Console.

Puoi anche utilizzare l'azione UpdateContainerInstancesStateAPI o il comando update-container-instances-state per modificare lo stato di un'istanza del contenitore in. `DRAINING`

AWS Management Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Alla pagina Clusters (Cluster), scegli un cluster che ospita le istanze.

4. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Infrastructure (Infrastruttura). Quindi, in Container instances (Istanze di container) e seleziona la casella di controllo per ciascuna istanza di container che desideri svuotare.
5. Scegli Operazioni, Drain.

Agente del container Linux di Amazon ECS

L'agente Amazon ECS è un processo che viene eseguito su ogni istanza di container registrata nel cluster. Facilita la comunicazione tra le istanze del contenitore e Amazon ECS.

Ogni versione dell'agente del container di Amazon ECS supporta una serie di funzioni differenti e assicura le correzioni dei bug delle versioni precedenti. Quando possibile, consigliamo sempre di utilizzare la versione più recente dell'agente del container di Amazon ECS. Per passare all'ultima versione dell'agente del container, consulta Aggiornamento dell'agente del container Amazon ECS.

Per vedere le caratteristiche e i miglioramenti inclusi in ogni versione dell'agente, consulta https://github.com/aws/amazon-ecs-agent/releases.

> **Important**
>
> La versione Docker minima per parametri affidabili è `v20.10.13` e successive, inclusa nell'AMI `20220607` ottimizzata per Amazon ECS e versioni successive.
> Le versioni dell'agente Amazon ECS `1.20.0` e successive non supportano più le versioni di Docker precedenti alla `1.9.0`.

Ciclo di vita

Quando l'agente del container di Amazon ECS registra un'istanza Amazon EC2 nel cluster, l'istanza Amazon EC2 segnala il suo stato come `ACTIVE` e lo stato di 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 ID dell'istanza di container duplicati per lo stesso ID dell'istanza Amazon EC2, puoi annullare in modo sicuro la registrazione dei duplicati che sono elencati come `ACTIVE` con uno stato di connessione dell'agente `FALSE`). Questo problema è stato risolto nella versione corrente dell'agente del container di Amazon ECS. Per ulteriori informazioni sull'aggiornamento alla versione corrente, consulta Aggiornamento dell'agente del container Amazon ECS.

Se modifichi lo stato di un'istanza di container in `DRAINING`, le nuove attività non vengono posizionate nell'istanza di container. Qualsiasi attività di servizio in esecuzione nell'istanza di container viene rimossa, se possibile, in modo che possano essere eseguiti gli aggiornamenti di sistema. Per ulteriori informazioni, consulta Drenaggio delle istanze di container Amazon ECS.

Se annulli la registrazione di un'istanza di container o la termini, lo stato dell'istanza di container passa immediatamente a `INACTIVE` e l'istanza di container non viene più segnalata quando elenchi le istanze di container. Tuttavia, puoi ancora descrivere l'istanza di container per un'ora in seguito alla terminazione. Dopo un'ora, la descrizione dell'istanza non è più disponibile.

> **Important**
>
> Puoi svuotare manualmente le istanze o creare un hook del ciclo di vita del gruppo Auto Scaling per impostare lo stato dell'istanza su `DRAINING`. Per ulteriori informazioni, consulta Hook del ciclo di vita di Amazon EC2 Auto Scaling.

AMI ottimizzate per Amazon ECS

Le varianti Linux dell'AMI ottimizzata per Amazon ECS utilizzano l'AMI Amazon Linux 2 come base. Il nome dell'AMI di origine di Amazon Linux 2 per ogni variante può essere recuperato interrogando l'API Archivio dei parametri Systems Manager. Per ulteriori informazioni, consulta Recupero di metadati AMI Linux ottimizzati per Amazon ECS. Quando avvii le istanze di container dall'AMI Amazon Linux 2 ottimizzata per Amazon ECS più recente, ricevi la versione dell'agente del container

corrente. Per avviare un'istanza di container con l'AMI Amazon Linux 2 ottimizzata per Amazon ECS, consulta Avvio di un'istanza di container Linux di Amazon ECS.

Informazioni aggiuntive

Nelle pagine seguenti vengono fornite ulteriori informazioni sulle modifiche:

- Registro delle modifiche di Amazon ECS Agent attivo GitHub
- Il codice sorgente dell'applicazione `ecs-init`, gli script e la configurazione per il pacchetto dell'agente fanno ora parte del repository dell'agente. Per le versioni `ecs-init` e i pacchetti precedenti di Amazon ecs-init, consulta il changelog di Amazon ecs-init su GitHub
- Note di rilascio di Amazon Linux 2.
- Note di rilascio di Docker Engine nella documentazione Docker
- Documentazione dei driver NVIDIA nella documentazione NVIDIA

Configurazione dell'agente del container Amazon ECS

L'agente container Amazon ECS supporta una serie di opzioni di configurazione, la maggior parte delle quali impostate tramite variabili di ambiente.

Se l'istanza di container è stata avviata con una variante Linux dell'AMI ottimizzata per Amazon ECS, puoi impostare queste variabili di ambiente nel file `/etc/ecs/ecs.config` e quindi riavviare l'agente. Puoi scrivere queste variabili di configurazione anche nelle istanze di container con i dati utente di Amazon EC2 al momento dell'avvio. Per ulteriori informazioni, consulta Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati.

Se l'istanza del contenitore è stata avviata con una variante Windows dell'AMI ottimizzata per Amazon ECS, puoi impostare queste variabili di ambiente con il PowerShell SetEnvironmentVariable comando e quindi riavviare l'agente. Per ulteriori informazioni, consulta Esegui comandi sull'istanza Windows all'avvio nella Guida per l'utente di Amazon EC2 e. the section called “Avvio delle istanze dei container”

Se procedi all'avvio manuale dell'agente del container Amazon ECS (per le AMI non ottimizzate per Amazon ECS), puoi utilizzare queste variabili di ambiente nel comando docker run utilizzato per avviare l'agente. Utilizza queste variabili con la sintassi `--env=`*`VARIABLE_NAME`*`=`*`VARIABLE_VALUE`*. Nel caso di informazioni sensibili, come ad esempio le credenziali di autenticazione a repository privati, le variabili di ambiente dell'agente vanno archiviate

in un file e trasmesse tutte in una volta con l'opzione `--env-file` *`path_to_env_file`*. Per aggiungere le variabili, puoi utilizzare i seguenti comandi.

```
sudo systemctl stop ecs
sudo vi /etc/ecs/ecs.config
# And add the environment variables with VARIABLE_NAME=VARIABLE_VALUE format.
sudo systemctl start ecs
```

Parametri disponibili

Per informazioni sui parametri di configurazione dell'agente container Amazon ECS disponibili, consulta Amazon ECS Container Agent on. GitHub

Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3

La configurazione dell'agente container Amazon ECS è controllata con la variabile di ambiente. Le varianti Linux dell'AMI ottimizzata per Amazon ECS cercano queste variabili in `/etc/ecs/ecs.config` all'avvio dell'agente del container e lo configurano di conseguenza. Alcune variabili di ambiente innocue, ad esempio `ECS_CLUSTER`, possono essere trasmesse all'istanza di container all'avvio tramite i dati utente di Amazon EC2 ed essere scritte su 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 utente o scritte `/etc/ecs/ecs.config` in un modo che consenta loro di essere visualizzate in un `.bash_history` file.

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 utente di Amazon EC2 per installare AWS CLI e copiare le informazioni di configurazione all'`/etc/ecs/ecs.config`avvio dell'istanza.

Come memorizzare un file **`ecs.config`** in Amazon S3

1. È necessario concedere le autorizzazioni ecs (Container Instance Role InstanceRole) per avere accesso in sola lettura ad Amazon S3. Puoi farlo assegnando AmazonS3 Access al ruolo ReadOnly. `ecsInstanceRole` Per informazioni su come allegare una politica a un ruolo, consulta Modifica della politica di autorizzazione di un ruolo (console) nella Guida per l'utente AWS Identity and Access Management

2. Crea un file `ecs.config` con variabili di configurazione dell'agente Amazon ECS valide utilizzando il seguente formato. In questo esempio viene configurata l'autenticazione di un registro privato. Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.

```
ECS_ENGINE_AUTH_TYPE=dockercfg
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"auth":"zq212MzEXAMPLE7o6T25Dk0i","email":"email@example.com"}}
```

> **Note**
>
> Per un elenco completo delle variabili di configurazione dell'agente Amazon ECS disponibili, consulta Amazon ECS Container Agent on. GitHub

3. Per archiviare il file di configurazione, crea un bucket privato in Amazon S3. Per ulteriori informazioni, consulta Create a Bucket (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 Aggiunta di un oggetto a un bucket 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'istanza Amazon EC2 con il tuo cluster Amazon ECS e tale istanza non utilizza un'AMI basata sull'AMI ottimizzata per Amazon ECS, puoi installare l'agente contenitore

Amazon ECS manualmente utilizzando la seguente procedura. A tale scopo, puoi scaricare l'agente da uno dei bucket Amazon S3 regionali o da Amazon Elastic Container Registry Public. Se scarichi da uno dei bucket regionali di Amazon S3, puoi opzionalmente verificare la validità del file dell'agente contenitore utilizzando la firma PGP.

> **Note**
>
> Le unità `systemd` per i servizi Amazon ECS e Docker hanno l'istruzione di attendere il completamento di `cloud-init` prima di avviare entrambi i servizi. Il processo `cloud-init` non viene considerato terminato fino a quando i dati utente di Amazon EC2 non hanno terminato l'esecuzione. Pertanto, l'avvio di Amazon ECS o Docker tramite i dati utente di Amazon EC2 può causare un deadlock. Per avviare l'agente del container utilizzando i dati utente di Amazon EC2 puoi utilizzare `systemctl enable --now --no-block ecs.service`.

Installazione dell'agente del container Amazon ECS in un'istanza EC2 non Amazon Linux

Per installare l'agente container Amazon ECS su un'istanza Amazon EC2, puoi scaricare l'agente da uno dei bucket Amazon S3 regionali e installarlo.

> **Note**
>
> Quando utilizzi un'AMI non Amazon Linux, l'istanza Amazon EC2 richiede il supporto `cgroupfs` per il driver `cgroup` in modo che l'agente Amazon ECS supporti i limiti delle risorse a livello di attività. Per ulteriori informazioni, consulta Amazon ECS agent on GitHub.

A titolo di riferimento, di seguito sono elencati i file più recenti dell'agente del container Amazon ECS, per regione e per ciascun tipo di architettura di sistema.

Regione	Nome Regione	File di init deb di Amazon ECS	File di init rbm di Amazon ECS
us-east-2	Stati Uniti orientali (Ohio)	Amazon ECS init amd64 (amd64)	Amazon ECS init x86_64 (x86_64)

Regione	Nome Regione	File di init deb di Amazon ECS	File di init rbm di Amazon ECS
		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 arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
us-west-1	Stati Uniti occidentali (California settentrionale)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
us-west-2	US West (Oregon)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-east-1	Asia Pacifico (Hong Kong)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-northeast-1	Asia Pacifico (Tokyo)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)

Regione	Nome Regione	File di init deb di Amazon ECS	File di init rbm di Amazon ECS
ap-northeast-2	Asia Pacifico (Seoul)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-south-1	Asia Pacifico (Mumbai)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-southeast-1	Asia Pacifico (Singapore)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ap-southeast-2	Asia Pacifico (Sydney)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
ca-central-1	Canada (Centrale)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
eu-central-1	Europa (Francoforte)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)

Regione	Nome Regione	File di init deb di Amazon ECS	File di init rbm di Amazon ECS
eu-west-1	Europa (Irlanda)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
eu-west-2	Europa (Londra)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
eu-west-3	Europa (Parigi)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
sa-east-1	Sud America (San Paolo)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 Amazon ECS init aarch64 (aarch64)
us-gov-east-1	AWS GovCloud (Stati Uniti orientali)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)
us-gov-west-1	AWS GovCloud (Stati Uniti occidentali)	Amazon ECS init amd64 (amd64) Amazon ECS init arm64 (arm64)	Amazon ECS init x86_64 (x86_64) Amazon ECS init aarch64 (aarch64)

Per installare l'agente del container Amazon ECS su un'istanza Amazon EC2 utilizzando un'AMI non Amazon Linux

1. Avvia un'istanza Amazon EC2 con un ruolo IAM che consenta l'accesso ad Amazon ECS. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.
2. Connettiti alla tua istanza.
3. Installare la versione più recente di Docker nell'istanza.
4. Controlla la versione Docker per verificare che il tuo sistema soddisfi il requisito di versione minima.

   > **Note**
   >
   > La versione Docker minima per parametri affidabili è `v20.10.13` e successive, inclusa nell'AMI `20220607` ottimizzata per Amazon ECS e versioni successive.
   > Le versioni dell'agente Amazon ECS `1.20.0` e successive non supportano più le versioni di Docker precedenti alla `1.9.0`.

   ```
   docker --version
   ```

5. Scarica il file di agente Amazon ECS appropriato per il sistema operativo e per l'architettura di sistema in uso e installalo.

   Per le architetture `deb`:

   ```
   ubuntu:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-ecs-agent-us-west-2/amazon-ecs-init-latest.amd64.deb
   ubuntu:~$ sudo dpkg -i amazon-ecs-init-latest.amd64.deb
   ```

   Per le architetture `rpm`:

   ```
   fedora:~$ curl -O https://s3.us-west-2.amazonaws.com/amazon-ecs-agent-us-west-2/amazon-ecs-init-latest.x86_64.rpm
   fedora:~$ sudo yum localinstall -y amazon-ecs-init-latest.x86_64.rpm
   ```

6. Modifica il `/lib/systemd/system/ecs.service` file e aggiungi la riga seguente alla fine della `[Unit]` sezione.

```
After=cloud-final.service
```

7. (Facoltativo) Per registrare l'istanza con un cluster diverso dal cluster `default`, modifica il file `/etc/ecs/ecs.config` e aggiungi i contenuti seguenti. L'esempio seguente specifica il cluster `MyCluster`.

   ```
   ECS_CLUSTER=MyCluster
   ```

   Per ulteriori informazioni su queste e altre opzioni di runtime dell'agente, consulta Configurazione dell'agente del container Amazon ECS.

   > **Note**
   >
   > Puoi facoltativamente archiviare le variabili di ambiente dell'agente in Amazon S3 (che possono essere scaricate nelle tue istanze di container all'avvio utilizzando i dati utente di Amazon EC2). Ciò è consigliato per le informazioni sensibili, come le credenziali di autenticazione per i repository privati. Per ulteriori informazioni, consulta Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3 e Utilizzo di immagini non AWS containerizzate in Amazon ECS.

8. Avviare il servizio `ecs`.

   ```
   ubuntu:~$ sudo systemctl start ecs
   ```

Esecuzione dell'agente Amazon ECS con modalità di rete host

Quando si esegue l'agente del container di Amazon ECS, `ecs-init` crea il relativo container dell'agente del container con la modalità di rete `host`. Questa è l'unica modalità di rete supportata per il container dell'agente del container.

Ciò permette di bloccare l'accesso all'endpoint del servizio per i metadati dell'istanza Amazon EC2 (`http://169.254.169.254`) per i container avviati dall'agente di container. Questo garantisce che i container non possano accedere alle credenziali del ruolo IAM dal profilo dell'istanza di container e che le attività utilizzino solo le credenziali per il ruolo del processo IAM. Per ulteriori informazioni, consulta Ruolo IAM dell'attività Amazon ECS.

Assicura inoltre che l'agente del container non debba contendersi le connessioni e il traffico di rete sul bridge `docker0`.

Parametri di configurazione del registro dell'agente container Amazon ECS

L'agente del container di Amazon ECS archivia i log nelle istanze del container.

Per l'agente del container versione 1.36.0 e successive, di default i log si trovano in `/var/log/ecs/ecs-agent.log` sulle istanze Linux e in `C:\ProgramData\Amazon\ECS\log\ecs-agent.log` sulle istanze Windows.

Per l'agente del container versione 1.35.0 e precedenti, di default i log si trovano in `/var/log/ecs/ecs-agent.log.`*`timestamp`* sulle istanze Linux e `C:\ProgramData\Amazon\ECS\log\ecs-agent.log.`*`timestamp`* sulle istanze Windows.

Di default, i log dell'agente vengono ruotati ogni ora per un massimo di 24 log archiviati.

Di seguito sono riportate le variabili di configurazione dell'agente del container che possono essere utilizzate per modificare il comportamento di registrazione dell'agente predefinito. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

`ECS_LOGFILE`

Valori di esempio: `/ecs-agent.log`

Valore predefinito in Linux: Null

Valore predefinito in Windows: Null

Determina la posizione in cui devono essere scritti i log degli agenti. Se stai eseguendo l'agente tramite`ecs-init`, che è il metodo predefinito quando si utilizza l'AMI ottimizzata per Amazon ECS, il percorso all'interno del contenitore è `/log` e lo `ecs-init` monta sull'host. `/var/log/ecs/`

`ECS_LOGLEVEL`

Valori id esempio: `crit`, `error`, `warn`, `info`, `debug`

Valore predefinito in Linux: `info`

Valore predefinito in Windows: `info`

Il livello di dettaglio da registrare.

`ECS_LOGLEVEL_ON_INSTANCE`

Valori di esempio: `none`, `crit`, `error`, `warn`, `info`, `debug`

Valore di default su Linux: `none`, se `ECS_LOG_DRIVER` viene impostato esplicitamente su un valore non vuoto; altrimenti lo stesso valore di `ECS_LOGLEVEL`

Valore di default su Windows: `none`, se `ECS_LOG_DRIVER` viene impostato esplicitamente su un valore non vuoto; altrimenti lo stesso valore di `ECS_LOGLEVEL`

Può essere usato per sovrascrivere `ECS_LOGLEVEL` e impostare un livello di dettaglio da registrare nel file di log nell'istanza, separato dal livello registrato nel driver di registrazione. Se un driver di registrazione è impostato in modo esplicito, i log in istanza sono disattivati per impostazione predefinita. Possono essere riattivati con questa variabile.

`ECS_LOG_DRIVER`

Valori di esempio: `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries` `syslog`, `splunk`

Valore predefinito in Linux: `json-file`

Valore predefinito in Windows: non applicabile

Determina il driver di registrazione utilizzato dal contenitore dell'agente.

`ECS_LOG_ROLLOVER_TYPE`

Valori di esempio: `size` e `hourly`

Valore predefinito in Linux: `hourly`

Valore predefinito in Windows: `hourly`

Determina se il file di registro dell'agente contenitore viene ruotato ogni ora o in base alle dimensioni. Di default, il file di log dell'agente viene ruotato ogni ora.

`ECS_LOG_OUTPUT_FORMAT`

Valori di esempio: `logfmt` e `json`

Valore predefinito in Linux: `logfmt`

Valore predefinito in Windows: `logfmt`

Determina il formato di output del log. Quando viene utilizzato il `json` formato, ogni riga del log è una mappa JSON strutturata.

`ECS_LOG_MAX_FILE_SIZE_MB`

Valori di esempio: `10`

Valore predefinito in Linux: `10`

Valore predefinito in Windows: `10`

Quando la `ECS_LOG_ROLLOVER_TYPE` variabile è impostata su`size`, questa variabile determina la dimensione massima (in MB) del file di registro prima che venga ruotato. Se il tipo di rollover è impostato su `hourly`, questa variabile viene ignorata.

`ECS_LOG_MAX_ROLL_COUNT`

Valori di esempio: `24`

Valore predefinito in Linux: `24`

Valore predefinito in Windows: `24`

Determina il numero di file di log ruotati da mantenere. I file di log più vecchi vengono eliminati una volta raggiunto questo limite.

Per l'agente del container versione 1.36.0 e successive, il seguente è un file di log di esempio quando viene utilizzato il formato `logfmt`.

```
level=info time=2019-12-12T23:43:29Z msg="Loading configuration" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Image excluded from cleanup: amazon/amazon-
ecs-agent:latest" module=parse.go
level=info time=2019-12-12T23:43:29Z msg="Image excluded from cleanup: amazon/amazon-
ecs-pause:0.1.0" module=parse.go
level=info time=2019-12-12T23:43:29Z msg="Amazon ECS agent Version: 1.36.0, Commit:
 ca640387" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Creating root ecs cgroup: /ecs"
 module=init_linux.go
level=info time=2019-12-12T23:43:29Z msg="Creating cgroup /ecs"
 module=cgroup_controller_linux.go
level=info time=2019-12-12T23:43:29Z msg="Loading state!" module=statemanager.go
level=info time=2019-12-12T23:43:29Z msg="Event stream ContainerChange start
 listening..." module=eventstream.go
```

```
level=info time=2019-12-12T23:43:29Z msg="Restored cluster 'auto-robc'" module=agent.go
level=info time=2019-12-12T23:43:29Z msg="Restored from checkpoint file. I
 am running as 'arn:aws:ecs:us-west-2:0123456789:container-instance/auto-
robc/3330a8a91d15464ea30662d5840164cd' in cluster 'auto-robc'" module=agent.go
```

Di seguito è riportato un file di log di esempio quando viene utilizzato il formato JSON.

```
{"time": "2019-11-07T22:52:02Z", "level": "info", "msg": "Starting Amazon Elastic
 Container Service Agent", "module": "engine.go"}
```

Per le versioni dell'agente del container 1.35.0 e precedenti, il seguente è il formato del file di log.

```
2016-08-15T15:54:41Z [INFO] Starting Agent: Amazon ECS Agent - v1.12.0 (895f3c1)
2016-08-15T15:54:41Z [INFO] Loading configuration
2016-08-15T15:54:41Z [WARN] Invalid value for task cleanup duration, will be overridden
 to 3h0m0s, parsed value 0, minimum threshold 1m0s
2016-08-15T15:54:41Z [INFO] Checkpointing is enabled. Attempting to load state
2016-08-15T15:54:41Z [INFO] Loading state! module="statemanager"
2016-08-15T15:54:41Z [INFO] Detected Docker versions [1.17 1.18 1.19 1.20 1.21 1.22]
2016-08-15T15:54:41Z [INFO] Registering Instance with ECS
2016-08-15T15:54:41Z [INFO] Registered! module="api client"
```

Configurazione delle istanze di container Amazon ECS per immagini Docker private

L'agente di container di Amazon ECS è in grado di eseguire l'autenticazione con registri privati utilizzando l'autenticazione di base. Quando si abilita l'autenticazione dei registri privati, puoi utilizzare le immagini Docker private nelle tue definizioni di processo. Questa funzionalità è supportata solo dai processi che utilizzano il tipo di avvio EC2.

Un altro metodo per abilitare l'autenticazione del registro privato consiste nell' AWS Secrets Manager 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 i processi utilizzando i tipi di avvio EC2 o Fargate. Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.

L'agente del container di Amazon ECS all'avvio ricerca due variabili di ambiente:

- `ECS_ENGINE_AUTH_TYPE`, che specifica il tipo di dati di autenticazione inviati.
- `ECS_ENGINE_AUTH_DATA`, che contiene le credenziali di autenticazione effettive.

Le varianti Linux della scansione dell'AMI ottimizzata per Amazon ECS eseguono la scansione del file `/etc/ecs/ecs.config` alla ricerca di queste variabili all'avvio dell'istanza di container e ogni volta che si avvia il servizio (con il comando sudo start ecs). Le AMI che non ottimizzate per Amazon ECS devono archiviare queste variabili di ambiente in un file e trasferirle con l'opzione `--env-file` *`path_to_env_file`* al comando docker run che avvia l'agente del container.

> **Important**
>
> Consigliamo di non inserire queste variabili di ambiente di autenticazione al momento dell'avvio dell'istanza con i dati utente di Amazon EC2 o di trasferirle con l'opzione `--env` al comando docker run. Questi metodi non sono ideali per i dati sensibili, come le credenziali di autenticazione. Per informazioni sull'aggiunta sicura delle credenziali di autenticazione alle istanze di container, consulta Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3.

Formati di autenticazione

Sono disponibili due formati per l'autenticazione di registri privati, `dockercfg` e `docker`.

Formato di autenticazione dockercfg

Il formato `dockercfg` utilizza le informazioni di autenticazione archiviate nel file di configurazione che viene creato quando esegui il comando docker login. Puoi creare questo file eseguendo il comando docker login sul sistema locale e inserendo il nome utente, la password e l'indirizzo e-mail del registro. Puoi inoltre effettuare l'accesso a un'istanza di container ed eseguire il comando qui. A seconda della tua versione Docker, questo file viene salvato come `~/.dockercfg` o `~/.docker/config.json`.

```
cat ~/.docker/config.json
```

Output:

```
{
  "auths": {
    "https://index.docker.io/v1/": {
      "auth": "zq212MzEXAMPLE7o6T25Dk0i"
    }
  }
```

```
}
```

> **Important**
>
> Le nuove versioni di Docker creano un file di configurazione come mostrato in precedenza con un oggetto `auths` esterno. L'agente Amazon ECS supporta solo i dati di autenticazione `dockercfg` nel formato seguente senza l'oggetto `auths`. Se è installata l'utility 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
{"username":"another_name","password":"another_password","email":"email@example-02.com"}}
```

Procedura

Utilizza la procedura seguente per attivare i registri privati per le istanze di container.

Come abilitare i registri privati nell'AMI ottimizzata per Amazon ECS

1. Accedi alla tua istanza di container con SSH.
2. Apri il file `/etc/ecs/ecs.config` e aggiungi i valori `ECS_ENGINE_AUTH_TYPE` ed `ECS_ENGINE_AUTH_DATA` per il registro e l'account:

```
sudo vi /etc/ecs/ecs.config
```

In questo esempio si esegue l'autenticazione di un account utente Docker Hub:

```
ECS_ENGINE_AUTH_TYPE=docker
ECS_ENGINE_AUTH_DATA={"https://index.docker.io/v1/":
{"username":"my_name","password":"my_password","email":"email@example.com"}}
```

3. Controlla se il tuo agente utilizza la variabile di ambiente `ECS_DATADIR` per salvarne lo stato:

```
docker inspect ecs-agent | grep ECS_DATADIR
```

Output:

```
"ECS_DATADIR=/data",
```

> **Important**
>
> Se il comando precedente non restituisce la variabile di ambiente `ECS_DATADIR`, è necessario arrestare qualsiasi attività in esecuzione in questa istanza di container prima di arrestare l'agente. Gli agenti più nuovi con la variabile di ambiente `ECS_DATADIR` salvano il proprio stato e puoi arrestarli e avviarli mentre le attività vengono eseguite senza problemi. Per ulteriori informazioni, consulta Aggiornamento dell'agente del container Amazon ECS.

4. Arresta il servizio `ecs`:

```
sudo stop ecs
```

Output:

```
ecs stop/waiting
```

5. Riavvia il servizio `ecs`.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS:

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS:

```
sudo stop ecs && sudo start ecs
```

6. (Facoltativo) Puoi verificare se l'agente è in esecuzione e visualizzare alcune informazioni sulla nuova istanza di container interrogando l'operazione API di introspezione dell'agente. Per ulteriori informazioni, consulta the section called “Introspezione dei contenitori”.

```
curl http://localhost:51678/v1/metadata
```

Pulizia automatica di attività e immagini in Amazon ECS

Ogni volta che si colloca un processo in un'istanza di container, l'agente del container di Amazon ECS verifica se le immagini a cui si fa riferimento nell'attività sono le più recenti del tag specificato nel repository. In caso contrario, il comportamento di default consente all'agente di estrarre le immagini dai rispettivi repository. Se aggiorni di frequente le immagini nelle tue attività e nei servizi, il tuo storage dell'istanza di container si può riempire rapidamente con le immagini Docker che non stai più utilizzando e probabilmente non riutilizzerai mai più. Ad esempio, puoi usare una pipeline per l'integrazione e l'implementazione continue (CI/CD).

> **Note**
>
> Il comportamento pull dell'immagine dell'agente Amazon ECS può essere personalizzato utilizzando il parametro `ECS_IMAGE_PULL_BEHAVIOR`. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Analogamente, i container che appartengono ad attività arrestate possono anche consumare lo storage dell'istanza di container con informazioni di log, volumi di dati e altri elementi. Questi elementi sono utili per il debug dei container che si sono arrestati inaspettatamente, ma la maggior parte di questo storage può essere liberata in sicurezza dopo un periodo di tempo.

Di default, l'agente del container di Amazon ECS elimina automaticamente le attività interrotte e le immagini Docker che non vengono utilizzate da nessun processo nelle tue istanze di container.

> **Note**
>
> La funzione di pulizia automatizzata delle immagini richiede almeno la versione 1.13.0 dell'agente del container di Amazon ECS. Per aggiornare il tuo agente all'ultima versione, consulta Aggiornamento dell'agente del container Amazon ECS.

Le seguenti variabili di configurazione dell'agente sono disponibili per regolare la tua esperienza di attività automatica e pulizia delle immagini. Per ulteriori informazioni su come impostare queste variabili sulle tue istanze di container, consulta Configurazione dell'agente del container Amazon ECS.

`ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION`

Questa variabile specifica il tempo di attesa prima di rimuovere qualsiasi container appartenente ad attività arrestate. Il processo di pulizia delle immagini non può eliminare un'immagine finché c'è un container che fa riferimento alla stessa. Quando nessun container (interrotto o in esecuzione) fa riferimento alle immagini, l'immagine diventa una possibile candidata per la pulizia. Per impostazione predefinita, questo parametro è impostato su 3 ore, ma puoi ridurre questo periodo fino a 1 secondo, se necessario per la tua applicazione. Il parametro viene ignorato se imposti il valore a meno di 1 secondo.

`ECS_DISABLE_IMAGE_CLEANUP`

Se imposti questa variabile su `true`, la pulizia automatizzata delle immagini viene disattivata sull'istanza di container e nessuna immagine viene rimossa automaticamente.

`ECS_IMAGE_CLEANUP_INTERVAL`

Questa variabile specifica la frequenza con cui il processo di pulizia automatizzata delle immagini deve verificare la presenza di immagini da eliminare. L'impostazione predefinita è ogni 30 minuti, ma puoi ridurre questo periodo a solo 10 minuti per rimuovere le immagini con maggiore frequenza.

`ECS_IMAGE_MINIMUM_CLEANUP_AGE`

Questa variabile specifica la quantità di tempo minima tra l'estrazione di un'immagine e il momento in cui diventa una candidata per la rimozione. Viene utilizzata per evitare la pulizia di immagini appena estratte. Il valore predefinito è 1 ora.

`ECS_NUM_IMAGES_DELETE_PER_CYCLE`

Questa variabile specifica il numero di immagini rimovibili durante un singolo ciclo di pulizia. Il valore predefinito è 5 e il valore minimo è 1.

Quando l'agente del container di Amazon ECS è in esecuzione e la pulizia automatizzata delle immagini non è disattivata, l'agente verifica le immagini Docker alle quali nessun container in esecuzione o arrestato fa riferimento a una frequenza determinata dalla variabile `ECS_IMAGE_CLEANUP_INTERVAL`. Se vengono rilevate immagini inutilizzate meno recenti del tempo di pulizia minimo specificato dalla variabile `ECS_IMAGE_MINIMUM_CLEANUP_AGE`, l'agente rimuove fino al numero massimo di immagini specificato con la variabile `ECS_NUM_IMAGES_DELETE_PER_CYCLE`. Le immagini a cui si fa riferimento meno di recente vengono eliminate per prime. Dopo aver rimosso le immagini, l'agente attende fino all'intervallo successivo e ripete il processo.

Pianifica i tuoi contenitori su Amazon ECS

Amazon Elastic Container Service (Amazon ECS) è un sistema di concorrenza ottimistico a stati condivisi che offre funzionalità di pianificazione flessibili per i carichi di lavoro containerizzati. I pianificatori di Amazon ECS utilizzano le stesse informazioni sullo stato del cluster fornite dall'API Amazon ECS per poter prendere decisioni di posizionamento adeguate.

Amazon ECS fornisce un pianificatore di servizi per processi e applicazioni di lunga durata. Offre inoltre la possibilità di eseguire attività autonome o attività pianificate per lavori in batch o attività a esecuzione singola. Puoi specificare le strategie di posizionamento dei processi e i vincoli per l'esecuzione dei processi più adatte alle tue esigenze. Ad esempio, puoi specificare se i processi vengono eseguiti in più zone di disponibilità o all'interno di una singola zona di disponibilità. Puoi inoltre integrare i processi con pianificatori di terze parti o personalizzati.

Opzione	Quando usare	Ulteriori informazioni
Servizio	Il service scheduler è adatto per servizi e applicazioni stateless a esecuzione prolungata. Il pianificatore di servizi, facoltativamente, assicura anche che i processi siano registrati nel load balancer Elastic Load Balancing. Puoi aggiornare i servizi gestiti dal pianificatore di servizi. Ciò potrebbe includere l'implementazione di una nuova definizione di attività o la modifica del numero di processi desiderati in esecuzione. Di default, il pianificatore di servizi distribuisce i processi su più zone di disponibilità. Puoi utilizzare vincoli e strategie di	Servizi Amazon ECS

Opzione	Quando usare	Ulteriori informazioni
	posizionamento dei processi per personalizzare le decisioni riguardo al posizionamento dei processi.	
Processo autonomo	Un'attività autonoma è adatta per processi come i lavori in batch che eseguono un lavoro e poi si interromp ono. 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 possibili tà che una singola istanza ottenga un numero sproporzi onato di processi.	Attività autonome di Amazon ECS

Opzione	Quando usare	Ulteriori informazioni
Processi pianificati	Un'attività pianificata è adatta quando si hanno attività da eseguire a intervalli prestabiliti nel cluster, è possibile utilizzare EventBridge Scheduler per creare una pianificazione. Puoi eseguire processi per un'operazione di backup o una scansione del log. La EventBridge pianificazione dello Scheduler che crei può eseguire una o più attività nel cluster in orari specifici. L'evento pianificato può essere impostato su un intervallo specifico (esecuzione ogni *N* minuti, ore o giorni). In caso contrario, per una pianificazione più complessa, puoi utilizzare un'espressione `cron`.	Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS

Opzioni di calcolo

Con Amazon ECS, puoi specificare l'infrastruttura su cui vengono eseguite le tue attività o i tuoi servizi. Puoi utilizzare una strategia di capacity provider o un tipo di lancio.

Per Fargate, i fornitori di capacità sono Fargate e Fargate Spot. Per EC2, il fornitore di capacità è il gruppo Auto Scaling con le istanze di container registrate.

La strategia dei fornitori di capacità distribuisce le attività tra i fornitori di capacità associati al cluster.

In una strategia di provider di capacità possono essere utilizzati solo i provider di capacità che sono già associati a un cluster e hanno uno stato `ACTIVE` o `UPDATING`. Puoi associare un provider di capacità a un cluster durante la creazione di un cluster.

In una strategia del provider di capacità, il valore di base opzionale indica il numero minimo di attività da eseguire su un provider di capacità specificato. Solo un provider di capacità in una strategia di provider di capacità può avere una base definita.

Il valore del peso indica la percentuale relativa del numero totale di attività avviate che utilizzano il provider di capacità specificato. Analizza l'esempio seguente. Ad esempio, supponiamo di avere una strategia che contiene due provider di capacità, ognuno con un peso pari a 1. Una volta raggiunta la percentuale di base, le attività si dividono equamente tra i due provider di capacità. Usando la stessa logica, assumiamo di specificare un peso pari a 1 per capacityProviderA e un peso pari a 4 per capacityProviderB. Quindi, per ogni attività che viene eseguita utilizzando capacityProviderA, quattro attività utilizzano capacityProviderB.

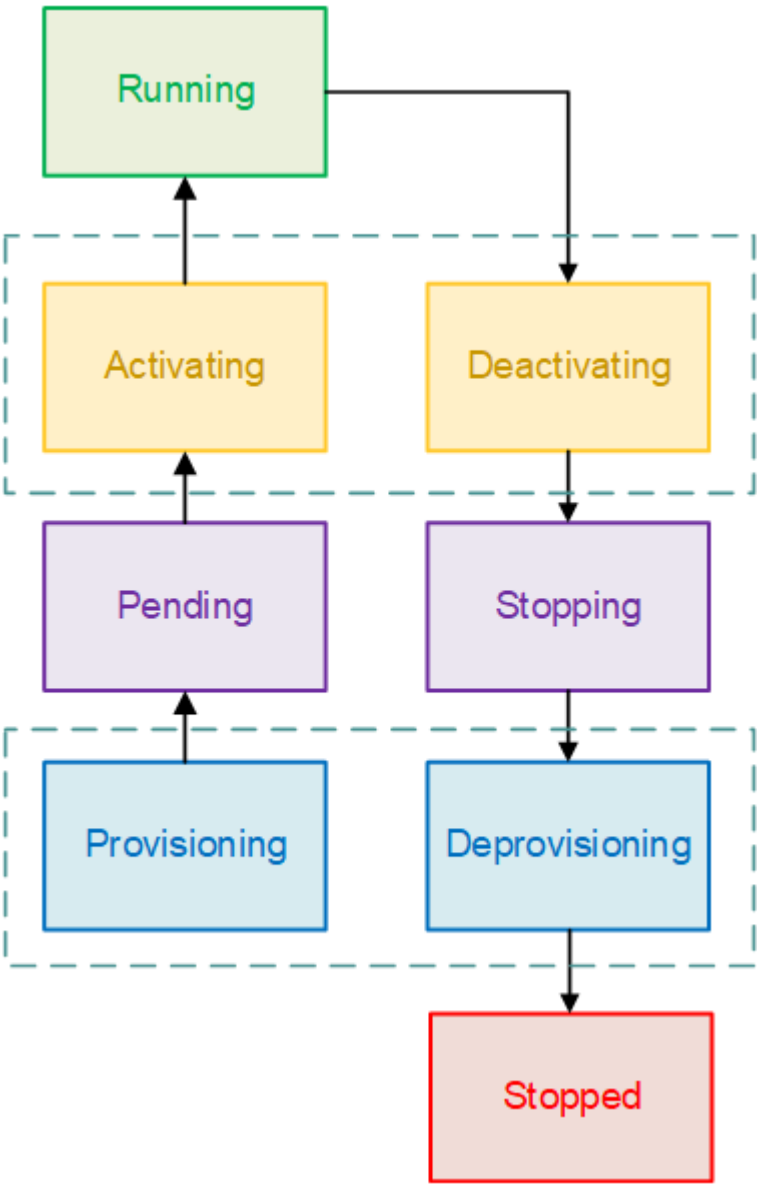
Il tipo di avvio avvia le tue attività direttamente su Fargate o sulle istanze Amazon EC2 che hai registrato manualmente nei tuoi cluster.

Ciclo di vita delle attività di Amazon ECS

Quando viene avviata un'attività, manualmente o come parte di un servizio, questa può attraversare diversi stati prima che termini da sola o venga interrotta manualmente. Alcune attività devono essere eseguite come processi in batch che passano in modo naturale da `PENDING` a `RUNNING` a `STOPPED`. Altre attività, che possono fare parte di un servizio, devono rimanere in esecuzione a tempo indeterminato o a essere ridimensionate in base alle esigenze.

Quando vengono richieste modifiche allo stato delle attività, ad esempio l'interruzione di un'attività o l'aggiornamento del conteggio desiderato di un servizio per aumentarne o ridurne le dimensioni, l'agente del container Amazon ECS traccia tali modifiche come ultimo stato noto dell'attività (`lastStatus`) e stato desiderato dell'attività (`desiredStatus`). È possibile scoprire sia l'ultimo stato noto che lo stato desiderato di un'attività nella console o descrivendo l'attività con l'API o l' AWS CLI.

Il diagramma di seguito mostra il flusso del ciclo di vita dell'attività.



Stati del ciclo di vita

Di seguito è descritto ciascuno stato del ciclo di vita dell'attività.

PROVISIONING

Amazon ECS deve eseguire operazioni aggiuntive prima che il processo venga avviato. Ad esempio, per le attività che utilizzano la modalità di rete `awsvpc`, deve essere predisposta l'interfaccia di rete elastica.

PENDING

Si tratta di uno stato di transizione in cui Amazon ECS è in attesa che l'agente del container effettui ulteriori operazioni. Un'attività rimane nello stato in sospeso fino a quando non sono disponibili risorse per l'attività.

ACTIVATING

Si tratta di uno stato di transizione in cui Amazon ECS deve eseguire operazioni aggiuntive dopo l'avvio dell'attività, ma prima che questa possa passare allo stato `RUNNING`. Ad esempio, per le attività con discovery dei servizi configurata, devono essere create le relative risorse. Per processi che fanno parte di un servizio che è configurato per l'utilizzo di più gruppi di destinazione di Elastic Load Balancing, la registrazione del gruppo di destinazione si verifica durante questo stato.

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 più 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 `SIGTERM` segnale per notificare la necessità di terminare e chiudere l'applicazione, quindi invierà un messaggio `SIGKILL` dopo aver atteso la `StopTimeout` durata impostata nella definizione dell'attività.

DEPROVISIONING

Amazon ECS deve eseguire operazioni aggiuntive dopo l'arresto del processo, ma prima che questo passi allo stato `STOPPED`. Ad esempio, per le attività che utilizzano la modalità di rete `awsvpc`, deve essere scollegata e rimossa l'interfaccia di rete elastica.

STOPPED

L'arresto dell'attività è riuscito.

Se l'operazione è stata interrotta a causa di un errore, consultaVisualizzazione degli errori delle attività interrotte da Amazon ECS.

ELIMINATO

Si tratta di uno stato di transizione quando un'operazione viene interrotta. Questo stato non viene visualizzato nella console, ma viene visualizzato in `describe-tasks`.

In che modo Amazon ECS colloca le attività sulle istanze di container

Puoi utilizzare il posizionamento delle attività per configurare Amazon ECS in modo da collocare le tue attività su istanze di container che soddisfano determinati criteri, ad esempio una zona di disponibilità o un tipo di istanza.

Di seguito sono riportati i componenti per il posizionamento delle attività:

- Strategia di posizionamento delle attività: l'algoritmo per selezionare le istanze del contenitore per il posizionamento delle attività o le attività da terminare. Ad esempio, Amazon ECS può selezionare istanze di container a caso oppure può selezionare istanze di container in modo tale che le attività siano distribuite uniformemente su un gruppo di istanze.
- Gruppo di attività: un gruppo di attività correlate, ad esempio attività di database.
- Vincolo di posizionamento delle attività: si tratta di regole che devono essere soddisfatte per inserire un'attività su un'istanza del contenitore. Se il vincolo non viene soddisfatto, l'attività non viene inserita e rimane nello stato. `PENDING` Ad esempio, è possibile utilizzare un vincolo per posizionare le attività solo su un particolare tipo di istanza.

Amazon ECS ha diversi algoritmi per i tipi di lancio.

Tipo di avvio EC2

Per le attività che utilizzano il tipo di avvio EC2, Amazon ECS deve determinare dove posizionare 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 posizione e termina i processi.

Le strategie di posizionamento delle attività predefinite dipendono dal fatto che le attività vengano eseguite manualmente (attività autonome) o all'interno di un servizio. Per le attività eseguite come parte di un servizio Amazon ECS, la strategia di posizionamento delle attività è `spread` e utilizza `attribute:ecs.availability-zone`. Non esiste un vincolo predefinito per il posizionamento delle attività nei servizi. Per ulteriori informazioni, consulta Pianifica i tuoi contenitori su Amazon ECS.

> **Note**
>
> Le strategie di posizionamento dei processi si basano sul miglior tentativo. Amazon ECS prova a posizionare i processi anche quando l'opzione di posizionamento ottimale non è disponibile. Tuttavia, i vincoli di posizionamento delle attività sono vincolanti, per cui potrebbero impedire il posizionamento delle attività.

Puoi utilizzare strategie e vincoli di posizionamento delle attività contemporaneamente. Ad esempio, puoi utilizzare una strategia di posizionamento e un vincolo di posizionamento delle attività per distribuire le attività all'interno di zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, ma solo per quanto riguarda le istanze G2.

Quando Amazon ECS posiziona i processi, utilizza la seguente procedura per selezionare le istanze di container:

1. Identifica le istanze del contenitore che soddisfano i requisiti di CPU, GPU, memoria e porta nella definizione dell'attività.
2. Identifica le istanze del contenitore che soddisfano i vincoli di posizionamento delle attività.
3. Identifica le istanze del contenitore che soddisfano le strategie di posizionamento delle attività.
4. Seleziona le istanze del contenitore per il posizionamento delle attività.

Tipo di avvio di Fargate

Per processi con tipo di avvio Fargate, le strategie e i vincoli di posizionamento dei processi non sono supportati. Fargate farà del suo meglio per distribuire le attività tra zone di disponibilità accessibili. Se il provider di capacità include sia Fargate che Fargate Spot, il comportamento di distribuzione sarà indipendente per ogni provider.

Usa strategie per definire il posizionamento delle attività di Amazon ECS

Per le attività che utilizzano il tipo di avvio EC2, Amazon ECS deve determinare dove posizionare 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 posizione e termina i processi.

Le strategie di posizionamento delle attività predefinite dipendono dal fatto che le attività vengano eseguite manualmente (attività autonome) o all'interno di un servizio. Per le attività eseguite come parte di un servizio Amazon ECS, la strategia di posizionamento delle attività è `spread` e utilizza `attribute:ecs.availability-zone`. Non esiste un vincolo predefinito per il posizionamento delle attività nei servizi. Per ulteriori informazioni, consulta Pianifica i tuoi contenitori su Amazon ECS.

> **Note**
>
> Le strategie di posizionamento dei processi si basano sul miglior tentativo. Amazon ECS prova a posizionare i processi anche quando l'opzione di posizionamento ottimale non è disponibile. Tuttavia, i vincoli di posizionamento delle attività sono vincolanti, per cui potrebbero impedire il posizionamento delle attività.

Puoi utilizzare strategie e vincoli di posizionamento delle attività contemporaneamente. Ad esempio, puoi utilizzare una strategia di posizionamento e un vincolo di posizionamento delle attività per distribuire le attività all'interno di zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, ma solo per quanto riguarda le istanze G2.

Quando Amazon ECS posiziona i processi, utilizza la seguente procedura per selezionare le istanze di container:

1. Identifica le istanze del contenitore che soddisfano i requisiti di CPU, GPU, memoria e porta nella definizione dell'attività.
2. Identifica le istanze del contenitore che soddisfano i vincoli di posizionamento delle attività.
3. Identifica le istanze del contenitore che soddisfano le strategie di posizionamento delle attività.
4. Seleziona le istanze del contenitore per il posizionamento delle attività.

Le strategie di posizionamento delle attività vengono specificate nella definizione del servizio o la definizione dell'attività utilizzando il `placementStrategy` parametro.

```
"placementStrategy": [
    {
        "field": "The field to apply the placement strategy against",
        "type": "The placement strategy to use"
    }
]
```

È possibile specificare le strategie quando si esegue un'attività (RunTask), si crea un nuovo servizio (CreateService) o si aggiorna un servizio esistente (UpdateService).

La tabella seguente descrive i tipi e i campi disponibili.

tipo	Valori di campo validi	
`binpack` Le attività vengono posiziona te 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 operazion 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 disponibi li rimanenti dopo la fine del	• cpu • memory	

tipo	Valori di campo validi	
processo avrà il processo terminato.		
`random` Le attività vengono posiziona te in modo casuale.	Non utilizzato	
`spread` Le attività vengono posiziona te in modo uniforme in base al valore specificato. Le attività di servizio vengono distribuite in base alle attività di tale servizio. Attività standalone sono distribui te 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.	• `instanceI d` (oppure`host`, che ha lo stesso effetto) • qualsiasi piattaforma o attributo personalizzato applicato a un'istanza del contenitore, ad esempio `attribute:ecs.avai lability-zone`	

Le strategie di posizionamento delle attività possono essere aggiornate anche per i servizi esistenti. Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

Puoi creare una strategia di posizionamento delle attività che utilizzi più strategie tramite la creazione di una serie di strategie nell'ordine in cui desideri che vengano eseguite. Ad esempio, se desideri distribuire le attività tra le zone di disponibilità e raggrupparle in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità, specifica la strategia della zona di disponibilità seguita dalla strategia della memoria. Per le strategie di esempio, consulta Esempi di strategie di collocamento delle attività di Amazon ECS.

Esempi di strategie di collocamento delle attività di Amazon ECS

Puoi specificare strategie di posizionamento delle attività con le seguenti azioni: CreateService, UpdateService, e RunTask.

Esempi

- Distribuire le attività in modo uniforme all'interno delle zone di disponibilità
- Distribuire le attività in modo uniforme su tutte le istanze
- Raggruppare le attività in bin packing in base alla memoria
- Posizionare le attività casualmente
- Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e distribuire le attività all'interno delle istanze in ciascuna zona di disponibilità
- Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e raggruppare le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità
- Distribuire le attività in modo uniforme tra le istanze e raggruppare le attività in bin packing in base alla memoria

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zona di disponibilità.

```
"placementStrategy": [
    {
        "field": "attribute:ecs.availability-zone",
        "type": "spread"
    }
```

```
]
```

Distribuire le attività in modo uniforme su tutte le istanze

La strategia seguente distribuisce le attività in modo uniforme all'interno di tutte le istanze.

```
"placementStrategy": [
    {
        "field": "instanceId",
        "type": "spread"
    }
]
```

Raggruppare le attività in bin packing in base alla memoria

La strategia seguente raggruppa le attività in bin packing in base alla memoria.

```
"placementStrategy": [
    {
        "field": "memory",
        "type": "binpack"
    }
]
```

Posizionare le attività casualmente

La strategia seguente posiziona le attività in modo casuale.

```
"placementStrategy": [
    {
        "type": "random"
    }
]
```

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e distribuire le attività all'interno delle istanze in ciascuna zona di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zone di disponibilità, quindi distribuisce le attività all'interno delle istanze in ciascuna zona di disponibilità.

```
"placementStrategy": [
    {
```

```
        "field": "attribute:ecs.availability-zone",
        "type": "spread"
    },
    {
        "field": "instanceId",
        "type": "spread"
    }
]
```

Distribuire le attività in modo uniforme all'interno delle zone di disponibilità e raggruppare le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità

La strategia seguente distribuisce le attività in modo uniforme all'interno delle zone di disponibilità, quindi raggruppa le attività in bin packing in base alla memoria all'interno di ciascuna zona di disponibilità.

```
"placementStrategy": [
    {
        "field": "attribute:ecs.availability-zone",
        "type": "spread"
    },
    {
        "field": "memory",
        "type": "binpack"
    }
]
```

Distribuire le attività in modo uniforme tra le istanze e raggruppare le attività in bin packing in base alla memoria

La seguente strategia distribuisce le attività in modo uniforme tra le istanze, quindi raggruppa le attività in bin packing in base alla memoria all'interno di ciascuna istanza.

```
"placementStrategy": [
    {
        "field": "instanceId",
        "type": "spread"
    },
    {
        "field": "memory",
        "type": "binpack"
    }
```

```
]
```

Attività Amazon ECS relative al gruppo

Puoi identificare una serie di attività correlate e inserirle in un gruppo di attività. Tutti i processi con lo stesso nome del gruppo di processi sono considerati come un set quando si utilizza la strategia di posizionamento dei processi `spread`. Ad esempio, supponiamo che stia eseguendo diverse applicazioni in un unico cluster, come database e server Web. Per assicurarti che i database siano bilanciati all'interno delle zone di disponibilità, aggiungili a un gruppo di processi denominato `databases`, quindi utilizza la strategia di posizionamento dei processi `spread`. Per ulteriori informazioni, consulta Usa strategie per definire il posizionamento delle attività di Amazon ECS.

I gruppi di processi possono essere utilizzati anche come vincolo di posizionamento dei processi. Quando specifichi un gruppo di attività nel vincolo `memberOf`, le attività vengono inviate solo alle istanze di container presenti nel gruppo di attività specificato. Per vedere un esempio, consulta Esempio di vincoli di posizionamento delle attività di Amazon ECS.

Di default, i processi autonomi utilizzano il nome della famiglia della definizione di attività (ad esempio, `family:my-task-definition`) come nome del gruppo di processi se non è specificato un nome di gruppo di processi personalizzato. I processi avviati come parte di un servizio utilizzano il nome del servizio come nome del gruppo di processi e non possono essere modificati.

Si applicano i seguenti requisiti per il gruppo di attività.

- Il nome di un gruppo di processi deve essere al massimo di 255 caratteri.
- Ciascuna attività può trovarsi in un solo gruppo.
- Dopo l'avvio di un processo, non puoi modificarne il gruppo di processi.

Definisci quali istanze di container Amazon ECS utilizza per le attività

Un vincolo di posizionamento delle attività è una regola relativa a un'istanza di contenitore che Amazon ECS utilizza per determinare se l'attività può essere eseguita sull'istanza. Almeno un'istanza di container deve corrispondere al vincolo. Se non sono presenti istanze corrispondenti al vincolo, l'attività rimane in uno stato `PENDING`. Quando crei un nuovo servizio o ne aggiorni uno esistente, puoi specificare i vincoli di posizionamento delle attività per le attività del servizio.

È possibile specificare i vincoli di posizionamento delle attività nella definizione del servizio, nella definizione dell'attività o nell'attività utilizzando il parametro. `placementConstraint`

```
"placementConstraint": [
    {
        "expression": "The expression that defines the task placement constraints",
        "type": "The placement constraint type to use"
    }
]
```

La tabella seguente descrive come utilizzare i parametri.

Constraint type (Tipo di vincolo)	Può essere specificato quando	
`distinctInstance` Posizionamento di ciascuna attività in una istanza di container differente. **Important** Consigliamo ai clienti che cercano un forte isolamento per le proprie attività di utilizzare Fargate. Fargate esegue ogni attività in un ambiente di virtualiz zazione hardware. Ciò garantisce che questi carichi di lavoro containerizzati non condividano interfacc e di rete, storage temporaneo Fargate, CPU o memoria con altre attività. Per ulteriori informazioni,	• Esecuzione di un'attività RunTask • Creando un nuovo servizio CreateService,	

Constraint type (Tipo di vincolo)	Può essere specificato quando	
vedere Panoramica a sulla sicurezza di. AWS Fargate		
`memberOf` 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 RegisterTaskdefinizione di attività • Creazione di una nuova revisione della definizione RegisterTask di un'attività • Aggiornamento di un servizio UpdateService	

Quando utilizzi il tipo di `memberOf` vincolo, puoi creare un'espressione utilizzando il linguaggio di interrogazione del cluster che definisce le istanze del contenitore in cui Amazon ECS può collocare le attività. L'espressione è un modo per raggruppare le istanze del contenitore in base agli attributi. L'espressione rientra nel `expression` parametro di. `placementConstraint`

Attributi dell'istanza del contenitore Amazon ECS

Puoi aggiungere metadati personalizzati alle istanze di container, note come attributi. Ogni attributo ha un nome e un valore di stringa facoltativo. Puoi utilizzare gli attributi integrati offerti da Amazon ECS oppure definire attributi personalizzati.

Le sezioni seguenti contengono attributi incorporati, facoltativi e personalizzati di esempio.

Attributi integrati

Amazon ECS applica automaticamente i seguenti attributi alle istanze di container.

`ecs.ami-id`

L'ID dell'AMI utilizzato per avviare l'istanza. Un valore di esempio per questo attributo è `ami-1234abcd`.

`ecs.availability-zone`

La zona di disponibilità dell'istanza. Un valore di esempio per questo attributo è `us-east-1a`.

`ecs.instance-type`

Il tipo di istanza dell'istanza. Un valore di esempio per questo attributo è `g2.2xlarge`.

`ecs.os-type`

Il sistema operativo dell'istanza. I valori possibili per questo attributo sono `linux` e `windows`.

`ecs.os-family`

La versione del sistema operativo dell'istanza.

Per le istanze Linux, il valore valido è `LINUX`. Per le istanze Windows, ECS imposta il valore nel formato `WINDOWS_SERVER_<OS_Release>_<FULL or CORE>`. I valori validi sono `WINDOWS_SERVER_2022_FULL`, `WINDOWS_SERVER_2022_CORE`, `WINDOWS_SERVER_20H2_CORE`, `WINDOWS_SERVER_2019_FULL`, `WINDOWS_SERVER_2019_CORE` e `WINDOWS_SERVER_2016_FULL`.

Questo è importante per i contenitori Windows e Windows containers on AWS Fargate perché la versione del sistema operativo di ogni contenitore Windows deve corrispondere a quella dell'host. Se la versione Windows dell'immagine del container è diversa da quella dell'host, il container non si avvia. Per ulteriori informazioni, consulta Compatibilità con la versione del container Windows sul sito Web della documentazione Microsoft.

Se il cluster esegue più versioni di Windows, puoi assicurarti che un'attività venga posizionata su un'istanza EC2 in esecuzione sulla stessa versione utilizzando il vincolo di posizionamento: `memberOf(attribute:ecs.os-family == WINDOWS_SERVER_<OS_Release>_<FULL or CORE>)`. Per ulteriori informazioni, consulta the section called “Recupero di metadati AMI Windows ottimizzati per Amazon ECS”.

`ecs.cpu-architecture`

Architettura della CPU per l'istanza. I valori di esempio per questo attributo sono `x86_64` e `arm64`.

`ecs.vpc-id`

VPC in cui è stata avviata l'istanza. Un valore di esempio per questo attributo è `vpc-1234abcd`.

`ecs.subnet-id`

La sottorete utilizzata dall'istanza. Un valore di esempio per questo attributo è `subnet-1234abcd`.

Attributi facoltativi

Amazon ECS può aggiungere i seguenti attributi alle istanze di container.

`ecs.awsvpc-trunk-id`

Se questo attributo esiste, l'istanza dispone di un'interfaccia di rete trunk. Per ulteriori informazioni, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.

`ecs.outpost-arn`

Se questo attributo esiste, contiene l'Amazon Resource Name (ARN) dell'Outpost. Per ulteriori informazioni, consulta the section called “Amazon Elastic Container Service su AWS Outposts”.

`ecs.capability.external`

Se questo attributo esiste, l'istanza viene identificata come un'istanza esterna. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio esterno.

Attributi personalizzati

Puoi applicare attributi personalizzati alle istanze di container. Ad esempio, puoi definire un attributo con il nome "stack" e un valore di "prod".

Quando si specificano attributi personalizzati, è necessario considerare quanto segue.

- Il `name` può contenere un massimo di 128 caratteri e può contenere lettere (maiuscole e minuscole), numeri, trattini, caratteri di sottolineatura, barre, barre inverse o punti.
- Il `value` può contenere un massimo di 128 caratteri e può contenere lettere (maiuscole e minuscole), numeri, trattini, caratteri di sottolineatura, punti, chiocciola (@), barre, barre inverse, due punti o spazi. Il valore non può contenere spazi iniziali o finali.

Crea espressioni per definire istanze di container per le attività di Amazon ECS

Le query di cluster sono espressioni che consentono di raggruppare gli oggetti. Ad esempio, puoi raggruppare le istanze di container per attributi, ad esempio la zona di disponibilità, il tipo di istanza o metadati personalizzati. Per ulteriori informazioni, consulta Attributi dell'istanza del contenitore Amazon ECS.

Dopo aver definito un gruppo di istanze di container, potrai personalizzare Amazon ECS per posizionare i processi su istanze di container basate sul gruppo. Per ulteriori informazioni, consulta Esecuzione di un'applicazione come attività Amazon ECS e Creazione di un servizio Amazon ECS utilizzando la console. Puoi inoltre applicare un filtro di gruppo per elencare le istanze di container.

Sintassi delle espressioni

Le espressioni presentano la sintassi seguente:

```
subject operator [argument]
```

Subject

L'attributo o il campo da valutare.

`agentConnected`

Seleziona le istanze di container in base allo stato di connessione dell'agente del container di Amazon ECS. È possibile usare questo filtro per cercare le istanze con agenti di container che sono disconnessi.

Operatori validi: equals (==), not_equals (!=), in, not_in (!in), matches (=~), not_matches (!~)

`agentVersion`

Seleziona le istanze di container in base alla versione dell'agente del container di Amazon ECS. È possibile usare questo filtro per trovare le istanze che eseguono le versioni obsolete dell'agente del container Amazon ECS.

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

`attribute:`*`attribute-name`*

Seleziona le istanze di container per attributo. Per ulteriori informazioni, consulta Attributi dell'istanza del contenitore Amazon ECS.

`ec2InstanceId`

Seleziona le istanze di container in base al loro ID istanza di Amazon EC2.

Operatori validi: equals (==), not_equals (!=), in, not_in (!in), matches (=~), not_matches (!~)

`registeredAt`

Seleziona le istanze di container in base alla loro data di registrazione dell'istanza di container. È possibile usare questo filtro per individuare le istanze appena registrate o le istanze molto vecchie.

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

Formati di data validi: 2018-06-18T22:28:28+00:00, 2018-06-18T22:28:28Z, 2018-06-18T22:28:28, 2018-06-18

`runningTasksCount`

Seleziona le istanze di container in base al numero di attività in esecuzione. È possibile usare questo filtro per trovare le istanze che sono vuote o quasi vuote (poche attività in esecuzione su di esse).

Operatori validi: equals (==), not_equals (!=), greater_than (>), greater_than_equal (>=), less_than (<), less_than_equal (<=)

`task:group`

Seleziona le istanze di container per gruppo di attività. Per ulteriori informazioni, consulta Attività Amazon ECS relative al gruppo .

Operatore

L'operatore di confronto. Sono supportati i seguenti operatori.

Operatore	Descrizione
==, equals	Uguaglianza stringhe
!=, not_equals	Disuguaglianza stringhe
>, greater_than	Maggiore di

Operatore	Descrizione
>=, 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
!~, 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, consulta 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 dell'istanza Amazon EC2

L'espressione seguente seleziona le istanze con i seguenti ID istanza di Amazon EC2.

```
ec2InstanceId in ['i-abcd1234', 'i-wxyx7890']
```

Esempio di vincoli di posizionamento delle attività di Amazon ECS

Di seguito sono elencati esempi di vincoli del posizionamento delle attività.

Questo esempio utilizza il `memberOf` vincolo per posizionare le attività sulle istanze t2. Può essere specificato con le seguenti azioni: CreateService, UpdateService, RegisterTask Definition 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'istanza EC2 prima delle attività di replica del servizio.

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, UpdateService, RegisterTaskDefinizione e RunTask.

```
"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

```
"placementConstraints": [
    {
```

```
        "type": "distinctInstance"
    }
]
```

Attività autonome di Amazon ECS

Puoi eseguire l'applicazione come attività quando hai un'applicazione che esegue alcune operazioni e poi interrompe, ad esempio un processo in batch. Se desideri eseguire un'attività una sola volta, puoi utilizzare la console AWS CLI, le API o gli SDK.

Se devi eseguire l'applicazione su una pianificazione basata sulla frequenza, basata su cronn o una tantum, puoi creare una pianificazione utilizzando Scheduler. EventBridge

Workflow delle attività

Quando avvii attività di Amazon ECS (attività autonome o tramite servizi Amazon ECS), un'attività viene creata e inizialmente spostata nello stato. `PROVISIONING` Quando un'attività è nello `PROVISIONING` stato, non esistono né l'attività né i contenitori perché Amazon ECS deve trovare la capacità di calcolo per collocare l'attività.

Amazon ECS seleziona la capacità di elaborazione appropriata per l'attività in base al tipo di avvio o alla configurazione del provider di capacità. Puoi utilizzare i provider di capacità e le strategie dei provider di capacità con i tipi di lancio Fargate e Amazon EC2. Con Fargate, non è necessario pensare al provisioning, alla configurazione e alla scalabilità della capacità del cluster. Fargate si occupa di tutta la gestione dell'infrastruttura per le tue attività. Per il tipo di lancio EC2, puoi gestire la capacità del cluster registrando le istanze Amazon EC2 nel cluster oppure puoi utilizzare l'auto scaling del cluster per semplificare la gestione della capacità di calcolo. La scalabilità automatica del cluster si occupa della scalabilità dinamica della capacità del cluster, in modo che tu possa concentrarti sulle attività in esecuzione. Amazon ECS determina dove collocare l'attività in base ai requisiti specificati nella definizione dell'attività, come CPU e memoria, nonché ai vincoli e alle strategie di posizionamento. Per ulteriori informazioni, consultare In che modo Amazon ECS colloca le attività sulle istanze di container.

Se utilizzi un provider di capacità con scalabilità gestita abilitata, le attività che non possono essere avviate a causa della mancanza di capacità di elaborazione vengono spostate `PROVISIONING` nello stato anziché fallire immediatamente. Dopo aver trovato la capacità di collocare l'attività, Amazon ECS fornisce gli allegati necessari (ad esempio, Elastic Network Interfaces (ENI) per le attività in modalità). `awsvpc` Utilizza l'agente container Amazon ECS per estrarre le immagini dei contenitori

e quindi avviare i contenitori. Dopo il completamento del provisioning e il lancio dei contenitori pertinenti, Amazon ECS sposta l'attività in stato. `RUNNING` Per informazioni sugli stati delle attività, consulta. Ciclo di vita delle attività di Amazon ECS

Ottimizza il tempo di avvio delle attività di Amazon ECS

Per velocizzare l'avvio delle attività, prendi in considerazione i seguenti consigli.

- Memorizza nella cache le immagini dei contenitori e le istanze binpack

  Se utilizzi il tipo di avvio EC2, puoi configurare il comportamento pull dell'agente container Amazon ECS su:. `ECS_IMAGE_PULL_BEHAVIOR` `prefer-cached` L'immagine viene recuperata in remoto se non è presente alcuna immagine memorizzata nella cache. In caso contrario, viene utilizzata l'immagine memorizzata nella cache dell'istanza. La pulizia automatica delle immagini è disattivata per il contenitore per garantire che l'immagine memorizzata nella cache non venga rimossa. Ciò riduce il tempo di recupero delle immagini per gli avvii successivi. L'effetto della memorizzazione nella cache è ancora maggiore quando nelle istanze del contenitore è presente un'elevata densità di attività, che è possibile configurare utilizzando la strategia di posizionamento. `binpack` La memorizzazione nella cache delle immagini dei container è particolarmente utile per i carichi di lavoro basati su Windows, che di solito hanno immagini container di grandi dimensioni (decine di GB). Quando utilizzi la strategia di `binpack` posizionamento, puoi anche prendere in considerazione l'utilizzo del trunking Elastic Network Interface (ENI) per collocare più attività con la modalità di rete su ogni istanza del `awsvpc` contenitore. Il trunking ENI aumenta il numero di attività che è possibile 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 le API di Amazon EC2, che aggiungono un sovraccarico di diversi secondi all'avvio delle attività. Al contrario, un vantaggio chiave dell'utilizzo della modalità di `awsvpc` rete è che ogni attività ha un gruppo di sicurezza che consente o nega il traffico. Ciò significa che hai una maggiore flessibilità per controllare le comunicazioni tra attività e servizi a un livello più granulare. Se la velocità di implementazione è la tua priorità, puoi prendere in considerazione l'utilizzo della `bridge` modalità per velocizzare l'avvio delle attività. Per ulteriori informazioni, consulta the section called “AWSVPC modalità di rete”.

- Tieni traccia del ciclo di vita dell'avvio delle attività per trovare opportunità di ottimizzazione

  Spesso è difficile conoscere il tempo necessario per avviare l'applicazione. L'avvio dell'immagine del contenitore, l'esecuzione di script di avvio e altre configurazioni durante l'avvio dell'applicazione possono richiedere una quantità di tempo sorprendente. Puoi utilizzare l'endpoint di metadati Task per pubblicare metriche per tenere traccia del tempo di avvio dell'applicazione da quando l'applicazione è pronta `ContainerStartTime` per servire il traffico. Grazie a questi dati, è possibile comprendere in che modo l'applicazione contribuisce al tempo totale di avvio e individuare aree in cui è possibile ridurre il sovraccarico non necessario specifico dell'applicazione e ottimizzare le immagini del contenitore. Per ulteriori informazioni, consulta Ottimizza la capacità e la disponibilità di Amazon ECS.
- Scegli un tipo di istanza ottimale (per il tipo di avvio EC2)

  La scelta del tipo di istanza corretto si basa sulla prenotazione di risorse (ad esempio, CPU, memoria) configurata per l'attività. Pertanto, quando si ridimensiona l'istanza, è possibile calcolare quante attività possono essere posizionate su una singola istanza. Un semplice esempio di attività ben posizionata è l'hosting di 4 attività che richiedono 0,5 vCPU e 2 GB di riserve di memoria in un'istanza m5.large (che supporta 2 vCPU e 8 GB di memoria). Le prenotazioni di questa definizione di attività sfruttano appieno le risorse dell'istanza.

Esecuzione di un'applicazione come attività Amazon ECS

Puoi creare un'attività per un processo singolo utilizzando. AWS Management Console

Per creare un'attività autonoma ()AWS Management Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. La console Amazon ECS ti consente di creare un'attività autonoma dalla pagina dei dettagli del cluster o dall'elenco di revisione delle definizioni delle attività. Utilizza i seguenti passaggi per creare un'attività autonoma in base alla pagina delle risorse scelta.

Per avviare un servizio da	Fasi
una pagina di dettaglio del cluster...	a. Nella pagina Cluster, seleziona il cluster in cui creare il servizio.

Per avviare un servizio da	Fasi
	b. Dalla scheda Processi, scegli Esegui nuovo processo.
una pagina di revisione della definizione delle attività...	a. Nella pagina Definizioni delle attività, scegli la famiglia di definizioni delle attività per visualizzare le revisioni per quella famiglia. b. Seleziona la revisione che desideri utilizzare. c. Dal menu Distribuisci, scegli Esegui attività.

3. (Facoltativo) Nella sezione Configurazione di elaborazione (avanzata) puoi scegliere come distribuire le tue attività. Puoi utilizzare una strategia di Capacity provider o un tipo Launch. Per utilizzare una strategia di provider di capacità, è necessario configurare i provider di capacità a livello di cluster. Se non hai configurato il cluster per utilizzare un provider di capacità, utilizza invece un tipo di avvio.

Metodo di distribuzione	Fasi
Strategia del provider di capacità	a. Nella sezione Compute options (Opzioni di 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

Metodo di distribuzione	Fasi
	cluster default (Usa impostazione predefinita del cluster. • Se il cluster non dispone di una strategia del provider di capacità predefinito o per utilizzare una strategia personalizzata, scegli Use custom (Usa personalizzato), Add capacity provider strategy (Aggiungi strategia del provider di capacità) e definisci la strategia del provider di capacità personalizzata specificando Base (Base), Capacity provider (Provider di capacità) e Weight (Peso). **Note** Per utilizzare un provider di capacità in una strategia , il provider di capacità deve essere associato al cluster.

Metodo di distribuzione	Fasi
Tipo di avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Launch type (Tipo di avvio), seleziona un tipo di avvio. c. (Facoltativo) Quando viene specificato il tipo di avvio Fargate, per Versione piattaforma specifica la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma `LATEST`.

4. Per Tipo di applicazione, scegli Processo.
5. Per Definizione dell'attività, scegli la famiglia di definizioni dell'attività e la revisione.

> **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.

6. Per Desired tasks (Attività desiderate), specifica il numero di attività da avviare.
7. Se la tua definizione di attività utilizza la modalità di rete `awsvpc`, espandi Networking (Rete). Per specificare una configurazione personalizzata, completa la procedura riportata di seguito.
   a. Per VPC seleziona il VPC da utilizzare.
   b. Per Subnets (Sottoreti), seleziona una o più sottoreti nel VPC che lo scheduler di attività deve prendere in considerazione quando posiziona le attività.

> **Important**
>
> Per la modalità di rete `awsvpc` sono supportate solo sottoreti private. Le attività non ricevono indirizzi IP pubblici. Pertanto, è necessario un gateway NAT per l'accesso a Internet in uscita e il traffico Internet in entrata viene instradato attraverso un load balancer.

c. Per Gruppi di sicurezza puoi scegliere un gruppo di sicurezza esistente o crearne uno nuovo. Per utilizzare un gruppo di sicurezza esistente, scegli il gruppo di sicurezza e passa alla fase successiva. Per creare un nuovo gruppo di sicurezza, scegliere Create a new security group (Crea un nuovo gruppo di sicurezza). È necessario specificare un nome e una descrizione del gruppo di sicurezza e aggiungere una o più regole in entrata per il gruppo di sicurezza.

d. Per IP pubblico scegli se assegnare automaticamente un indirizzo IP pubblico all'interfaccia di rete elastica (ENI) del processo stesso.

   AWS Fargate alle attività può essere assegnato un indirizzo IP pubblico quando vengono eseguite in una sottorete pubblica in modo che abbiano un percorso verso Internet. Per ulteriori informazioni, consulta Reti di processi Fargate nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate.

8. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

   Il nome del volume e il tipo di volume vengono configurati durante la creazione di una revisione della definizione di attività e non possono essere modificati quando si esegue un'attività autonoma. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione dell'attività ed eseguire un'attività utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per il tipo di volume EBS, scegli il tipo di volume EBS che desideri allegare all'attività.	

Per configurare questo tipo di volume	Esegui questa operazione	
	b. Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettete il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per `io1` i tipi di volume e. `io2` `gp3` d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). MiBps Questo valore è configurabile solo per il tipo di volume. `gp3` e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno snapshot se desideri creare un volume da uno	

Per configurare questo tipo di volume	Esegui questa operazione	
	snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Per la politica di terminazione, deseleziona la casella di controllo se desideri che il volume configurato per l'allegato all'attività venga preservato dopo il termine dell'attività. Per impostazione predefinita, i volumi EBS allegati alle attività vengono eliminati quando l'attività viene terminata. g. Per Tipo di file system, scegli il tipo di file system che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico. L'impostazione predefinita per Linux è`XFS`. Per i volumi creati da un'istantanea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea.	

Per configurare questo tipo di volume	Esegui questa operazione	
	Se il tipo di file system non corrisponde, l'operazione non verrà avviata. h. Per il ruolo Infrastructure, scegli un ruolo IAM con le autorizzazioni necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy `AmazonECSInfrastructureRolePolicyForVolumes` gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta. Ruolo IAM dell'infrastruttura Amazon ECS i. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è	

Per configurare questo tipo di volume	Esegui questa operazione	
	configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Managemen t Service (AWS KMS) specificata nell'impo stazione. Se scegli Default e la crittografia predefini ta di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittogra fia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazi one predefinita o se crei un volume da un'istant anea crittografata. j. Se hai scelto Personali zzato per la crittografia, devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se	

Per configurare questo tipo di volume	Esegui questa operazione	
	scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizza zioni corrette definite nella tua politica. AWS KMS key Per ulteriori informazi oni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. k. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione dell'attività o fornendo tag personali zzati. Se desideri propagare i tag dalla definizio ne dell'attività, scegli Definizione dell'attività per Propagate i tag da. Se scegli Non propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto.	

Per configurare questo tipo di volume	Esegui questa operazione	
	Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS.	

9. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Task Placement (Posizionamento attività), quindi scegli una tra le seguenti opzioni.

   Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

   - AZ Balanced Spread: distribuisci le attività tra le zone di disponibilità e tra le istanze di container nella zona di disponibilità.
   - AZ Balanced BinPack: distribuisci le attività tra le zone di disponibilità e tra le istanze di container con la minima memoria disponibile.
   - BinPack— Distribuisci le attività in base alla quantità minima disponibile di CPU o memoria.
   - Un'attività per host: posiziona al massimo un'attività dal servizio su ogni istanza del contenitore.
   - Personalizzato: definisci la tua strategia di posizionamento delle attività.

   Se hai scelto Custom (Personalizzato), definisci l'algoritmo per il posizionamento delle attività e le regole che vengono prese in considerazione durante il posizionamento delle attività.

   - In Strategy (Strategia), per Type (Tipo) e Field (Campo), scegli l'algoritmo e l'entità da utilizzare per l'algoritmo.

     Puoi aggiungere un massimo di 5 strategie.
   - In Vincolo, per Tipo ed Espressione, scegli la regola e l'attributo per il vincolo.

     Ad esempio, per impostare il vincolo per posizionare le attività su istanze T2, per Expression (Espressione), immetti attribute:ecs.instance-type =~ t2.*.

Puoi aggiungere un massimo di 10 vincoli.

10. (Facoltativo) Per sovrascrivere il ruolo IAM dell'attività o il ruolo di esecuzione dell'attività definito nella definizione dell'attività, espandi Task overrides (Sostituzione dei processi), quindi completa la procedura seguente:

    a. Per Ruolo attività, scegli un ruolo IAM per questa attività. Per ulteriori informazioni, consulta Ruolo IAM dell'attività Amazon ECS.

       Vengono visualizzati solo i ruoli con la relazione di attendibilità `ecs-tasks.amazonaws.com`. Per istruzioni su come creare manualmente un ruolo IAM, consulta Creazione del ruolo IAM dell'attività.

    b. Per Ruolo di esecuzione attività, scegli un ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

11. (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 ulteriori informazioni sul comando di esecuzione di Docker, consulta Riferimento Docker Run nel manuale di riferimento di Docker.
    - 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."
      ```

12. (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).

13. Scegli Crea.

Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS

EventBridge Scheduler è uno scheduler senza server che consente di creare, eseguire e gestire attività da un unico servizio gestito centralizzato. Fornisce funzionalità di pianificazione una tantum e ricorrenti indipendentemente dai bus e dalle regole degli eventi. EventBridge Scheduler è altamente personalizzabile e offre una migliore scalabilità rispetto alle regole EventBridge pianificate, con un set più ampio di operazioni e servizi API 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

  Puoi configurare le pianificazioni una tantum in qualsiasi fuso orario.

Puoi pianificare Amazon ECS utilizzando Amazon EventBridge Scheduler.

Sebbene sia possibile creare un'attività pianificata nella console Amazon ECS, attualmente la console EventBridge Scheduler offre più funzionalità.

Prima di pianificare un'attività, completa la procedura seguente:

1. Utilizza la console VPC per ottenere gli ID delle sottoreti in cui vengono eseguite le attività e gli ID dei gruppi di sicurezza per le sottoreti. Per ulteriori informazioni, consulta Visualizzazione delle sottoreti e Visualizzazione dei gruppi di sicurezza nella Guida per l'utente di Amazon VPC.
2. Configura il ruolo di esecuzione di EventBridge Scheduler. Per ulteriori informazioni, consulta Configurare il ruolo di esecuzione nella Amazon EventBridge Scheduler User Guide.

Per creare una nuova pianificazione utilizzando la console

1. Apri la console Amazon EventBridge Scheduler all'indirizzo https://console.aws.amazon.com/scheduler/home.
2. Nella pagina Pianificazioni, scegli Crea pianificazione.
3. Nella pagina Specifica i dettagli della pianificazione, nella sezione Nome e descrizione della pianificazione, effettua le seguenti operazioni:

   a. Per Nome pianificazione, inserisci un nome per la pianificazione. Ad esempio, **`MyTestSchedule`**.

   b. (Facoltativo) Per Descrizione, inserisci una descrizione per la pianificazione. Ad esempio, **`TestSchedule`**.

   c. Per Schedule group, scegli un gruppo di pianificazione. Se non hai un gruppo, scegli predefinito. Per creare un gruppo di pianificazioni, scegli crea la tua pianificazione.

      I gruppi di pianificazione vengono utilizzati per aggiungere tag a gruppi di pianificazioni.
4. Scegli le opzioni di pianificazione.

Ricorrenza	Esegui questa operazione…
Pianificazione una tantum 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 ta utilizzando un'espressione cron o un'espressione rate.	a. Per Tipo di pianificazione, esegui una delle seguenti operazioni: • Per utilizzare un'espres sione Cron per definire la pianificazione, scegli

Ricorrenza	Esegui questa operazione…
	Pianificazione basata su cron e immetti l'espressione Cron. • Per utilizzare un'espressione di frequenza per definire la pianificazione, scegli Pianificazione basata su frequenza e inserisci l'espressione di frequenza. Per ulteriori informazioni sulle espressioni cron e rate, consulta Schedule types on EventBridge Scheduler nella Amazon EventBridge Scheduler User Guide. b. Per Finestra temporale flessibile, scegli Disattivata 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. Seleziona Successivo.

7. Nella pagina Seleziona destinazione, procedi come segue:

a. Scegli Tutte le API, quindi inserisci ECS nella casella di ricerca.

b. Seleziona Amazon ECS.

c. Nella casella di ricerca, inserisci RunTask, quindi scegli. RunTask

d. Per Cluster ECS, scegli il cluster.

e. Per Attività ECS, scegli la definizione dell'attività da utilizzare.

f. Per utilizzare un tipo di avvio, espandi Opzioni di calcolo, quindi seleziona Tipo di avvio. Quindi, scegli il tipo di avvio.

Quando viene specificato il tipo di avvio Fargate, per Versione piattaforma inserisci la versione della piattaforma da utilizzare. Se non è specificata, viene utilizzata la versione della piattaforma `LATEST`.

g. Per Sottoreti, inserisci gli ID delle sottoreti in cui eseguire l'attività.

h. Per Gruppi di sicurezza, inserisci gli ID dei gruppi di sicurezza per la sottorete.

i. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Vincolo di posizionamento, quindi inserisci i vincoli.

Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

j. (Facoltativo) Per identificare le attività, configura i tag nella sezione Tag.

Per fare in modo che Amazon ECS assegni automaticamente i tag a tutte le attività appena avviate con i tag di definizione delle attività, seleziona Abilita tag gestiti da Amazon ECS.

8. Seleziona 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, inserite il numero massimo di ore e minuti in cui EventBridge Scheduler deve conservare un evento non elaborato.
   - La durata massima è 24 ore.
   - Per Numero massimo di tentativi, inserisci il numero massimo di volte in cui EventBridge Scheduler riprova la pianificazione se la destinazione restituisce un errore.

     Il valore massimo è 185 tentativi.

   Con le politiche di ripetizione dei tentativi, se una pianificazione non riesce a richiamare l'obiettivo, EventBridge Scheduler esegue nuovamente la pianificazione. Se configurato, è necessario impostare il tempo di conservazione massimo e i nuovi tentativi per la pianificazione.

c. Scegli dove EventBridge Scheduler archivia gli eventi non consegnati.

Opzione Dead-letter queue (DLQ)	Esegui questa operazione…	
Non conservare	Scegliere None (Nessuno).	
Archivia l'evento nello stesso spazio in Account AWS cui stai creando il programma	a. Scegli Seleziona una coda Amazon SQS in my Account AWS as a DLQ. b. Scegli il nome della risorsa Amazon (ARN) della coda di Amazon SQS.	

Opzione Dead-letter queue (DLQ)	Esegui questa operazione…	
Archivia l'evento in un luogo diverso Account AWS da quello in cui stai creando il programma	a. Scegli Specificare una coda Amazon SQS tra le altre Account AWS come DLQ. b. Inserisci il nome della risorsa Amazon (ARN) della coda di Amazon SQS.	

d. Per utilizzare una chiave gestita dal cliente per crittografare l'input di destinazione, in Crittografia scegli Personalizza le impostazioni di crittografia (avanzate).

   Se scegli questa opzione, inserisci l'ARN di una chiave KMS esistente scegli Crea una AWS KMS key per accedere alla console AWS KMS . Per ulteriori informazioni su come EventBridge Scheduler crittografa i dati inattivi, consulta Encryption at rest nella Amazon EventBridge Scheduler User Guide.

e. Per Autorizzazioni, scegli Usa ruolo esistente, quindi seleziona il ruolo.

   Per fare in modo che EventBridge Scheduler crei un nuovo ruolo di esecuzione per te, scegli Crea nuovo ruolo per questa pianificazione. Inserisci, quindi, un nome per Nome ruolo. Se scegli questa opzione, EventBridge Scheduler assegna al ruolo le autorizzazioni necessarie per la destinazione basata sul modello.

10. Seleziona Successivo.
11. Nella pagina Rivedi e crea pianificazione, rivedi i dettagli della pianificazione. In ogni sezione, scegli Modifica per tornare a tale passaggio e modificarne i dettagli.
12. Scegli Crea pianificazione.

    Puoi visualizzare un elenco delle pianificazioni nuove ed esistenti nella pagina Pianificazioni. Nella colonna Stato, accertati che la nuova pianificazione sia Abilitata.

Passaggi successivi

È possibile utilizzare la console EventBridge Scheduler o gestire la pianificazione. AWS CLI Per ulteriori informazioni, consulta Managing a planning nella Amazon EventBridge Scheduler User Guide.

Interruzione di un'attività Amazon ECS

Se non hai più bisogno di mantenere in esecuzione un'attività autonoma, puoi interromperla. La console Amazon ECS semplifica l'interruzione di una o più attività.

Se desideri interrompere un servizio, consultaEliminazione di un servizio Amazon ECS tramite la console.

Per interrompere un'attività autonoma ()AWS Management Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Cluster, scegli il cluster per accedere alla pagina dei dettagli del cluster.
4. Nella pagina dei dettagli del cluster, scegli la scheda Attività.
5. Puoi filtrare le attività per tipo di avvio utilizzando l'elenco Filtra tipo di avvio.

Attività da interrompere	Fasi
Una o più	a. Seleziona le attività, quindi scegli Stop, Stop selected. b. Nella pagina di conferma dell'interruzione dell'atti vità, scegli Interrompi
Tutti	**Important** Se scegli di interrompere tutte le attività utilizzando la console, Amazon ECS interrompe tutte le attività autonome

Attività da interrompere	Fasi
	e le attività che fanno parte di un servizio. Pertanto, raccomandiamo cautela quando si utilizza questa opzione. a. Seleziona Interrompi, Interrompi tutto. b. Nella pagina di conferma dell'interruzione dell'attività, inserisci Interrompi tutte le attività, quindi scegli Interrompi.

Servizi Amazon ECS

Puoi utilizzare un servizio Amazon ECS per eseguire e mantenere simultaneamente un determinato numero di istanze di una definizione di attività in un cluster Amazon ECS. Se una delle tue attività non riesce o si interrompe, il pianificatore del servizio Amazon ECS avvia un'altra istanza della definizione di attività per sostituirla. Ciò consente di mantenere il numero desiderato di attività nel servizio.

A scelta, puoi eseguire il tuo servizio con un load balancer. Il load balancer distribuisce il traffico nelle attività associate al servizio.

Ti consigliamo di utilizzare il pianificatore di servizi per le applicazioni e i servizi stateless di lunga durata. Il pianificatore di servizi garantisce che venga seguita la strategia di pianificazione specificata e ripianifica i processi qualora un processo dia esito negativo. Ad esempio, se l'infrastruttura sottostante riporta un errore, il pianificatore del servizio può riprogrammare un'attività. Puoi applicare vincoli e strategie di posizionamento delle attività per personalizzare il modo in cui il pianificatore posiziona e termina le attività. Se un processo in un servizio viene arresto, il pianificatore avvia un nuovo processo per sostituirlo. Questo processo continua finché il servizio non raggiunge il numero di attività desiderato in base alla strategia di pianificazione utilizzata dal servizio. La strategia di pianificazione del servizio è indicata anche come tipo di servizio.

Il pianificatore di servizi sostituisce inoltre le attività ritenute non integre dopo l'esito negativo di un controllo dell'integrità del container o di un sistema di bilanciamento del carico del gruppo di destinazione. Questa sostituzione dipende dai parametri di definizione del servizio `maximumPercent` e `desiredCount`. Se un'attività è contrassegnata come non integra, il pianificatore di servizi avvierà innanzitutto un'attività di sostituzione. Quindi, accade quanto segue.

- Se lo stato di integrità dell'attività sostitutiva è pari a`HEALTHY`, l'utilità di pianificazione del servizio interrompe l'attività non integra
- Se lo stato di integrità dell'attività di sostituzione è `UNHEALTHY`, il pianificatore interromperà l'attività di sostituzione non integra o l'attività esistente non integra per far sì che il numero totale delle attività sia pari a `desiredCount`.

Se il parametro `maximumPercent` impedisce al pianificatore di avviare un'attività di sostituzione, il pianificatore interromperà un'attività non integra alla volta, in modo casuale, per liberare spazio, e poi avvierà un'attività di sostituzione. Il processo di avvio e arresto continua fino a quando tutte le attività non integre vengono sostituite con attività integre. Dopo aver sostituito tutte le attività non integre e aver avviato solo quelle integre, se il numero totale delle attività supera `desiredCount`, le attività integre vengono interrotte casualmente fino a quando il numero totale delle attività è pari a `desiredCount`. Per ulteriori informazioni sui parametri `maximumPercent` e `desiredCount`, consulta Parametri di definizione del servizio.

Il pianificatore del servizio include una logica che limita la frequenza dei tentativi di riavvio delle attività se queste non riescono a riavviarsi ripetutamente. Se un'attività si interrompe senza essere entrata in uno stato `RUNNING`, il pianificatore del servizio inizia a rallentare in modo incrementale i tentativi di avvio ed emette un messaggio di evento del servizio. Questo comportamento previene l'utilizzo di risorse non necessarie per le attività non riuscite prima che sia risolto il problema. Dopo l'aggiornamento del servizio, il pianificatore del servizio riprende il normale comportamento. Per ulteriori informazioni, consulta Logica di accelerazione del servizio Amazon ECS e Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS.

Sono disponibili due strategie del pianificatore del servizio:

- `REPLICA`: la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le decisione riguardo al posizionamento delle attività. Per ulteriori informazioni, consulta Strategia di replica.

- `DAEMON`: la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Quando si utilizza questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta Strategia daemon.

> **Note**
>
> I processi Fargate non supportano la strategia di pianificazione `DAEMON`

Strategia daemon

La strategia di pianificazione daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Il service scheduler valuta i vincoli di posizionamento delle attività per le attività in esecuzione e interrompe le attività che non soddisfano i vincoli di posizionamento. Quando si utilizza questa strategia, non è necessario specificare il numero desiderato di attività, una strategia di posizionamento delle attività o utilizzare le politiche di Service Auto Scaling.

Amazon ECS riserva risorse di calcolo dell'istanza di container tra cui CPU, memoria e interfacce di rete per i processi del daemon. Quando avvii un servizio daemon in un cluster con altri servizi di replica, Amazon ECS assegna la priorità all'attività del daemon. Ciò significa che l'attività daemon è la prima attività da avviare sulle istanze e l'ultima attività da interrompere dopo l'interruzione di tutte le attività di replica. La strategia garantisce che le risorse non vengano utilizzate da attività di replica in sospeso e che siano disponibili per le attività del daemon.

Il pianificatore del servizio del daemon non posiziona alcuna attività sulle istanze con stato `DRAINING`. Se un'istanza di container passa a uno stato `DRAINING`, le attività daemon in esecuzione su di essa vengono arrestate. Il pianificatore del servizio inoltre monitora l'aggiunta di nuove istanze di container al cluster, alle quali aggiunge le attività daemon.

Quando si specifica una configurazione di distribuzione, il valore del `maximumPercent` parametro deve essere `100` (specificato in percentuale), che è il valore predefinito utilizzato se non è impostato. Il valore predefinito per il `minimumHealthyPercent` parametro è `0` (specificato come percentuale).

Quando modifichi i vincoli di posizionamento per il servizio daemon devi riavviare il servizio. Amazon ECS aggiorna dinamicamente le risorse riservate sulle istanze idonee per l'attività del daemon. Per le istanze esistenti, il pianificatore prova a posizionare l'attività sull'istanza.

Una modifica della dimensione del processo o della prenotazione della risorsa del container nella definizione di attività avvia una nuova implementazione. Amazon ECS raccoglie la CPU e le prenotazioni di memoria aggiornate per il daemon e quindi blocca tale capacità per il processo del daemon.

Se ci sono risorse insufficienti per uno dei casi precedenti, si verifica quanto segue:

- Il posizionamento del processo ha esito negativo.
- Viene generato un CloudWatch evento.
- Amazon ECS continua a provare a pianificare il processo sull'istanza attendendo che le risorse diventino disponibili.
- Amazon ECS libera tutte le istanze riservate che non soddisfano più i criteri di vincolo di posizionamento e interrompe i processi del daemon corrispondenti.

La strategia di pianificazione del daemon può essere utilizzata nei seguenti casi:

- Esecuzione di container di applicazioni
- Esecuzione di container di supporto per processi di registrazione, monitoraggio e traccia

Le attività che utilizzano il tipo di avvio Fargate oppure i tipi di controller di implementazione `CODE_DEPLOY` o `EXTERNAL` non supportano la strategia di pianificazione del daemon.

Quando il pianificatore di servizi arresta i processi in esecuzione, prova a mantenere il bilanciamento tra le zone di disponibilità nel cluster. Il pianificatore utilizza la logica seguente:

- Se è stata definita una strategia di posizionamento, utilizzare tale strategia per selezionare le attività da terminare. Ad esempio, se per un servizio è definita una strategia di distribuzione tra zone di disponibilità, viene selezionata un'attività che lascia le attività rimanenti con la migliore distribuzione.
- Se non è stata definita una strategia di posizionamento, mantieni il bilanciamento tra le zone di disponibilità nel cluster con la logica seguente:
  - Ordina le istanze di container valide. Dai priorità alle istanze che hanno il maggior numero di attività in esecuzione per questo servizio nella rispettiva zona di disponibilità. Ad esempio, se nella zona A è presente un'attività del servizio in esecuzione e nelle zone B e C ne sono presenti due, le istanze di container nella zona B o C sono considerate ottimali per l'arresto.

- Arresta l'attività del servizio in un'istanza di container in una zona di disponibilità ottimale in base alle fasi precedenti. Dai la priorità alle istanze di container con il maggior numero di attività in esecuzione per questo servizio.

Strategia di replica

La strategia di pianificazione delle repliche posiziona e gestisce il numero desiderato di attività nel cluster.

Per un servizio che esegue attività su Fargate, quando il pianificatore di servizi avvia nuove attività o arresta quelle in esecuzione, il pianificatore fa del suo meglio per mantenere il bilanciamento tra le zone di disponibilità. Non è necessario specificare strategie di posizionamento delle attività o vincoli.

Quando crei un servizio che esegue le attività su istanze EC2, puoi specificare facoltativamente le strategie e i vincoli di posizionamento delle attività per personalizzare le decisioni relative al 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 router dei log del daemon da eseguire prima della registrazione delle attività), crea un vincolo di posizionamento delle attività che assicuri che le attività del servizio daemon vengano posizionate sull'istanza EC2 prima delle attività del servizio di replica. Per ulteriori informazioni, consulta Esempio di vincoli di posizionamento delle attività di Amazon ECS.
- Se hai definito una strategia di posizionamento, utilizzala per selezionare un'istanza tra le candidate rimanenti.
- Se non è stata definita una strategia di posizionamento, bilancia le attività tra le zone di disponibilità nel cluster con la logica seguente:
  - Ordina le istanze di container valide. Dai priorità alle istanze che hanno il minor numero di attività in esecuzione per questo servizio nella rispettiva zona di disponibilità. Ad esempio, se nella

zona A è presente un'attività del servizio in esecuzione e nelle zone B e C nessuna, le istanze di container valide nella zona B o C sono considerate ottimali per il posizionamento.

- Posiziona la nuova attività del servizio in un'istanza di container valida in una zona di disponibilità ottimale in base alle fasi precedenti. Dai la priorità alle istanze di container con il minor numero di attività in esecuzione per questo servizio.

Best practice per i parametri del servizio Amazon ECS

Per garantire che non si verifichino tempi di inattività delle applicazioni, il processo di distribuzione è il seguente:

1. Avvia i nuovi contenitori di applicazioni mantenendo attivi i contenitori esistenti.
2. Verifica che i nuovi contenitori siano integri.
3. Fermate i vecchi contenitori.

A seconda della configurazione di distribuzione e della quantità di spazio libero e non riservato nel cluster, potrebbero essere necessari più cicli per completare la procedura, sostituire tutte le vecchie attività con nuove attività.

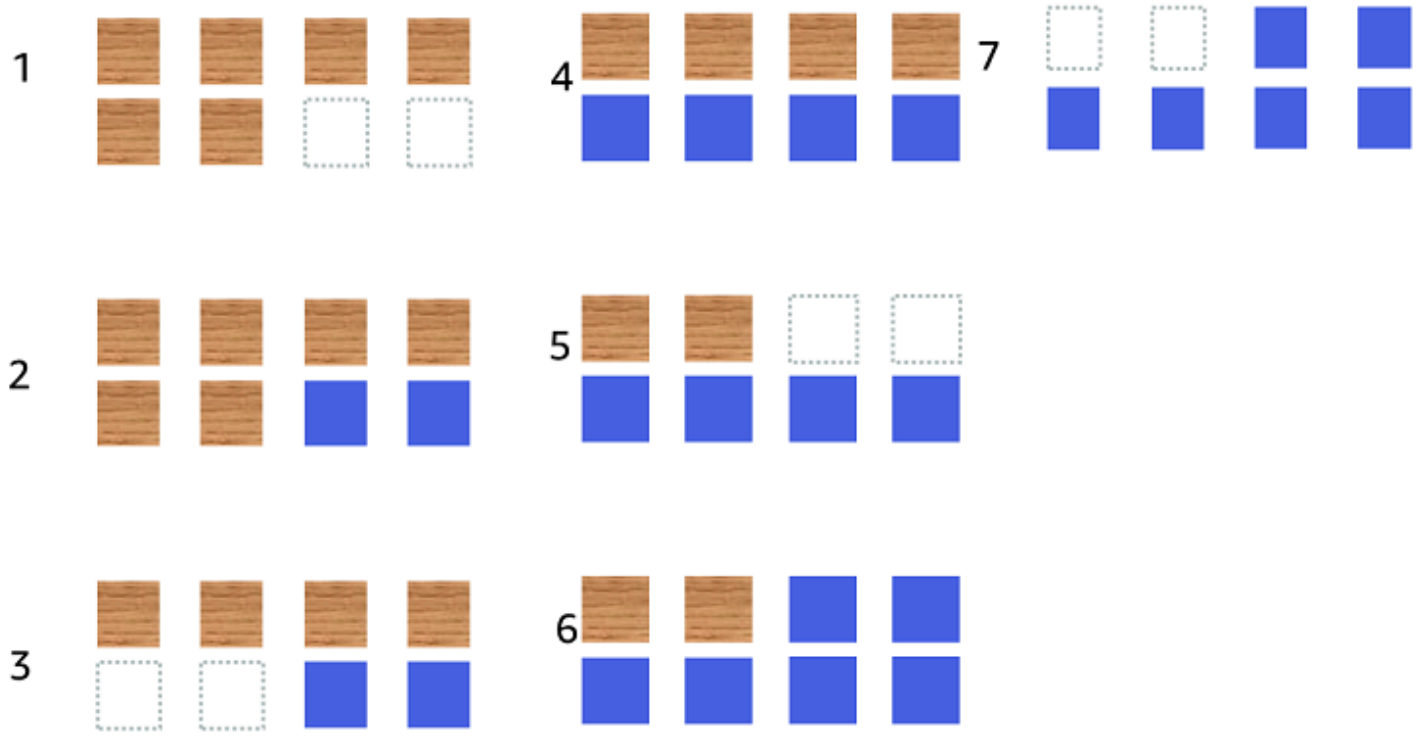
Esistono due opzioni di configurazione del servizio ECS che è possibile utilizzare per modificare il numero:

- `minimumHealthyPercent`: 100% (impostazione predefinita)

  Il limite inferiore al numero di attività del servizio che devono rimanere `RUNNING` invariate durante una distribuzione. Si tratta di una percentuale dell'`desiredCount`arrotondato al numero intero più vicino. Questo parametro consente di eseguire la distribuzione senza utilizzare capacità aggiuntiva del cluster.
- `maximumPercent`: 200% (impostazione predefinita)

  Il limite massimo al numero di attività del servizio consentite `PENDING` nello stato `RUNNING` o durante una distribuzione. Si tratta di una percentuale del numero `desiredCount` arrotondato per difetto al numero intero più vicino.

Prendiamo in considerazione il seguente servizio con sei task tan, distribuito in un cluster che può ospitare otto attività in totale. Le opzioni di configurazione del servizio Amazon ECS predefinite non consentono alla distribuzione di superare il 100% delle sei attività desiderate.



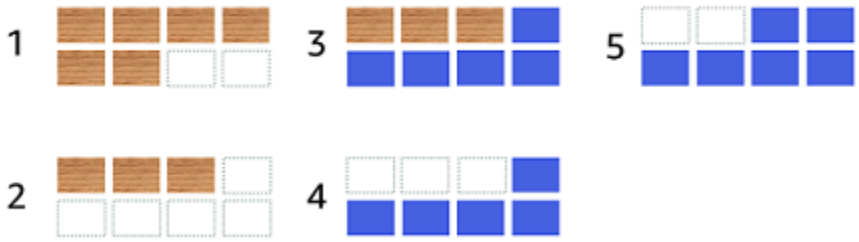
Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire le attività abbronzate con le attività blu.
2. L'utilità di pianificazione avvia due nuove attività blu perché le impostazioni predefinite richiedono che vi siano sei attività in esecuzione.
3. Lo scheduler interrompe due delle attività di colore marrone chiaro perché ci saranno un totale di sei attività (quattro marrone chiaro e due blu).
4. Lo scheduler avvia due attività blu aggiuntive.
5. Lo scheduler interrompe due delle seguenti attività.
6. Lo scheduler avvia due attività blu aggiuntive.
7. Lo scheduler chiude le ultime due attività abbronzate.

Nell'esempio precedente, se si utilizzano i valori predefiniti per le opzioni, c'è un'attesa di 2,5 minuti per ogni nuova attività che inizia. Inoltre, il sistema di bilanciamento del carico potrebbe dover attendere 5 minuti prima che la vecchia attività si interrompa.

È possibile velocizzare l'implementazione impostando il `minimumHealthyPercent` valore al 50%.

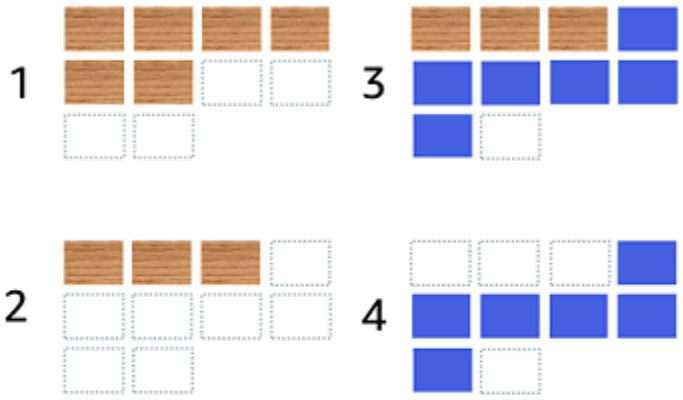
Prendiamo in considerazione il seguente servizio con sei task tan, distribuito in un cluster che può ospitare otto attività in totale.



Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire le attività abbronzate con le attività blu.
2. Lo scheduler interrompe tre delle attività tan. Sono ancora in esecuzione tre attività tan che soddisfano il `minimumHealthyPercent` valore.
3. Lo scheduler avvia cinque attività blu.
4. Lo scheduler interrompe le restanti tre attività abbronzate.
5. Lo scheduler avvia le ultime attività blu.

Puoi anche aggiungere altro spazio libero in modo da poter eseguire attività aggiuntive.



Il processo di distribuzione è il seguente:

1. L'obiettivo è sostituire le attività abbronzate con le attività blu.
2. Lo scheduler interrompe tre delle attività tan
3. Lo scheduler avvia sei attività blu
4. Lo scheduler interrompe le tre attività tan.

Utilizza i seguenti valori per le opzioni di configurazione del servizio Amazon ECS quando le attività sono inattive da qualche tempo e non hanno un tasso di utilizzo elevato.

- `minimumHealthyPercent`: 50%
- `maximumPercent`: 200%

Creazione di un servizio Amazon ECS utilizzando la console

Puoi creare un servizio utilizzando la console.

Quando utilizzi la console, considera i seguenti aspetti:

- Esistono due opzioni di calcolo che distribuiscono le attività.
  - Una strategia per provider di capacità fa sì che Amazon ECS distribuisca le attività in uno o più provider di capacità.
  - Un tipo di avvio fa sì che Amazon ECS avvii le attività direttamente su Fargate o sulle istanze Amazon EC2 registrate nei cluster.
- Le definizioni di processo che utilizzano la modalità di rete `awsvpc` o i servizi configurati per l'utilizzo di un load balancer devono disporre di una configurazione di rete. Di default, la console seleziona l'Amazon VPC di default insieme a tutte le sottoreti e il gruppo di sicurezza di default all'interno dell'Amazon VPC di default.
- La strategia di posizionamento delle attività predefinita distribuisce le attività in modo uniforme tra le zone di disponibilità.
- Quando utilizzi il tipo di avvio per l'implementazione del servizio, per impostazione predefinita il servizio viene avviato nelle sottoreti del cluster VPC.
- Per la strategia per provider di capacità, la console seleziona un'opzione di calcolo di default. Di seguito viene descritto l'ordine utilizzato dalla console per selezionare un valore di default:
  - Se il cluster dispone di una strategia di provider di capacità definita, questa è selezionata.

  - Se nel cluster non è stata definita una strategia predefinita per i provider di capacità, ma al cluster sono stati aggiunti i provider di capacità Fargate, viene selezionata una strategia personalizzata per i provider di capacità che utilizza il provider di `FARGATE` capacità.
  - Se nel cluster non è stata definita una strategia predefinita per i provider di capacità, ma al cluster sono stati aggiunti uno o più provider di capacità di gruppo Auto Scaling, viene selezionata l'opzione Usa personalizzato (avanzato) ed è necessario definire manualmente la strategia.
  - Se nel cluster non è presente una strategia di provider di capacità di default definita e non sono stati aggiunti provider di capacità al cluster, è selezionato il tipo di avvio Fargate.
- Le opzioni predefinite di rilevamento degli errori di distribuzione prevedono l'utilizzo dell'opzione di interruzione del circuito di distribuzione di Amazon ECS con l'opzione Rollback on failures.

  Per ulteriori informazioni, consulta In che modo l'interruttore di distribuzione di Amazon ECS rileva i guasti.
- Se desideri utilizzare l'opzione di distribuzione blu/verde, determina come si spostano le applicazioni. CodeDeploy Sono disponibili le seguenti opzioni:
  - CodeDeployDefault.ecs AllAt Once: sposta tutto il traffico verso il container Amazon ECS aggiornato contemporaneamente
  - CodeDeployDefault.ecsLinear10 PercentEvery 1Minutes: sposta il 10% del traffico ogni minuto fino a quando tutto il traffico non viene spostato.
  - CodeDeploydefault.ecsLinear10 3Minutes: sposta il 10% del traffico ogni 3 minuti fino a quando tutto il traffico non viene PercentEvery spostato.
  - CodeDeploydefault.ecsCanary10percent5minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato cinque minuti più tardi.
  - CodeDeploydefault.ecsCanary10Percent15minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato 15 minuti più tardi.
- Se hai bisogno di un'applicazione per connetterti ad altre applicazioni in esecuzione su Amazon ECS, determina l'opzione più adatta alla tua architettura. Per ulteriori informazioni, consulta Interconnetti i servizi Amazon ECS.
- È necessario utilizzare AWS CloudFormation o distribuire un servizio che utilizzi uno dei seguenti parametri: AWS Command Line Interface
  - Policy di monitoraggio con un parametro personalizzato
  - Servizio di aggiornamento: non è possibile aggiornare la configurazione di `awsvpc` rete e il periodo di tolleranza per il controllo dello stato di salute.

Per informazioni su come creare un servizio utilizzando il AWS CLI, vedere create-servicenella Guida AWS Command Line Interface di riferimento.

Per informazioni su come creare un servizio utilizzando AWS CloudFormation, consulta AWS::ECS::Servicela Guida per l'AWS CloudFormation utente.

Creazione rapida di un servizio

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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella pagina di navigazione, scegli Cluster.
3. Nella pagina Cluster, scegli il cluster in cui creare il servizio.
4. Nella scheda Services (Servizi), scegli Create (Crea).
5. In Deployment configuration (Configurazione dell'implementazione), specifica come viene implementata l'applicazione.
    a. Per Application type (Tipo di applicazione), scegli Service (Servizio).
    b. Per Definizione di processo, scegli la famiglia di definizioni di processi e la revisione da utilizzare.
    c. In Service name (Nome servizio), specifica un nome per il servizio.
    d. Per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e gestire nel servizio.

6. (Facoltativo) Per identificare il servizio e le attività, espandi la sezione Tags (Tag), quindi configura i tag.

   Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag di definizione delle attività, seleziona Turn on Amazon ECS managed tags (Attiva i tag gestiti da Amazon ECS), quindi seleziona Task definitions (Definizioni di attività).

   Per fare in modo che Amazon ECS contrassegni automaticamente tutte le attività appena avviate con il nome del cluster e i tag del servizio, seleziona Turn on Amazon ECS managed tags (Attiva i tag gestiti da Amazon ECS), quindi seleziona Service (Servizio).

   Aggiungi o rimuovi un tag.

   - [Aggiungi un tag] Scegli Add tag (Aggiungi tag), quindi effettuare le seguenti operazioni:
     - In Chiave, immetti il nome della chiave.
     - In Valore, immetti il valore della chiave.
   - [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).

Creazione di un servizio utilizzando parametri definiti

Per creare un servizio utilizzando parametri definiti, segui questi passaggi.

Per creare un servizio (console Amazon ECS)

1. Apri la console all'indirizzo 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).

Per avviare un servizio da	Fasi
Tipo di avvio	a. Nella pagina Definizioni delle attività, seleziona il pulsante di opzione accanto alla definizione dell'attività. b. Nel menu Distribuisci, scegli Crea servizio.

3. (Facoltativo) Scegli come vengono distribuite le attività nell'infrastruttura cluster. Espandi Compute configuration (Configurazione di calcolo) quindi scegli la tua opzione.

Metodo di distribuzione	Fasi
Strategia del provider di capacità	a. In Opzioni di calcolo, scegli Strategia del provider di capacità. b. Scegli una strategia: • Per utilizzare una strategia del provider di capacità predefinita del cluster, scegli Use cluster default (Usa impostazione predefinita del cluster. • Se il cluster non dispone di una strategia del provider di capacità predefinita o per utilizzare una strategia personalizzata, scegli Usa personalizzato, Aggiungi strategia del provider di capacità e definisci la strategia

Metodo di distribuzione	Fasi
	personalizzata specifica ndo 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 avvio	a. Nella sezione Compute option (Opzioni di calcolo), seleziona Launch type (Tipo di avvio). b. Per Launch type (Tipo di avvio), seleziona un tipo di avvio. c. (Facoltativo) Quando viene specificato il tipo di avvio Fargate, per Versione piattaforma specifica la versione della piattaforma da utilizzare. Se non è specificata, di default viene utilizzata la versione della piattaforma `LATEST`.

4. Per specificare come viene distribuito il servizio, vai alla sezione Configurazione di distribuzione, quindi scegli le opzioni.

a. Per Tipo di applicazione, lascia la scelta Servizio.

b. Per Task definition (Definizione di processo) e Revision (Revisione), scegli la famiglia di definizioni di attività e la revisione da utilizzare.

c. In Service name (Nome servizio), specifica un nome per il servizio.

d. 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.

e. Se hai scelto Replica, per Desired tasks (Attività desiderate), immetti il numero di attività da avviare e mantenere nel servizio.

f. Determina il tipo di implementazione per il servizio. Espandi le opzioni di distribuzione, quindi specifica i seguenti parametri.

Il tipo di distribuzione	Fasi	
Aggiornamento in sequenza	a. Per Min running tasks (Numero minimo di attività in esecuzione), specifica il limite inferiore per il numero di attività nel servizio che devono rimanere nello stato `RUNNING` durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Per ulteriori informazioni, consulta Configurazione dell'implementazione. b. Per Max running tasks (Numero massimo di attività in esecuzione), specifica il limite superiore per il numero di attività del servizio consentite nello stato `RUNNING` o `PENDING` durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per difetto al numero intero più vicino).	

Il tipo di distribuzione	Fasi	
Implementazione blu/verde	a. Per la configurazione della distribuzione, scegliete come CodeDeploy indirizzare il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione. b. Per Service role for CodeDeploy, scegli il ruolo IAM utilizzato dal servizio per autorizzare le richieste API Servizi AWS.	

g. Per configurare il modo in cui Amazon ECS rileva e gestisce gli errori di implementazione, espandi Deployment failure detection (Rilevamento degli errori di implementazione), quindi scegli le tue opzioni.

   i. Per interrompere un'implementazione quando le attività non possono essere avviate, seleziona Use the Amazon ECS deployment circuit breaker (Usa l'interruttore automatico di implementazione di Amazon ECS).

      Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di implementazione completato quando l'interruttore di distribuzione imposta la distribuzione su uno stato fallito, seleziona Rollback on failures.

   ii. Per interrompere una distribuzione in base alle metriche dell'applicazione, seleziona Usa CloudWatch allarmi. Quindi, dal nome CloudWatch dell'allarme, scegli gli allarmi. Per creare un nuovo allarme, vai alla CloudWatch console.

      Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di distribuzione completato quando un CloudWatch allarme imposta la distribuzione su uno stato fallito, seleziona Rollback in caso di errori.

5. (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 tuo servizio esegue un'applicazione client di rete che deve solo connettersi ad altri servizi nel namespace, scegli Solo lato client.
   - Se il servizio esegue un'applicazione di rete o di servizio Web, deve fornire endpoint per questo servizio e si connette ad altri servizi nello spazio dei nomi, scegli Client and server (Client e server).

b. Per utilizzare uno spazio dei nomi differente da quello del cluster predefinito, per Namespace (Spazio dei nomi), scegli lo spazio dei nomi del servizio.

c. (Facoltativo) Seleziona l'opzione Use log collection (Usa raccolta di log) per specificare una configurazione di log. Per ogni driver di log disponibile, sono disponibili opzioni del driver di log da specificare. L'opzione predefinita invia i log dei contenitori 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 dei container a Firehose. Sono disponibili le opzioni predefinite del driver di registro, che inviano i log a un flusso di distribuzione Firehose. Per specificare un nome del flusso di consegna diverso, modifica i valori dell'opzione del driver.
   - Amazon Kinesis Data Streams: configura l'attività per inviare i log dei container a Kinesis Data Streams. Vengono fornite le opzioni predefinite del driver di registro, che inviano i log a un flusso Kinesis Data Streams. Per specificare un nome del flusso diverso, modifica i valori dell'opzione del driver.
   - Amazon OpenSearch Service: configura l'attività per inviare i log dei container a un dominio OpenSearch di servizio. Devono essere fornite le opzioni del driver di log.
   - Amazon S3: configura l'attività per inviare i log dei container a un bucket Amazon S3. Vengono fornite le opzioni di driver di registro predefinite, ma è necessario specificare un nome di bucket Amazon S3 valido.

6. (Facoltativo) Per utilizzare Service Discovery, seleziona Usa service discovery, quindi specifica quanto segue.

   a. Per utilizzare un nuovo spazio dei nomi, scegli Crea un nuovo spazio dei nomi in Configura lo spazio dei nomi, quindi fornisci un nome e una descrizione del namespace. Per utilizzare uno spazio dei nomi esistente, scegli Seleziona uno spazio dei nomi esistente, quindi scegli lo spazio dei nomi che desideri utilizzare.

   b. Fornisci informazioni sul servizio Service Discovery come il nome e la descrizione del servizio.

   c. Per fare in modo che Amazon ECS esegua controlli periodici dello stato di salute a livello di contenitore, seleziona Abilita la propagazione dello stato delle attività di Amazon ECS.

   d. Per Tipo di record DNS seleziona il tipo di record DNS da creare per il servizio. Amazon ECS service discovery supporta solo i record A e SRV, a seconda della modalità di rete specificata dalla definizione dell'attività. Per informazioni su questi tipi di record, consulta Tipi di record DNS supportati nella Guida per gli sviluppatori di Amazon Route 53.

      - Se la definizione di attività specificata dalla tua attività di servizio usa la modalità di rete `bridge` o `host`, sono supportati solo i record di tipo SRV. Scegli un nome di container e una combinazione di porte da associare al record.
      - Se la definizione di attività specificata dalla tua attività di servizio usa la modalità di rete `awsvpc`, seleziona il tipo di record A o SRV. Se scegli A, vai al passaggio successivo. Se scegli SRV, specifica la porta sulla quale si trova il servizio oppure un nome di container e una combinazione di porte da associare al record.

      Per TTL, inserisci il tempo in secondi per cui un set di record viene memorizzato nella cache dai resolver DNS e dai browser Web.

7. (Facoltativo) Per configurare un load balancer per il servizio, espandi Load balancing (Bilanciamento del carico).

   Scegli il load balancer.

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione
Application Load Balancer	a. Per Seleziona tipo di load balancer, scegli 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 Load balancer name (Nome load balancer), immetti un nome univoco. d. Per Choose container to load balance (Scegli il container per il bilanciamento del carico), scegli il container che ospita il servizio. e. Per Listener, specifica una porta e un protocollo per l'Application Load Balancer su cui ascoltare le richieste di connessione. Di default, il load balancer sarà configurato per utilizzare la porta 80 e HTTP.

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 impostazi one predefinita, il gruppo di destinazione instrader à le richieste al primo container definito nella definizione di attività. g. Per il ritardo di degroregi strazione, inserite il numero di secondi in cui il sistema di bilanciam ento del carico modifiche rà lo stato di destinazione. `UNUSED` Il valore predefini to 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 connessio ne tra l'Application Load Balancer e il container.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	L'opzione predefinita è la directory root (/). i. Per Health check grace period (Periodo di tolleranza per il controllo dell'integrità) specifica il periodo di tempo (in secondi) durante il quale lo scheduler di servizi deve ignorare i controlli dell'integrità della destinazione Elastic Load Balancing non integri.	

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 bilanciam ento 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 destinazi one a cui il Network Load Balancer instraderà le richieste. Per impostazi one predefinita, il gruppo di destinazione instrader à le richieste al primo container definito nella definizione di attività. e. Per Ritardo di degregist razione, immettete il numero di secondi in cui il sistema di bilanciam ento del carico modifiche rà lo stato di destinazione.	

Per usare questo sistema di bilanciamento del carico	Esegui questa operazione	
	UNUSED Il valore predefinito è 300 secondi. f. Per Health check path (Percorso del controllo dell'integrità), specifica un percorso esistente all'interno del container in cui Network Load Balancer deve inviare periodicamente le richieste per verificare l'integrità della connessione tra l'Application Load Balancer e il container. L'opzione predefinita è la directory root (/). g. Per Health check grace period (Periodo di tolleranza per il controllo dell'integrità) specifica il periodo di tempo (in secondi) durante il quale lo scheduler di servizi deve ignorare i controlli dell'integrità della destinazione Elastic Load Balancing non integri.	

8. (Facoltativo) Per configurare il servizio Auto Scaling, espandi Service auto scaling, quindi specifica i seguenti parametri.

   a. Per utilizzare la scalabilità automatica del servizio, seleziona Service auto scaling (Scalabilità automatica del servizio).

b. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.

c. In Numero massimo di attività, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.

d. Scegli il tipo di policy. In Tipo di politica di scalabilità, scegli una delle seguenti opzioni.

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
Monitoraggio degli obiettivi	a. In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. b. In Policy name (Nome policy), immetti il nome della policy. c. Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. • Utilizzo della ServiceAverageCPU ECS: utilizzo medio della CPU del servizio. • ECS ServiceAverageMemoryUtilization: utilizzo medio della memoria del servizio. • ALB RequestCountPerTarget: Numero di richieste completate per destinazione in un gruppo target Application Load Balancer. d. Per Target value (Valore di destinazione), inserisci il valore mantenuto dal servizio per il parametro selezionato.	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	e. Per il periodo di recupero con scalabilità orizzontale, inserisci il periodo di tempo, in secondi, dopo un'attività di scalabilità orizzontale (aggiunta di attività) che deve trascorrere prima che possa iniziare un'altra attività di scalabilità orizzontale. f. Per il periodo di recupero scalabile, inserite la quantità di tempo, in secondi, che deve trascorrere dopo un'attività di scalabilità (rimozione delle attività) che deve trascorrere prima che possa iniziare un'altra attività scalabile. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Seleziona Disattiva la scalabilità orizzontale se desideri che la tua politica di scalabilità venga ridimensionata in base	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	all'aumento del traffico, ma non è necessario che venga adattata 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 Azione, seleziona se aggiungere, rimuovere o impostare un conteggio specifico desiderato per il servizio. • Se hai scelto di aggiungere o rimuovere attività, in Valore inserisci il numero di attività (o la percentuale di attività esistenti) da aggiungere o rimuovere quando viene avviata l'azione di ridimensionamento. Se hai scelto di impostare il conteggio desiderato, inserisci il numero di attività. Per Tipo, scegli se Valore è un numero intero o un valore percentuale del conteggio desiderato esistente. • Per Limite inferiore e Limite superiore, inserisci i relativi limiti della regolazione di dimensionamento per fasi. Per impostazi	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	one predefinita, il limite inferiore per una policy di aggiunta è la soglia di allarme e il limite superiore è positivo (+) infinito. Per impostazione predefinita, il limite superiore per una policy di rimozione è la soglia di allarme e il limite inferiore è negativo (-) infinito. • (Facoltativo) Aggiungi opzioni di dimension amento aggiuntive. Scegli Aggiungi nuova azione di ridimensi onamento, quindi ripeti i passaggi delle azioni di ridimensionamento. • Per Periodo di recupero, inserite la quantità di tempo, in secondi, per attendere che una precedent e attività di ridimensi onamento abbia effetto. Per una politica di aggiunta, si tratta del periodo successivo a un'attivi tà di scalabilità	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	orizzontale in cui la politica di scalabilità blocca le attività di scalabilità orizzontale e limita il numero di attività che possono essere scalate orizzontalmente alla volta. Per quanto riguarda una politica di rimozione, si tratta del periodo successivo a un'attività di scalabilità che deve trascorrere prima che possa iniziare un'altra attività scalabile.	

9. (Facoltativo) Per utilizzare una strategia di posizionamento delle attività diversa da quella predefinita, espandi Task Placement (Posizionamento attività), quindi scegli una tra le seguenti opzioni.

   Per ulteriori informazioni, consulta In che modo Amazon ECS colloca le attività sulle istanze di container.

   - AZ Balanced Spread: distribuisci le attività tra le zone di disponibilità e tra le istanze di container nella zona di disponibilità.
   - AZ Balanced BinPack: distribuisci le attività tra le zone di disponibilità e tra le istanze di container con la minima memoria disponibile.
   - BinPack— Distribuisci le attività in base alla quantità minima disponibile di CPU o memoria.
   - Un'attività per host: posiziona al massimo un'attività dal servizio su ogni istanza del contenitore.
   - Personalizzato: definisci la tua strategia di posizionamento delle attività.

Se hai scelto Custom (Personalizzato), definisci l'algoritmo per il posizionamento delle attività e le regole che vengono prese in considerazione durante il posizionamento delle attività.

- In Strategy (Strategia), per Type (Tipo) e Field (Campo), scegli l'algoritmo e l'entità da utilizzare per l'algoritmo.

  Puoi aggiungere un massimo di 5 strategie.
- In Vincolo, per Tipo ed Espressione, scegli la regola e l'attributo per il vincolo.

  Ad esempio, per impostare il vincolo per posizionare le attività su istanze T2, per Expression (Espressione), immetti attribute:ecs.instance-type =~ t2.*.

  Puoi aggiungere un massimo di 10 vincoli.

10. Se la tua definizione di attività utilizza la modalità di rete `awsvpc`, espandi Networking (Rete). Per specificare una configurazione personalizzata, completa la procedura riportata di seguito.

    a. Per VPC seleziona il VPC da utilizzare.

    b. Per Subnets (Sottoreti), seleziona una o più sottoreti nel VPC che lo scheduler di attività deve prendere in considerazione quando posiziona le attività.

    > **Important**
    >
    > Per la modalità di rete `awsvpc` sono supportate solo sottoreti private. I processi non ricevono indirizzi IP pubblici. Pertanto, è necessario un gateway NAT per l'accesso a Internet in uscita e il traffico Internet in entrata viene instradato attraverso un load balancer.

    c. Per Gruppi di sicurezza è possibile selezionare un gruppo di sicurezza esistente o crearne uno nuovo. Per utilizzare un gruppo di sicurezza esistente, seleziona il gruppo di sicurezza e passa alla fase successiva. Per creare un nuovo gruppo di sicurezza, scegliere Create a new security group (Crea un nuovo gruppo di sicurezza). È necessario specificare un nome e una descrizione del gruppo di sicurezza e aggiungere una o più regole in entrata per il gruppo di sicurezza.

11. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

Il nome del volume e il tipo di volume vengono configurati quando si crea una revisione della definizione dell'attività e non possono essere modificati durante la creazione di un servizio. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione di attività e creare un servizio utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per il tipo di volume EBS, scegli il tipo di volume EBS che desideri allegare all'attività. b. Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettete il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per `io1` i tipi di volume e. `io2` `gp3` d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). MiBps Questo valore è configurabile solo per il tipo di volume. `gp3`	

Per configurare questo tipo di volume	Esegui questa operazione	
	e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno snapshot se desideri creare un volume da uno snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Per Tipo di file system, scegli il tipo di file system che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico . L'impostazione predefini ta per Linux è`XFS`. Per i volumi creati da un'istant anea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. g. Per il ruolo Infrastru cture, scegli un ruolo IAM con le autorizzazioni	

Per configurare questo tipo di volume	Esegui questa operazione	
	necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy `AmazonECSInfrastructureRolePolicyForVolumes` gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Ruolo IAM dell'infrastruttura Amazon ECS. h. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specificata nell'impostazione. Se scegli Default	

Per configurare questo tipo di volume	Esegui questa operazione	
	e la crittografia predefinita di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittografia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazione predefinita o se crei un volume da un'istantanea crittografata. i. Se hai scelto Personalizzato per la crittografia, devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizzazioni corrette definite nella tua politica. AWS KMS	

Per configurare questo tipo di volume	Esegui questa operazione	
	key Per ulteriori informazioni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. j. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione o dal servizio dell'attività o fornendo tag personalizzati. Se desideri propagare i tag dalla definizione dell'attività, scegli Definizione attività per Propagare i tag da. Se desideri propagare i tag dal servizio, scegli Servizio da cui propaga i tag. Se scegli Non propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS,	

Per configurare questo tipo di volume	Esegui questa operazione	
	consulta Tagging dei volumi Amazon EBS.	

12. (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).

Aggiornamento di un servizio Amazon ECS tramite la console

Puoi aggiornare un servizio Amazon ECS utilizzando la console Amazon ECS. L'attuale configurazione del servizio è precompilata. Puoi aggiornare la definizione di attività, il numero desiderato di attività, la strategia del provider di capacità, la versione della piattaforma e la configurazione dell'implementazione (o qualsiasi combinazione di queste impostazioni).

Per informazioni su come aggiornare la configurazione di implementazione blu/verde, consulta Aggiornamento di una distribuzione blu/verde di Amazon ECS tramite la console.

Quando utilizzi la console, considera i seguenti aspetti:

Se desideri interrompere temporaneamente il servizio, imposta Attività desiderate su 0. Quindi, quando sei pronto per avviare il servizio, aggiorna il servizio con il conteggio originale delle attività desiderate.

Quando utilizzi la console, considera i seguenti aspetti:

- È necessario utilizzare il AWS Command Line Interface per aggiornare un servizio che utilizza uno dei seguenti parametri:
  - Distribuzioni blu/verde
  - Service Discovery: è possibile visualizzare solo la configurazione di Service Discovery.
  - Policy di monitoraggio con un parametro personalizzato
  - Servizio di aggiornamento: non è possibile aggiornare la configurazione di `awsvpc` rete e il periodo di tolleranza per il controllo dello stato di salute.

  Per informazioni su come aggiornare un servizio utilizzando il AWS CLI, vedere update-servicenella Guida AWS Command Line Interface di riferimento.
- Se stai modificando le porte utilizzate dai container in una definizione di attività, potrebbe essere necessario aggiornare i gruppi di sicurezza per le istanze di container in modo che funzionino con le porte aggiornate.
- Amazon ECS non aggiorna automaticamente i gruppi di sicurezza associati ai sistemi di bilanciamento del carico Elastic Load Balancing o alle istanze di container di Amazon ECS.
- Se il servizio utilizza un sistema di bilanciamento del carico, la configurazione di quest'ultimo definita per il servizio al momento della creazione non può essere modificata. Puoi invece utilizzare AWS CLI o l'SDK per modificare la configurazione del bilanciamento del carico. Per informazioni su come modificare la configurazione, consulta il riferimento UpdateServiceall'API di Amazon Elastic Container Service.
- Se aggiorni la definizione di attività per il servizio, il nome e la porta del container specificati nella configurazione del sistema di bilanciamento del carico devono rimanere nella definizione di attività.

Puoi aggiornare un servizio esistente per modificarne alcuni parametri di configurazione, ad esempio il numero di processi gestiti dal servizio, specificare quale definizione di attività viene utilizzata dai processi oppure, se i processi utilizzano il tipo di avvio Fargate, modificare la versione della piattaforma utilizzata dal servizio. Un servizio che utilizza una versione della piattaforma Linux non può essere aggiornato per utilizzare una versione della piattaforma Windows e viceversa. Se hai un'applicazione che ha bisogno di capacità maggiore, puoi aumentare la capacità del servizio. Se hai capacità superflua, puoi ridurre il numero di attività desiderate nel servizio e liberare risorse.

Se si desidera utilizzare un'immagine container aggiornata per le attività, è possibile creare una nuova revisione della definizione delle attività in tale immagine e distribuirla al servizio utilizzando l'opzione Forza nuova distribuzione nella console.

Il pianificatore del servizio utilizza i parametri di percentuale minima di attività integre e di percentuale massima (nella configurazione di implementazione del servizio) per determinare la strategia di distribuzione.

Se il servizio utilizza il tipo di distribuzione aggiornamento in sequenza (`ECS`), la percentuale minima di attività integre rappresenta il numero minimo di attività del servizio che devono rimanere nello stato `RUNNING` durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Il parametro si applica anche in presenza di istanze di container nello stato `DRAINING`, se il servizio contiene processi che utilizzano il tipo di avvio EC2. Utilizza questo parametro per eseguire l'implementazione senza impiegare capacità aggiuntiva del cluster. Ad esempio, se il servizio ha un numero desiderato di quattro attività e una percentuale minima di attività integre del 50%, lo scheduler può arrestare due attività esistenti per liberare capacità del cluster prima di avviare due nuove attività. Le attività dei servizi che non utilizzano un sistema di bilanciamento del carico vengono considerate integre se sono in stato `RUNNING`. Le attività per i servizi che utilizzano un sistema di bilanciamento del carico vengono considerate integre se sono in stato `RUNNING` e se sono considerate integre da tale sistema. Il valore predefinito per la percentuale minima di integrità è 100%.

Se il servizio utilizza il tipo di implementazione aggiornamento in sequenza (`ECS`), il parametro di percentuale massima rappresenta il numero massimo di attività del servizio che possono restare in stato `PENDING`, `RUNNING` o `STOPPING` durante un'implementazione, espresso come percentuale del numero di attività desiderato (arrotondata per difetto al numero intero più vicino). Il parametro si applica anche in presenza di istanze di container nello stato `DRAINING`, se il servizio contiene processi che utilizzano il tipo di avvio 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, Questo a condizione che le risorse del cluster necessarie per questa operazione siano disponibili. Il valore predefinito per la percentuale massima è 200%.

Quando il pianificatore del servizio sostituisce un'attività durante un aggiornamento, se il servizio utilizza un load balancer, rimuove prima l'attività da tale sistema e attende la fine delle connessioni. Quindi, viene emesso l'equivalente del comando docker stop ai container in esecuzione nell'attività. Questo determina un segnale `SIGTERM` e un timeout della durata di 30 secondi, dopo il quale viene inviato un segnale `SIGKILL` e i container vengono forzatamente arrestati. Se il container gestisce il segnale `SIGTERM` normalmente ed esce entro 30 secondi dalla ricezione, non viene inviato un segnale `SIGKILL`. Il pianificatore del servizio avvia e arresta le attività secondo quanto definito dalle impostazioni di percentuale minima di attività integre e percentuale massima.

Il pianificatore di servizi sostituisce inoltre le attività ritenute non integre dopo l'esito negativo di un controllo dell'integrità del container o di un sistema di bilanciamento del carico del gruppo di destinazione. Questa sostituzione dipende dai parametri di definizione del servizio `maximumPercent` e `desiredCount`. Se un'attività è contrassegnata come non integra, il pianificatore di servizi avvierà innanzitutto un'attività di sostituzione. Quindi, accade quanto segue.

- Se lo stato di integrità dell'attività sostitutiva è pari a`HEALTHY`, l'utilità di pianificazione del servizio interrompe l'attività non integra
- Se lo stato di integrità dell'attività di sostituzione è `UNHEALTHY`, il pianificatore interromperà l'attività di sostituzione non integra o l'attività esistente non integra per far sì che il numero totale delle attività sia pari a `desiredCount`.

Se il parametro `maximumPercent` impedisce al pianificatore di avviare un'attività di sostituzione, il pianificatore interromperà un'attività non integra alla volta, in modo casuale, per liberare spazio, e poi avvierà un'attività di sostituzione. Il processo di avvio e arresto continua fino a quando tutte le attività non integre vengono sostituite con attività integre. Dopo aver sostituito tutte le attività non integre e aver avviato solo quelle integre, se il numero totale delle attività supera `desiredCount`, le attività integre vengono interrotte casualmente fino a quando il numero totale delle attività è pari a `desiredCount`. Per ulteriori informazioni sui parametri `maximumPercent` e `desiredCount`, consulta Parametri di definizione del servizio.

> **Important**
>
> Se stai modificando le porte utilizzate dai container in una definizione di attività, potrebbe essere necessario aggiornare i gruppi di sicurezza per le istanze di container in modo che funzionino con le porte aggiornate.
> Se aggiorni la definizione di attività per il servizio, il nome del container e la porta del container specificati al momento della creazione del servizio devono rimanere nella definizione di attività.
> Amazon ECS non aggiorna automaticamente i gruppi di sicurezza associati ai sistemi di bilanciamento del carico Elastic Load Balancing o alle istanze di container di Amazon ECS.

Per aggiornare un servizio (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella pagina Clusters (Cluster), scegli il cluster.

3. Nella pagina dei dettagli del cluster, nella sezione Servizi, seleziona la casella di controllo accanto al servizio, quindi scegli Aggiorna.
4. Per fare in modo che il tuo servizio inizi una nuova implementazione, seleziona Force new deployment (Forza una nuova implementazione).
5. Per Definizione dell'attività, scegli la famiglia di definizioni dell'attività e la revisione.

   > **Important**
   >
   > La console verifica che la famiglia e la revisione delle definizioni delle attività selezionate siano compatibili con la configurazione di elaborazione definita. Se ricevi un avviso, verifica sia la compatibilità della definizione dell'attività che la configurazione di elaborazione selezionata.

6. In Attività desiderate, inserisci il numero di attività che desideri eseguire per il servizio.
7. Per Min running tasks (Numero minimo di attività in esecuzione), specifica il limite inferiore per il numero di attività nel servizio che devono rimanere nello stato `RUNNING` durante un'implementazione, espresso come percentuale del numero di attività desiderate (arrotondata per eccesso al numero intero più vicino). Per ulteriori informazioni, consulta Configurazione dell'implementazione.
8. 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).
9. Per configurare il modo in cui Amazon ECS rileva e gestisce gli errori di implementazione, espandi Deployment failure detection (Rilevamento degli errori di implementazione), quindi scegli le tue opzioni.

   a. Per interrompere un'implementazione quando le attività non possono essere avviate, seleziona Use the Amazon ECS deployment circuit breaker (Usa l'interruttore automatico di implementazione di Amazon ECS).

      Per fare in modo che il software ripristini automaticamente la distribuzione all'ultimo stato di distribuzione completato quando l'interruttore di distribuzione imposta la distribuzione su uno stato fallito, seleziona Rollback in caso di errori.

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.

10. Per modificare le opzioni di elaborazione, espandi la configurazione di Compute, quindi procedi come segue:

    a. Per i servizi attivi AWS Fargate, per la versione della piattaforma, scegli la nuova versione.

    b. Per i servizi che utilizzano una strategia per i provider di capacità, per la strategia dei fornitori di capacità, procedi come segue:

       - Per aggiungere un provider di capacità aggiuntivo, scegli Aggiungi altro. Quindi, scegli il provider in Provider di capacità.
       - Per rimuovere un provider di capacità, scegli Rimuovi a destra del provider.

       Un servizio che utilizza un provider di capacità di gruppo Auto Scaling non può essere aggiornato per utilizzare un provider di capacità Fargate. Un servizio che utilizza un provider di capacità Fargate non può essere aggiornato per utilizzare un provider di capacità di gruppo Auto Scaling.

11. (Facoltativo) Per configurare il servizio Auto Scaling, espandi Service auto scaling, quindi specifica i seguenti parametri.

    a. Per utilizzare la scalabilità automatica del servizio, seleziona Service auto scaling (Scalabilità automatica del servizio).

    b. In Numero minimo di attività, immettere il limite inferiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non scenderà al di sotto di questo conteggio.

    c. In Numero massimo di attività, immettere il limite superiore del numero di attività da utilizzare per la scalabilità automatica del servizio. Il numero desiderato non sarà superiore a questo conteggio.

    d. Scegli il tipo di policy. In Tipo di politica di scalabilità, scegli una delle seguenti opzioni.

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
Monitoraggio degli obiettivi	a. In Tipo di policy di dimensionamento, scegli Monitoraggio obiettivi. b. In Policy name (Nome policy), immetti il nome della policy. c. Per Parametro del servizio ECS, seleziona uno dei seguenti parametri. • Utilizzo della ServiceAverageCPU ECS: utilizzo medio della CPU del servizio. • ECS ServiceAverageMemoryUtilization: utilizzo medio della memoria del servizio. • ALB RequestCountPerTarget: Numero di richieste completate per destinazione in un gruppo target Application Load Balancer. d. Per Target value (Valore di destinazione), inserisci il valore mantenuto dal servizio per il parametro selezionato.	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	e. Per il periodo di recupero con scalabilità orizzontale, inserisci il periodo di tempo, in secondi, dopo un'attività di scalabilità orizzontale (aggiunta di attività) che deve trascorrere prima che possa iniziare un'altra attività di scalabilità orizzontale. f. Per il periodo di recupero scalabile, inserite la quantità di tempo, in secondi, che deve trascorrere dopo un'attività di scalabilità (rimozione delle attività) che deve trascorrere prima che possa iniziare un'altra attività scalabile. g. Per evitare che la policy esegua un'attività di riduzione orizzontale, seleziona Turn off scale-in (Disattiva riduzione orizzontale). h. • (Facoltativo) Seleziona Disattiva la scalabilità orizzontale se desideri che la tua politica di scalabilità venga ridimensionata in base	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	all'aumento del traffico, ma non è necessario che venga adattata 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 Azione, seleziona se aggiungere, rimuovere o impostare un conteggio specifico desiderato per il servizio. • Se hai scelto di aggiungere o rimuovere attività, in Valore inserisci il numero di attività (o la percentuale di attività esistenti) da aggiungere o rimuovere quando viene avviata l'azione di ridimensionamento. Se hai scelto di impostare il conteggio desiderato, inserisci il numero di attività. Per Tipo, scegli se Valore è un numero intero o un valore percentuale del conteggio desiderato esistente. • Per Limite inferiore e Limite superiore, inserisci i relativi limiti della regolazione di dimensionamento per fasi. Per impostazi	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	one predefinita, il limite inferiore per una policy di aggiunta è la soglia di allarme e il limite superiore è positivo (+) infinito. Per impostazione predefinita, il limite superiore per una policy di rimozione è la soglia di allarme e il limite inferiore è negativo (-) infinito. • (Facoltativo) Aggiungi opzioni di dimensionamento aggiuntive. Scegli Aggiungi nuova azione di ridimensionamento, quindi ripeti i passaggi delle azioni di ridimensionamento. • Per Periodo di recupero, inserite la quantità di tempo, in secondi, per attendere che una precedente attività di ridimensionamento abbia effetto. Per una politica di aggiunta, si tratta del periodo successivo a un'attività di scalabilità	

Per utilizzare questo tipo di policy...	Esegui questa operazione…	
	orizzontale in cui la politica di scalabilità blocca le attività di scalabilità orizzontale e limita il numero di attività che possono essere scalate orizzontalmente alla volta. Per quanto riguarda una politica di rimozione, si tratta del periodo successivo a un'attività di scalabilità che deve trascorrere prima che possa iniziare un'altra attività scalabile.	

12. (Facoltativo) Per utilizzare Service Connect, seleziona Turn on Service Connect (Attiva Service Connect), quindi specifica quanto segue:

    a. In Service Connect configuration (Configurazione Service Connect), specifica la modalità client.

        - Se il servizio esegue un'applicazione client di rete che deve connettersi solo ad altri servizi nello spazio dei nomi, scegli Client side only (Solo lato client).
        - Se il servizio esegue un'applicazione di rete o di servizio Web, deve fornire endpoint per questo servizio e si connette ad altri servizi nello spazio dei nomi, scegli Client and server (Client e server).

    b. Per utilizzare uno spazio dei nomi differente da quello del cluster predefinito, per Namespace (Spazio dei nomi), scegli lo spazio dei nomi del servizio.

13. Se l'attività utilizza un volume di dati compatibile con la configurazione al momento della distribuzione, è possibile configurare il volume espandendo Volume.

Il nome del volume e il tipo di volume vengono configurati quando si crea una revisione della definizione dell'attività e non possono essere modificati quando si aggiorna un servizio. Per aggiornare il nome e il tipo di volume, è necessario creare una nuova revisione della definizione di attività e aggiornare il servizio utilizzando la nuova revisione.

Per configurare questo tipo di volume	Esegui questa operazione	
Amazon EBS	a. Per il tipo di volume EBS, scegli il tipo di volume EBS che desideri allegare all'attività. b. Per Dimensione (GiB), immettere un valore valido per la dimensione del volume in gibibyte (GiB). È possibile specificare una dimensione del volume minima di 1 GiB e una massima di 16.384 GiB. Questo valore è obbligatorio a meno che non si fornisca un ID di istantanea. c. Per IOPS, immettete il numero massimo di operazioni di input/output (IOPS) che il volume deve fornire. Questo valore è configurabile solo per `io1` i tipi di volume e. `io2` `gp3` d. Per Throughput (MiB/s), immettere la velocità effettiva che il volume deve fornire, in mebibyte al secondo (o MiB/s). MiBps Questo valore è configurabile solo per il tipo di volume. `gp3`	

Per configurare questo tipo di volume	Esegui questa operazione	
	e. Per Snapshot ID, scegli uno snapshot del volume Amazon EBS esistente o inserisci l'ARN di uno snapshot se desideri creare un volume da uno snapshot. Puoi anche creare un nuovo volume vuoto non scegliendo o inserendo un ID snapshot. f. Per Tipo di file system, scegli il tipo di file system che verrà utilizzato per l'archiviazione e il recupero dei dati sul volume. È possibile scegliere il sistema operativo predefinito o un tipo di file system specifico . L'impostazione predefini ta per Linux è`XFS`. Per i volumi creati da un'istant anea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. g. Per il ruolo Infrastru cture, scegli un ruolo IAM con le autorizzazioni	

Per configurare questo tipo di volume	Esegui questa operazione	
	necessarie che consentano ad Amazon ECS di gestire i volumi Amazon EBS per le attività. Puoi allegare la policy `AmazonECSInfrastructureRolePolicyForVolumes` gestita al ruolo oppure puoi utilizzare la policy come guida per creare e allegare una policy personalizzata con autorizzazioni che soddisfino le tue esigenze specifiche. Per ulteriori informazioni sulle autorizzazioni necessarie, consulta Ruolo IAM dell'infrastruttura Amazon ECS. h. Per Encryption, scegli Default se desideri utilizzare la crittografia Amazon EBS per impostazioni predefinite. Se sul tuo account è configurata la crittografia di default, il volume verrà crittografato con la chiave AWS Key Management Service (AWS KMS) specificata nell'impostazione. Se scegli Default	

Per configurare questo tipo di volume	Esegui questa operazione	
	e la crittografia predefini ta di Amazon EBS non è attivata, il volume non verrà crittografato. Se scegli Personalizzato, puoi specificarne uno a tua scelta per la crittogra fia AWS KMS key dei volumi. Se scegli Nessuno, il volume non sarà crittografato a meno che la crittografia non sia configurata per impostazi one predefinita o se crei un volume da un'istant anea crittografata. i. Se hai scelto Personali zzato per la crittografia, devi specificare quello AWS KMS key che desideri utilizzare. Per la chiave KMS, scegli AWS KMS key o inserisci una chiave ARN. Se scegli di crittografare il tuo volume utilizzando una chiave simmetrica gestita dal cliente, assicurati di disporre delle autorizza zioni corrette definite nella tua politica. AWS KMS	

Per configurare questo tipo di volume	Esegui questa operazione	
	key Per ulteriori informazioni, consulta la sezione Crittografia dei dati per i volumi Amazon EBS. j. (Facoltativo) In Tag, puoi aggiungere tag al tuo volume Amazon EBS propagando i tag dalla definizione o dal servizio dell'attività o fornendo tag personalizzati. Se desideri propagare i tag dalla definizione dell'attività, scegli Definizione attività per Propagare i tag da. Se desideri propagare i tag dal servizio, scegli Servizio da cui propaga i tag. Se scegli Non propagare o se non scegli un valore, i tag non vengono propagati. Se desideri fornire i tuoi tag, scegli Aggiungi tag, quindi fornisci la chiave e il valore per ogni tag aggiunto. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS,	

Per configurare questo tipo di volume	Esegui questa operazione	
	consulta Tagging dei volumi Amazon EBS.	

14. (Facoltativo) Per identificare il tuo servizio, espandi la sezione Tags (Tag), quindi configura i tag.
    - [Aggiungi un tag] Scegli Aggiungi tag ed esegui le seguenti operazioni:
      - In Chiave, immetti il nome della chiave.
      - In Valore, immetti il valore della chiave.
    - [Rimuovere un tag] Accanto al tag, scegliere Remove tag (Rimuovi tag).
15. Scegli Aggiorna.

Aggiornamento di una distribuzione blu/verde di Amazon ECS tramite la console

Puoi aggiornare una configurazione di implementazione blu/verde utilizzando la console Amazon ECS. L'attuale configurazione di implementazione blu/verde è precompilata. Puoi aggiornare le seguenti opzioni di implementazione blu/verde:

- Nome del gruppo di distribuzione: le impostazioni di distribuzione CodeDeploy
- Nome dell'applicazione: il gruppo CodeDeploy di distribuzione
- Configurazione di distribuzione: in che modo CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione
- Listener di test sul load balancer: CodeDeploy utilizza il listener di test per indirizzare il traffico di test verso l'attività sostitutiva impostata durante una distribuzione

È necessario configurare la nuova opzione prima di aggiornare la configurazione.

Per aggiornare una configurazione di implementazione blu/verde (console Amazon ECS)

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella pagina Clusters (Cluster) seleziona il cluster.
3. Nella pagina Cluster overview (Panoramica del cluster), seleziona il servizio, quindi scegli Update (Aggiorna).

4. Espandi le opzioni di distribuzione - Powered by CodeDeploy, quindi scegli le opzioni da aggiornare:

   - Per modificare il gruppo CodeDeploy di distribuzione, per Nome applicazione, scegli il gruppo di distribuzione.
   - Per modificare le impostazioni CodeDeploy di distribuzione, per Nome del gruppo di distribuzione, scegli il gruppo.
   - Per modificare il modo in cui CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione, per la configurazione di Deployment, scegli l'opzione.

5. Seleziona gli hook degli eventi del ciclo di vita dell'implementazione e le relative funzioni Lambda da eseguire nell'ambito della nuova revisione dell'implementazione del servizio. Gli hook del ciclo di vita disponibili sono:

   - BeforeInstall— Utilizzate questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda prima della creazione del set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita non attiva un ripristino dello stato precedente.
   - AfterInstall— Utilizza questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda dopo la creazione del set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.
   - BeforeAllowTraffico: utilizza questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda prima che il traffico di produzione venga reindirizzato al set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.
   - AfterAllowTraffico: utilizza questo hook di eventi del ciclo di vita della distribuzione per richiamare una funzione Lambda dopo che il traffico di produzione è stato reindirizzato al set di attività sostitutivo. Il risultato della funzione Lambda per questo evento del ciclo di vita può attivare un ripristino dello stato precedente.

6. Per modificare il listener di test, espandi Load balancing, quindi per Test listener for deployment, scegli il listener di test. CodeDeploy
7. Scegli Aggiorna.

Eliminazione di un servizio Amazon ECS tramite la console

Puoi eliminare un servizio Amazon ECS utilizzando la console. Prima dell'eliminazione, la capacità del servizio viene automaticamente ridotta a zero. Le risorse del sistema di bilanciamento del carico o le risorse di rilevamento del servizio che sono associate al servizio non sono interessate dall'eliminazione del servizio. Per eliminare le risorse di Elastic Load Balancing, consulta uno dei seguenti argomenti, a seconda del tipo di sistema di bilanciamento del carico: Eliminazione di un Application Load Balancer o Eliminazione di Network Load Balancer.

Quando elimini un servizio, se ci sono ancora attività in esecuzione che richiedono la pulizia, lo stato del servizio passa da `ACTIVE` a `DRAINING` e il servizio non è più visibile nella console o nel funzionamento dell'`ListServices`API. Dopo che tutte le attività sono passate allo stato `STOPPING` o `STOPPED`, lo stato del servizio passa da `DRAINING` a `INACTIVE`. I servizi nello `INACTIVE` stato `DRAINING` o possono ancora essere visualizzati con l'operazione `DescribeServices` API.

> **Important**
>
> Se tenti di creare un nuovo servizio con lo stesso nome di un servizio esistente in uno `ACTIVE` dei due stati, riceverai un errore. `DRAINING`

Procedura

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nella pagina Clusters (Cluster), seleziona il cluster per il servizio.
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Nella pagina Cluster : ***name*** (Cluster: nome) scegli la scheda Services (Servizi).
5. Selezionare i servizi e quindi seleziona Delete (Elimina).
6. Per eliminare un servizio anche se non è stato ridotto a zero attività, seleziona Force delete service (Forza eliminazione servizio).
7. Alla richiesta di conferma, inserisci delete, quindi scegli Elimina.

Implementa i servizi Amazon ECS sostituendo le attività

Quando crei un servizio che utilizza il tipo di distribuzione rolling update (`ECS`), lo scheduler del servizio Amazon ECS sostituisce le attività attualmente in esecuzione con nuove attività. Il numero di

attività che Amazon ECS aggiunge o rimuove durante un aggiornamento in sequenza è controllato dalla configurazione dell'implementazione del servizio. La configurazione di implementazione comprende:

- La `minimumHealthyPercent` rappresenta il limite inferiore del numero di processi che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento, come percentuale del numero desiderato di processi per il servizio. Questo valore viene arrotondato per eccesso. Ad esempio, se la percentuale minima di integrità è `50`, il numero di processi desiderato è quattro, il pianificatore può interrompere due processi esistenti prima di avviare due nuovi processi. Allo stesso modo, se la percentuale di integrità minima è 75% e il numero di processi desiderato è due, il pianificatore non può interrompere alcun processo a causa del valore risultante che è anche due.

  Se le attività non funzionano correttamente, lo strumento di pianificazione dei servizi Amazon ECS avvierà prima le attività sostitutive e le manterrà finché le `minimumHealthyPercent` attività sostitutive non diventeranno integre. Man mano che le attività sostitutive vengono avviate e diventano corrette, quelle non sane verranno gradualmente interrotte.
- La `maximumPercent` rappresenta il limite superiore del numero di processi che devono essere in esecuzione per un servizio durante un'implementazione o quando un'istanza di container è in fase di svuotamento, come percentuale del numero desiderato di processi per il servizio. Questo valore viene arrotondato per difetto. Ad esempio, se la percentuale massima è `200` e il numero di processi desiderato è quattro, il pianificatore può avviare quattro nuovi processi prima di avviare arrestare i quattro processi esistenti. Allo stesso modo, se la percentuale di integrità massima è `125` e il numero di processi desiderato è tre, il pianificatore non può interrompere alcun processo a causa del valore risultante che è esso stesso tre.

> **Important**
>
> Quando si imposta una percentuale di integrità minima o massima, è necessario assicurarsi che lo scheduler possa arrestare o avviare almeno un'attività quando viene attivata un'implementazione. Se il servizio dispone di un'implementazione bloccata a causa di una configurazione di distribuzione non valida, verrà inviato un messaggio di evento del servizio. Per ulteriori informazioni, consulta service (*`nome-servizio`*) non è riuscito ad arrestare o avviare i processi durante un'implementazione a causa della configurazione della implementazione del servizio. Aggiorna il valore minimumHealthyPercent o MaximumPercent e riprova..

Un'implementazione in sequenza utilizza l'interruttore di implementazione per determinare se le attività raggiungono uno stato stazionario. L'interruttore di implementazione può facoltativamente eseguire il rollback di un'implementazione in caso di errore.

Rilevamento degli errori

Esistono due metodi che forniscono un sistema per identificare rapidamente quando un'implementazione non è riuscita e quindi, facoltativamente, per eseguire il rollback dell'esito negativo per ripristinare l'ultima implementazione funzionante.

- the section called “ In che modo l'interruttore di distribuzione rileva i guasti”
- the section called “In che modo gli CloudWatch allarmi rilevano gli errori di distribuzione”

I metodi possono essere utilizzati separatamente o insieme. Quando 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.

In che modo l'interruttore di distribuzione di Amazon ECS rileva i guasti

L'interruttore di implementazione è il meccanismo di aggiornamento in sequenza che determina se le attività raggiungono uno stato stazionario. L'interruttore di implementazione dispone di un'opzione che consente di eseguire automaticamente il rollback di un'implementazione con esito negativo a un'implementazione nello stato `COMPLETED`.

Quando la distribuzione di un servizio cambia stato, Amazon ECS invia un evento di modifica dello stato di distribuzione del servizio a EventBridge. Ciò fornisce un modo programmatico per monitorare lo stato delle implementazioni dei servizi. Per ulteriori informazioni, consulta Eventi di modifica dello stato di implementazione del servizio Amazon ECS. Ti consigliamo di creare e monitorare una EventBridge regola con un `eventName` of `SERVICE_DEPLOYMENT_FAILED` in modo da poter intraprendere azioni manuali per avviare la distribuzione. Per ulteriori informazioni, consulta Creating an EventBridge Rule nella 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 lo stato `COMPLETED`. Quando l'interruttore di implementazione non rileva un'implementazione nello stato `COMPLETED`, non avvia nuove attività e l'implementazione si blocca.

Quando crei un servizio, lo scheduler tiene traccia delle attività che non sono state avviate in due fasi.

- Fase 1: lo scheduler monitora le attività per vedere se passano allo stato RUNNING.
  - Operazione riuscita: la distribuzione ha la possibilità di passare allo stato COMPLETED perché più di un'attività è passata allo stato RUNNING. Il criterio di errore viene ignorato e l'interruttore passa alla fase 2.
  - Errore: alcune attività consecutive non sono passate allo stato RUNNING e la distribuzione potrebbe passare allo stato FAILED.
- Fase 2: la distribuzione entra in questa fase quando è presente almeno un'attività nello stato RUNNING. L'interruttore automatico controlla i controlli di integrità relativi alle attività della distribuzione corrente oggetto di valutazione. I controlli di integrità convalidati sono Elastic Load Balancing AWS Cloud Map , controlli dello stato del servizio e controlli dello stato dei container.
  - Operazione riuscita: almeno un'attività è in corso e i controlli di integrità sono stati superati.
  - Fallimento: le attività sostituite a causa di errori nei controlli di integrità hanno raggiunto la soglia di errore.

Considerate quanto segue quando utilizzate il metodo dell'interruttore automatico di distribuzione su un servizio. EventBridge genera la regola.

- La risposta `DescribeServices` fornisce informazioni sullo stato di un'implementazione, `rolloutState` e `rolloutStateReason`. Quando viene avviata una nuova implementazione, lo stato di implementazione inizia in `IN_PROGRESS`. Quando il servizio raggiunge uno stato stazionario, lo stato di implementazione passa a `COMPLETED`. Se il servizio non riesce a raggiungere uno stato stazionario e l'interruttore automatico è abilitato, l'implementazione passerà a uno stato `FAILED`. Una implementazione in uno stato `FAILED` non avvierà nuove attività.
- Oltre agli eventi di modifica dello stato dell'implementazione del servizio che Amazon ECS invia per le implementazioni che sono state avviate e completate, Amazon ECS invia anche un evento quando un'implementazione con l'interruttore automatico attivato non riesce. Questi eventi forniscono dettagli sul motivo per cui un'implementazione non è riuscita o se un'implementazione

è stata avviata a causa di un ripristino dello stato precedente. Per ulteriori informazioni, consulta Eventi di modifica dello stato di implementazione del servizio Amazon ECS.

- Se viene avviata una nuova implementazione perché un'implementazione precedente non è riuscita e si è verificato il ripristino dello stato precedente, il campo `reason` dell'evento di modifica dello stato di implementazione del servizio indicherà che l'implementazione è stata avviata a causa di un ripristino dello stato precedente.
- L'interruttore automatico di implementazione è supportato solo per i servizi Amazon ECS che utilizzano il controller di implementazione (`ECS`) dell'aggiornamento in sequenza.
- È necessario utilizzare la console Amazon ECS o, AWS CLI quando si utilizza, l'interruttore automatico di distribuzione con l' CloudWatch opzione. Per ulteriori informazioni, consulta the section called “Creazione di un servizio utilizzando parametri definiti” e create-service in Informazioni di riferimento sull'AWS Command Line Interface .

L'`create-service` AWS CLI esempio seguente mostra come creare un servizio Linux quando l'interruttore di distribuzione viene utilizzato con l'opzione rollback.

```
aws ecs create-service \
     --service-name MyService \
     --deployment-controller type=ECS \
     --desired-count 3 \
     --deployment-configuration "deploymentCircuitBreaker={enable=true,rollback=true}"
 \
     --task-definition sample-fargate:1 \
     --launch-type FARGATE \
     --platform-family LINUX \
     --platform-version 1.4.0 \
     --network-configuration
 "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=EN
```

Esempio:

L'implementazione 1 è in uno stato `COMPLETED`.

L'implementazione 2 non può essere avviata, quindi l'interruttore esegue il rollback all'implementazione 1. L'implementazione 1 passa allo stato `IN_PROGRESS`.

L'implementazione 3 viene avviata e non è presente alcuna implementazione nello stato `COMPLETED`, quindi l'implementazione non è in grado di eseguire il rollback o avviare attività.

Soglia di errore

L'interruttore automatico di implementazione calcola il valore di soglia e lo utilizza per determinare quando modificare lo stato della distribuzione in `FAILED`.

L'interruttore automatico di distribuzione ha una soglia minima di 3 e una soglia massima di 200 e utilizza i valori della formula seguente per determinare l'errore di distribuzione.

```
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 quando `RUNNING` nello stato sono presenti una o più attività. L'interruttore automatico di implementazione esegue controlli dell'integrità delle seguenti risorse per i processi nell'implementazione corrente:
   - Load balancer Elastic Load Balancing
   - AWS Cloud Map servizio
   - Controlli dell'integrità dei container di Amazon ECS

   Quando un controllo dell'integrità di un processo HA ESITO NEGATIVO, l'interruttore automatico di implementazione aumenta di uno il numero di errori. Quando il numero di errori è uguale alla soglia, l'implementazione viene contrassegnata come `FAILED`.

La tabella seguente mostra alcuni esempi.

Conteggio attività desiderato	Calcolo	Threshold
1	`3 <= 0.5 * 1 => 200`	3 (il valore calcolato è inferiore al minimo)
25	`3 <= 0.5 * 25 => 200`	13 (il valore viene arrotondato per eccesso)
400	`3 <= 0.5 * 400 => 200`	200
800	`3 <= 0.5 * 800 => 200`	200 (il valore calcolato è superiore al massimo)

Ad esempio, quando la soglia è 3, l'interruttore si avvia con il conteggio dei guasti impostato su 0. Quando un'attività non riesce a raggiungere `RUNNING` lo stato, l'interruttore automatico di dispiegamento aumenta il numero di guasti di uno. Quando il numero di errori è uguale a 3, la distribuzione viene contrassegnata come. `FAILED`

Per altri esempi sull'utilizzo dell'opzione di rollback, consulta Annuncio dell'interruttore automatico di implementazione di Amazon ECS.

In che modo CloudWatch gli allarmi rilevano gli errori di distribuzione di Amazon ECS

Puoi configurare Amazon ECS in modo che la distribuzione non sia riuscita quando rileva che uno specifico CloudWatch allarme è entrato nello `ALARM` stato.

Facoltativamente, è possibile impostare la configurazione per ripristinare un'implementazione non riuscita all'ultima implementazione completata.

L'`create-service` AWS CLI esempio seguente mostra come creare un servizio Linux quando gli allarmi di distribuzione vengono utilizzati con l'opzione rollback.

```
aws ecs create-service \
     --service-name MyService \
     --deployment-controller type=ECS \
     --desired-count 3 \
     --deployment-configuration
 "alarms={alarmNames=[alarm1Name,alarm2Name],enable=true,rollback=true}" \
     --task-definition sample-fargate:1 \
```

```
    --launch-type FARGATE \
    --platform-family LINUX \
    --platform-version 1.4.0 \
    --network-configuration
"awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=EN
```

Considera quanto segue quando utilizzi il metodo Amazon CloudWatch alarms su un servizio.

- Il tempo di incorporamento è un periodo di tempo dopo che una nuova versione del servizio è stata aumentata e la vecchia versione del servizio è stata ridotta, durante il quale Amazon ECS continua a monitorare l'allarme associato all'implementazione. Amazon ECS calcola questo periodo di tempo in base alla configurazione degli allarmi associata all'implementazione.
- Il parametro della richiesta `deploymentConfiguration` ora contiene il tipo di dati `alarms`. È possibile specificare i nomi degli allarmi, se utilizzare il metodo e se avviare un rollback quando gli allarmi indicano un errore di implementazione. Per ulteriori informazioni, consulta il riferimento CreateServiceall'API di Amazon Elastic Container Service.
- La risposta `DescribeServices` fornisce informazioni sullo stato di un'implementazione, `rolloutState` e `rolloutStateReason`. Quando viene avviata una nuova implementazione, lo stato di rollout inizia come `IN_PROGRESS`. Quando il servizio raggiunge uno stato stazionario ed è stato raggiunto il tempo di incorporamento, lo stato di rollout passa a `COMPLETED`. Se il servizio non riesce a raggiungere uno stato stazionario e l'allarme è passato allo stato `ALARM`, l'implementazione passerà a uno stato `FAILED`. Una implementazione in uno stato `FAILED` non avvierà nuovi processi.
- Oltre agli eventi di modifica dello stato dell'implementazione del servizio che Amazon ECS invia per le implementazioni che sono state avviate e completate, Amazon ECS invia un evento anche quando un'implementazione che utilizza gli allarmi non riesce. Questi eventi forniscono dettagli sul motivo per cui un'implementazione non è riuscita o se un'implementazione è stata avviata a causa di un ripristino dello stato precedente. Per ulteriori informazioni, consulta Eventi di modifica dello stato di implementazione del servizio Amazon ECS.
- Se una nuova implementazione viene avviata perché un'implementazione precedente non è riuscita ed è stato abilitato il ripristino dello stato precedente, il campo `reason` dell'evento di modifica dello stato di implementazione del servizio indicherà che l'implementazione è stata avviata a causa di un ripristino dello stato precedente.
- Se utilizzi l'interruttore di distribuzione e gli CloudWatch allarmi Amazon per rilevare i guasti, entrambi possono avviare un errore di distribuzione non appena vengono soddisfatti i criteri per entrambi i metodi. Un rollback si verifica quando si utilizza l'opzione di rollback per il metodo che ha restituito l'errore di implementazione.

- Gli CloudWatch allarmi Amazon sono supportati solo per i servizi Amazon ECS che utilizzano il controller di distribuzione rolling update (`ECS`).
- Puoi configurare questa opzione utilizzando la console Amazon ECS o il AWS CLI. Per ulteriori informazioni, consulta the section called “Creazione di un servizio utilizzando parametri definiti” e create-service in Informazioni di riferimento sull'AWS Command Line Interface .
- Potresti notare che lo stato di implementazione rimane `IN_PROGRESS` per un periodo di tempo prolungato. Questo perché Amazon ECS non modifica lo stato finché non ha eliminato l'implementazione attiva e ciò avviene solo dopo il tempo di incorporamento. A seconda della configurazione degli allarmi, l'implementazione potrebbe richiedere diversi minuti in più rispetto a quando non sono utilizzati gli allarmi (anche se il nuovo set di attività principali è stato aumentato e la vecchia implementazione è stata ridotta). Se utilizzi i CloudFormation timeout, valuta la possibilità di aumentarli. Per ulteriori informazioni, consulta Creazione delle condizioni di attesa in un modello nella Guida per l'utente di AWS CloudFormation .
- Amazon ECS chiama `DescribeAlarms` per eseguire il polling degli allarmi. Le chiamate verranno `DescribeAlarms` conteggiate ai fini delle quote di CloudWatch servizio associate al tuo account. Se disponi di altri AWS servizi che effettuano chiamate`DescribeAlarms`, Amazon ECS potrebbe avere un impatto sulla necessità di interrogare gli allarmi. Ad esempio, se un altro servizio effettua un numero di `DescribeAlarms` chiamate sufficiente a raggiungere la quota, tale servizio viene limitato e anche Amazon ECS lo è e non è in grado di generare allarmi. Se viene generato un allarme durante il periodo di limitazione, Amazon ECS potrebbe non ricevere l'allarme e il rollback potrebbe non verificarsi. Non ci sono altri impatti sull'implementazione. Per ulteriori informazioni sulle quote di CloudWatch servizio, consulta le quote di CloudWatch servizio nella Guida per l'utente. CloudWatch
- Se all'inizio di una implementazione un allarme si trova nello stato `ALARM`, Amazon ECS non monitorerà gli allarmi per la durata di tale implementazione (Amazon ECS ignora la configurazione degli allarmi). Questo comportamento risolve il caso in cui si desideri avviare una nuova distribuzione per correggere un errore di distribuzione iniziale.

Allarmi consigliati

È consigliabile utilizzare i seguenti parametri di allarme:

- Se utilizzi un Application Load Balancer, utilizza i parametri `HTTPCode_ELB_5XX_Count` e `HTTPCode_ELB_4XX_Count` di Application Load Balancer. Questi parametri controllano i picchi HTTP. Per ulteriori informazioni sulle metriche di Application Load Balancer, consulta CloudWatch le metriche per il tuo Application Load Balancer nella User Guide for Application Load Balancer.

- Per un'applicazione esistente, utilizza i parametri `CPUUtilization` e `MemoryUtilization`. Questi parametri controllano la percentuale di CPU e memoria utilizzate dal cluster o dal servizio. Per ulteriori informazioni, consulta the section called “Considerazioni”.
- Se utilizzi le Amazon Simple Queue Service code nelle tue attività, utilizza la metrica di `ApproximateNumberOfMessagesNotVisible` Amazon SQS. Questo parametro controlla il numero dei messaggi nella coda che vengono differiti e non sono disponibili per la lettura immediata. Per ulteriori informazioni sui parametri di Amazon SQS, consulta Parametri disponibili per CloudWatch Amazon SQS nella Amazon Simple Queue Service Developer Guide.

Convalida lo stato di un servizio Amazon ECS prima della distribuzione

Il tipo di distribuzione blu/verde utilizza il modello di distribuzione blu/verde controllato da. CodeDeploy Utilizza questo tipo di implementazione per verificare una nuova implementazione di un servizio prima di inviarvi traffico di produzione. Per ulteriori informazioni, consulta Cosa c'è CodeDeploy nella guida per l'AWS CodeDeploy utente. Convalida lo stato di un servizio Amazon ECS prima della distribuzione

Esistono tre modi in cui il traffico può variare durante una distribuzione blu/verde:

- Canarie: il traffico viene spostato in due incrementi. Puoi scegliere tra opzioni canary predefinite che specificano la percentuale del traffico reinstradato al set di attività aggiornato nel primo incremento e l'intervallo, in minuti, prima che il traffico rimanente venga reinstradato nel secondo incremento.
- Lineare: il traffico viene spostato in incrementi uguali con un numero uguale di minuti tra ogni incremento. Puoi scegliere tra opzioni lineari predefinite che specificano la percentuale del traffico reinstradato in ogni incremento e l'intervallo di tempo, in minuti, tra ciascun incremento.
- Tutto in una volta: tutto il traffico viene spostato contemporaneamente dal set di attività originale al set di attività aggiornato.

Di seguito sono riportati i componenti utilizzati da Amazon ECS quando un servizio utilizza il tipo di distribuzione blu/verde: CodeDeploy

CodeDeploy applicazione

Una raccolta di CodeDeploy risorse. È costituita da uno o più gruppi di distribuzione.

CodeDeploy gruppo di distribuzione

Le impostazioni di distribuzione. Comprendono:

- Cluster e servizio Amazon ECS
- Informazioni listener e gruppo target del load balancer
- Strategia di rollback dell'implementazione
- Impostazioni di reinstradamento del traffico
- Impostazioni di terminazione della revisione originale
- Configurazione dell'implementazione
- CloudWatch configurazione degli allarmi che può essere impostata per interrompere le distribuzioni
- Impostazioni SNS o CloudWatch Events per le notifiche

Per ulteriori informazioni, consulta Utilizzo di gruppi di implementazione nella Guida per l'utente di AWS CodeDeploy .

CodeDeploy configurazione di distribuzione

Speciifica in che modo CodeDeploy indirizza il traffico di produzione verso l'attività sostitutiva impostata durante una distribuzione. Sono disponibili le seguenti configurazioni predefinite di distribuzione lineare e canary. È inoltre possibile creare distribuzioni lineari e canary personalizzate. Per ulteriori informazioni, consulta Utilizzo di configurazioni di implementazione nella Guida per l'utente di AWS CodeDeploy .

- CodeDeployDefault.ecs AllAt Once: sposta tutto il traffico verso il container Amazon ECS aggiornato contemporaneamente
- CodeDeployDefault.ecsLinear10 PercentEvery 1Minutes: sposta il 10% del traffico ogni minuto fino a quando tutto il traffico non viene spostato.
- CodeDeploydefault.ecsLinear10 3Minutes: sposta il 10% del traffico ogni 3 minuti fino a quando tutto il traffico non viene PercentEvery spostato.
- CodeDeploydefault.ecsCanary10percent5minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato cinque minuti più tardi.
- CodeDeploydefault.ecsCanary10Percent15minutes: sposta il 10% del traffico nel primo incremento. Il restante 90% viene reinstradato 15 minuti più tardi.

Revisione

Una revisione è il file delle specifiche dell' CodeDeploy applicazione (file). AppSpec Nel AppSpec file, si specifica l'ARN completo della definizione dell'attività e il contenitore e la porta del set di attività sostitutivo in cui il traffico deve essere instradato quando viene creata una nuova distribuzione. Il nome del container deve essere uno dei nomi di container cui si fa riferimento nella definizione delle attività. Se la configurazione di rete o la versione della piattaforma è stata aggiornata nella definizione del servizio, è necessario specificare anche tali dettagli nel AppSpec file. Puoi inoltre specificare le funzioni Lambda da eseguire durante gli eventi del ciclo di vita dell'implementazione. Le funzioni Lambda consentono di eseguire i test e restituire i parametri durante l'implementazione. Per ulteriori informazioni, consulta AppSpec File Reference nella Guida AWS CodeDeploy per l'utente.

Considerazioni

Quando utilizzi il tipo di distribuzione blu/verde, tieni conto di quanto segue:

- Quando crei per la prima volta un servizio Amazon ECS che utilizza il tipo di implementazione blu/verde, viene creato anche un set di processi Amazon ECS.
- Il servizio deve essere configurato per utilizzare un sistema Application Load Balancer o un Network Load Balancer. Di seguito sono elencati i requisiti del load balancer:
  - È necessario aggiungere al load balancer un listener di produzione che viene utilizzato per instradare il traffico di produzione.
  - Al load balancer può essere aggiunto un test opzionale che viene utilizzato per instradare il traffico di test. Se specifichi un listener di test, CodeDeploy indirizza il traffico di test all'attività sostitutiva impostata durante una distribuzione.
  - I listener di produzione e di test devono appartenere entrambi allo stesso load balancer.
  - È necessario definire un gruppo target per il load balancer. Il gruppo target instrada il traffico verso l'attività originale impostata in un servizio attraverso il listener di produzione.
  - Quando viene utilizzato un Network Load Balancer, è supportata solo la configurazione dell'implementazione `CodeDeployDefault.ECSAllAtOnce`.
- Per i servizi configurati per utilizzare la scalabilità automatica del servizio e il tipo di implementazione blu/verde, la scalabilità automatica non viene bloccata durante un'implementazione, ma l'implementazione potrebbe non riuscire in alcune circostanze. Di seguito viene descritto questo comportamento in modo più dettagliato.

  - Se un servizio è scalabile e viene avviata una distribuzione, viene creato il set di attività verde e CodeDeploy aspetterà fino a un'ora prima che il set di attività verde raggiunga lo stato stazionario e non sposterà il traffico finché non lo farà.
  - Se un servizio è in fase di implementazione blu/verde e si verifica un evento di dimensionamento, il traffico continuerà a spostarsi per 5 minuti. Se il servizio non raggiunge lo stato stazionario entro 5 minuti, CodeDeploy interromperà la distribuzione e la contrassegnerà come fallita.
  - Se un servizio è in fase di implementazione blu/verde e si verifica un evento di dimensionamento, il numero dei processi desiderato potrebbe essere impostato su un valore imprevisto. Ciò è causato dal dimensionamento automatico che considera il numero dei processi in esecuzione come capacità corrente, ovvero il doppio del numero appropriato di processi utilizzati nel calcolo del numero dei processi desiderato.
- I processi che utilizzano il tipo di avvio Fargate oppure i tipi di controller di implementazione `CODE_DEPLOY` non supportano la strategia di pianificazione `DAEMON`.
- Quando si crea inizialmente un' CodeDeploy applicazione e un gruppo di distribuzione, è necessario specificare quanto segue:
  - È necessario definire due gruppi target per il load balancer. Un gruppo di destinazione deve essere il gruppo di destinazione iniziale definito per il load balancer quando il servizio Amazon ECS è stato creato. L'unico requisito del secondo gruppo target è che non può essere associato a un load balancer diverso rispetto a quello utilizzato dal servizio.
- Quando crei una CodeDeploy distribuzione per un servizio Amazon ECS, CodeDeploy crea un set di attività sostitutivo (o set di attività verde) nella distribuzione. Se hai aggiunto un listener di test al load balancer, CodeDeploy indirizza il traffico di test verso il set di attività sostitutivo. Questo è il momento in cui è possibile eseguire i test di convalida. Quindi CodeDeploy reindirizza il traffico di produzione dal set di attività originale al set di attività sostitutivo in base alle impostazioni di reindirizzamento del traffico per il gruppo di distribuzione.

Autorizzazioni IAM richieste

Le implementazioni blu/verdi sono rese possibili da una combinazione di Amazon ECS e API. CodeDeploy Gli utenti devono disporre delle autorizzazioni appropriate per questi servizi prima di poter utilizzare le distribuzioni blu/verdi di Amazon ECS in o con gli SDK o. AWS Management Console AWS CLI

Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, Amazon ECS richiede le seguenti autorizzazioni. Queste autorizzazioni sono state aggiunte alla policy IAM `AmazonECS_FullAccess`. Per ulteriori informazioni, consulta AmazonECS_ FullAccess.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:GetApplication",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:ListApplications",
                "codedeploy:ListDeploymentGroups",
                "codedeploy:ListDeployments",
                "codedeploy:StopDeployment",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:ListDeploymentTargets",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetApplicationRevision",
                "codedeploy:RegisterApplicationRevision",
                "codedeploy:BatchGetApplicationRevisions",
                "codedeploy:BatchGetDeploymentGroups",
                "codedeploy:BatchGetDeployments",
                "codedeploy:BatchGetApplications",
                "codedeploy:ListApplicationRevisions",
                "codedeploy:ListDeploymentConfigs",
                "codedeploy:ContinueDeployment",
                "sns:ListTopics",
                "cloudwatch:DescribeAlarms",
                "lambda:ListFunctions"
            ],
            "Resource": ["*"]
        }
    ]
}
```

> **Note**
>
> Oltre alle autorizzazioni Amazon ECS standard necessarie per eseguire processi e servizi, gli utenti hanno bisogno anche delle autorizzazioni `iam:PassRole` per utilizzare i ruoli IAM per le attività.

CodeDeploy necessita delle autorizzazioni per chiamare le API di Amazon ECS, modificare Elastic Load Balancing, richiamare funzioni Lambda CloudWatch e descrivere gli allarmi, oltre alle autorizzazioni per modificare il conteggio desiderato del servizio per tuo conto. Prima di creare un servizio Amazon ECS che usa il tipo di implementazione blu/verde, occorre creare un ruolo IAM (`ecsCodeDeployRole`). Per ulteriori informazioni, consulta Ruolo CodeDeploy IAM di Amazon ECS.

Gli esempi di policy IAM Crea un esempio di servizio Amazon ECS e Esempio di aggiornamento del servizio Amazon ECS mostrano le autorizzazioni necessarie affinché gli utenti possano utilizzare le implementazioni blu/verde di Amazon ECS nella AWS Management Console.

Implementazione di un servizio Amazon ECS utilizzando una distribuzione blu/verde

Scopri come creare un servizio Amazon ECS contenente un'attività Fargate che utilizza il tipo di distribuzione blu/verde con. AWS CLI

> **Note**
>
> È stato aggiunto il supporto per l'esecuzione di un'implementazione blu/verde per AWS CloudFormation. Per ulteriori informazioni, consulta Eseguire le distribuzioni blu/green di Amazon ECS tramite l' CodeDeploy utilizzo AWS CloudFormation nella Guida per l'utente.AWS CloudFormation

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di è installata e configurata. AWS CLI Per ulteriori informazioni sull'installazione o l'aggiornamento di AWS CLI, vedere Installazione di. AWS Command Line Interface
- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.

- AWS L'utente dispone delle autorizzazioni richieste specificate nell'esempio di AmazonECS_FullAccess policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.
- Viene creato il ruolo CodeDeploy IAM di Amazon ECS. Per ulteriori informazioni, consulta Ruolo CodeDeploy IAM di Amazon ECS.

Fase 1: Creazione di un Application Load Balancer

I servizi di Amazon ECS che utilizzano il tipo di implementazione blu/verde richiedono l'utilizzo di un servizio di Application Load Balancer o di un Network Load Balancer. In questa esercitazione viene utilizzato un Application Load Balancer.

Per creare un Application Load Balancer

1. Utilizza il comando create-load-balancer 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. Utilizzare il comando create-target-group 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

Utilizza il comando register-task-definition 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": "tutorial-task-def",
        "networkMode": "awsvpc",
        "containerDefinitions": [
            {
                "name": "sample-app",
                "image": "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: #00FFFF;} </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",
        "executionRoleArn": "arn:aws:iam::aws_account_id:role/ecsTaskExecutionRole"
}
```

Quindi, registra la definizione di attività utilizzando il file `fargate-task.json` creato.

```
aws ecs register-task-definition \
     --cli-input-json file://fargate-task.json \
     --region us-east-1
```

Fase 4: Creazione di un servizio Amazon ECS

Utilizza il comando create-service per creare un servizio.

Crea innanzitutto un file denominato `service-bluegreen.json` con i seguenti contenuti.

```
{
    "cluster": "tutorial-bluegreen-cluster",
    "serviceName": "service-bluegreen",
    "taskDefinition": "tutorial-task-def",
    "loadBalancers": [
        {
            "targetGroupArn":
 "arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4",
            "containerName": "sample-app",
            "containerPort": 80
        }
    ],
    "launchType": "FARGATE",
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "CODE_DEPLOY"
    },
    "platformVersion": "LATEST",
    "networkConfiguration": {
       "awsvpcConfiguration": {
          "assignPublicIp": "ENABLED",
          "securityGroups": [ "sg-abcd1234" ],
          "subnets": [ "subnet-abcd1234", "subnet-abcd5678" ]
       }
    },
    "desiredCount": 1
```

```
}
```

Quindi, crea il servizio utilizzando il file `service-bluegreen.json` creato.

```
aws ecs create-service \
     --cli-input-json file://service-bluegreen.json \
     --region us-east-1
```

L'output include l'ARN del servizio, con il formato seguente:

```
arn:aws:ecs:region:aws_account_id:service/service-bluegreen
```

Ottieni il nome DNS del sistema di bilanciamento del carico utilizzando il comando seguente.

```
 aws elbv2 describe-load-balancers --name bluegreen-alb  --query
 'LoadBalancers[*].DNSName'
```

Immetti il nome DNS nel browser Web e dovresti visualizzare una pagina Web che mostra l'applicazione di esempio con uno sfondo blu.

Fase 5: Creazione delle risorse AWS CodeDeploy

Utilizza i seguenti passaggi per creare CodeDeploy l'applicazione, il gruppo target Application Load Balancer per il gruppo di CodeDeploy distribuzione e il gruppo di CodeDeploy distribuzione.

Per creare risorse CodeDeploy

1. Utilizzate il comando create-application per creare un' CodeDeploy applicazione. Specificare la piattaforma di calcolo `ECS`.

   ```
   aws deploy create-application \
        --application-name tutorial-bluegreen-app \
        --compute-platform ECS \
        --region us-east-1
   ```

   L'output include l'ID applicazione, con il formato seguente:

   ```
   {
       "applicationId": "b8e9c1ef-3048-424e-9174-885d7dc9dc11"
   ```

```
}
```

2. Utilizzate il comando create-target-group per creare un secondo gruppo target di Application Load Balancer, che verrà utilizzato durante la creazione del gruppo di distribuzione. CodeDeploy

```
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 comando create-deployment-group per creare un gruppo di distribuzione. CodeDeploy

Crea innanzitutto un file denominato `tutorial-deployment-group.json` con i seguenti contenuti. Questo esempio utilizza la risorsa che è stata creata. Per la`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 CodeDeploy distribuzione

Prima di creare una CodeDeploy distribuzione, aggiorna la definizione `command` dell'attività `fargate-task.json` come segue per modificare il colore di sfondo dell'app di esempio in verde.

```
{
    ...
        "containerDefinitions": [
            {
                ...
                "command": [
                    "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #097969;} </style> </
head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample App</h1>
 <h2>Congratulations!</h2> <p>Your application is now running on a container in Amazon
 ECS.</p> </div></body></html>' >  /usr/local/apache2/htdocs/index.html && httpd-
foreground\""
                ]
            }
        ],
       ...
}
```

Registra la definizione di attività aggiornata utilizzando il comando seguente.

```
aws ecs register-task-definition \
     --cli-input-json file://fargate-task.json \
     --region us-east-1
```

Ora, utilizzate i seguenti passaggi per creare e caricare un file di specifiche dell'applicazione (AppSpec file) e una CodeDeploy distribuzione.

Per creare e monitorare una CodeDeploy distribuzione

1. Crea e carica un AppSpec file utilizzando i seguenti passaggi.

a. Crea un file denominato `appspec.yaml` con il contenuto del gruppo di CodeDeploy distribuzione. Questo esempio utilizza la definizione di attività aggiornata.

```
version: 0.0
Resources:
  - TargetService:
      Type: AWS::ECS::Service
      Properties:
        TaskDefinition: "arn:aws:ecs:region:aws_account_id:task-
definition/tutorial-task-def:2"
        LoadBalancerInfo:
          ContainerName: "sample-app"
          ContainerPort: 80
        PlatformVersion: "LATEST"
```

b. Usa il comando s3 mb per creare un bucket Amazon S3 per il file. AppSpec

```
aws s3 mb s3://tutorial-bluegreen-bucket
```

c. Usa il comando s3 cp per caricare il AppSpec file nel bucket Amazon S3.

```
aws s3 cp ./appspec.yaml s3://tutorial-bluegreen-bucket/appspec.yaml
```

2. Crea la CodeDeploy distribuzione utilizzando i seguenti passaggi.

a. Crea un file denominato `create-deployment.json` con il contenuto della CodeDeploy distribuzione. Questo esempio utilizza le risorse create in precedenza nel tutorial.

```
{
    "applicationName": "tutorial-bluegreen-app",
    "deploymentGroupName": "tutorial-bluegreen-dg",
    "revision": {
        "revisionType": "S3",
        "s3Location": {
            "bucket": "tutorial-bluegreen-bucket",
            "key": "appspec.yaml",
            "bundleType": "YAML"
        }
    }
}
```

b. Utilizzare il comando create-deployment per creare l'implementazione.

```
aws deploy create-deployment \
     --cli-input-json file://create-deployment.json \
     --region us-east-1
```

L'output include l'ID di distribuzione, con il formato seguente:

```
{
    "deploymentId": "d-RPCR1U3TW"
}
```

3. Utilizzare il comando get-deployment-target per ottenere i dettagli dell'implementazione, specificando il `deploymentId` dall'output precedente.

```
aws deploy get-deployment-target \
--deployment-id "d-IMJU3A8TW" \
--target-id tutorial-bluegreen-cluster:service-bluegreen \
--region us-east-1
```

Inizialmente, lo stato dell'implementazione è `InProgress`. Il traffico viene indirizzato al set di attività originale, che presenta `taskSetLabel` con valore `BLUE`, uno stato `PRIMARY` e un `trafficWeight` di `100.0`. Il set di attività di sostituzione ha `taskSetLabel` con valore `GREEN`, uno stato `ACTIVE` e un `trafficWeight` di `0.0`. Il browser Web in cui hai inserito il nome DNS mostra ancora l'app di esempio con uno sfondo blu.

```
{
"deploymentTarget": {
"deploymentTargetType": "ECSTarget",
"ecsTarget": {
    "deploymentId": "d-RPCR1U3TW",
    "targetId": "tutorial-bluegreen-cluster:service-bluegreen",
    "targetArn": "arn:aws:ecs:region:aws_account_id:service/service-bluegreen",
    "lastUpdatedAt": "2023-08-10T12:07:24.797000-05:00",
    "lifecycleEvents": [
        {
            "lifecycleEventName": "BeforeInstall",
            "startTime": "2023-08-10T12:06:22.493000-05:00",
            "endTime": "2023-08-10T12:06:22.790000-05:00",
            "status": "Succeeded"
        },
        {
```

```
            "lifecycleEventName": "Install",
            "startTime": "2023-08-10T12:06:22.936000-05:00",
            "status": "InProgress"
        },
        {
            "lifecycleEventName": "AfterInstall",
            "status": "Pending"
        },
        {
            "lifecycleEventName": "BeforeAllowTraffic",
            "status": "Pending"
        },
        {
            "lifecycleEventName": "AllowTraffic",
            "status": "Pending"
        },
        {
            "lifecycleEventName": "AfterAllowTraffic",
            "status": "Pending"
        }
    ],
    "status": "InProgress",
    "taskSetsInfo": [
        {
            "identifer": "ecs-svc/9223370493423413672",
            "desiredCount": 1,
            "pendingCount": 0,
            "runningCount": 1,
            "status": "ACTIVE",
            "trafficWeight": 0.0,
            "targetGroup": {
                "name": "bluegreentarget2"
            },
            "taskSetLabel": "Green"
        },
        {
            "identifer": "ecs-svc/9223370493425779968",
            "desiredCount": 1,
            "pendingCount": 0,
            "runningCount": 1,
            "status": "PRIMARY",
            "trafficWeight": 100.0,
            "targetGroup": {
                "name": "bluegreentarget1"
```

```
            },
            "taskSetLabel": "Blue"
        }
    ]
}
}
}
```

Continua a recuperare i dettagli di implementazione utilizzando il comando fino a quando lo stato è `Succeeded`, come mostrato nel seguente output. Il traffico viene ora instradato al set di attività di sostituzione, che presenta uno stato `PRIMARY` e un `trafficWeight` di `100.0`. Aggiorna il browser Web in cui hai inserito il nome DNS del sistema di bilanciamento del carico e ora dovresti visualizzare l'app di esempio con uno sfondo verde.

```
{
"deploymentTarget": {
"deploymentTargetType": "ECSTarget",
"ecsTarget": {
    "deploymentId": "d-RPCR1U3TW",
    "targetId": "tutorial-bluegreen-cluster:service-bluegreen",
    "targetArn": "arn:aws:ecs:region:aws_account_id:service/service-bluegreen",
    "lastUpdatedAt": "2023-08-10T12:07:24.797000-05:00",
    "lifecycleEvents": [
        {
            "lifecycleEventName": "BeforeInstall",
            "startTime": "2023-08-10T12:06:22.493000-05:00",
            "endTime": "2023-08-10T12:06:22.790000-05:00",
            "status": "Succeeded"
        },
        {
            "lifecycleEventName": "Install",
            "startTime": "2023-08-10T12:06:22.936000-05:00",
            "endTime": "2023-08-10T12:08:25.939000-05:00",
            "status": "Succeeded"
        },
        {
            "lifecycleEventName": "AfterInstall",
            "startTime": "2023-08-10T12:08:26.089000-05:00",
            "endTime":  "2023-08-10T12:08:26.403000-05:00",
            "status": "Succeeded"
        },
        {
```

```
            "lifecycleEventName": "BeforeAllowTraffic",
            "startTime": "2023-08-10T12:08:26.926000-05:00",
            "endTime":  "2023-08-10T12:08:27.256000-05:00",
            "status": "Succeeded"
        },
        {
            "lifecycleEventName": "AllowTraffic",
            "startTime": "2023-08-10T12:08:27.416000-05:00",
            "endTime": "2023-08-10T12:08:28.195000-05:00",
            "status": "Succeeded"
        },
        {
            "lifecycleEventName": "AfterAllowTraffic",
            "startTime": "2023-08-10T12:08:28.715000-05:00",
            "endTime":  "2023-08-10T12:08:28.994000-05:00",
            "status": "Succeeded"
        }
    ],
    "status": "Succeeded",
    "taskSetsInfo": [
        {
            "identifer": "ecs-svc/9223370493425779968",
            "desiredCount": 1,
            "pendingCount": 0,
            "runningCount": 1,
            "status": "ACTIVE",
            "trafficWeight": 0.0,
            "targetGroup": {
                "name": "bluegreentarget1"
            },
            "taskSetLabel": "Blue"
        },
        {
            "identifer": "ecs-svc/9223370493423413672",
            "desiredCount": 1,
            "pendingCount": 0,
            "runningCount": 1,
            "status": "PRIMARY",
            "trafficWeight": 100.0,
            "targetGroup": {
                "name": "bluegreentarget2"
            },
            "taskSetLabel": "Green"
        }
```

```
    ]
}
}
}
```

Fase 7: pulire

Una volta terminato questo tutorial, rimuovi le risorse associate per evitare costi aggiuntivi per risorse che non utilizzi.

Rimozione delle risorse del tutorial

1. Utilizzate il comando delete-deployment-group per eliminare il gruppo di distribuzione. CodeDeploy

```
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'applicazione. CodeDeploy

```
aws deploy delete-application \
     --application-name tutorial-bluegreen-app \
     --region us-east-1
```

3. Utilizza il comando delete-service per eliminare il servizio Amazon ECS. L'uso del flag `--force` consente di eliminare un servizio anche se il numero di attività non è stato ridotto a zero.

```
aws ecs delete-service \
     --service arn:aws:ecs:region:aws_account_id:service/service-bluegreen \
     --force \
     --region us-east-1
```

4. Utilizza il comando delete-cluster per eliminare il cluster Amazon ECS.

```
aws ecs delete-cluster \
     --cluster tutorial-bluegreen-cluster \
     --region us-east-1
```

5. Usa il comando s3 rm per eliminare il AppSpec file dal bucket Amazon S3.

```
aws s3 rm s3://tutorial-bluegreen-bucket/appspec.yaml
```

6. Utilizza il comando s3 rb per eliminare il bucket Amazon S3.

```
aws s3 rb s3://tutorial-bluegreen-bucket
```

7. Utilizza il comando delete-load-balancer 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. Utilizza il comando delete-target-group per eliminare i due gruppi di destinazione di Application Load Balancer.

```
aws elbv2 delete-target-group \
     --target-group-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget1/209a844cd01825a4 \
     --region us-east-1
```

```
aws elbv2 delete-target-group \
     --target-group-arn
 arn:aws:elasticloadbalancing:region:aws_account_id:targetgroup/
bluegreentarget2/708d384187a3cfdc \
     --region us-east-1
```

Implementa i servizi Amazon ECS utilizzando un controller di terze parti

Il tipo di implementazione esterna consente di usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implementazione per un servizio Amazon ECS. I dettagli del servizio vengono gestiti tramite le operazioni API del servizio di gestione (`CreateService`, `UpdateService` e `DeleteService`) o della gestione dei set di attività (`CreateTaskSet`, `UpdateTaskSet`, `UpdateServicePrimaryTaskSet` e `DeleteTaskSet`). Ciascuna operazione API gestisce un sottoinsieme di parametri di definizione del servizio.

L'operazione API `UpdateService` aggiorna i parametri del periodo di tolleranza per il conteggio desiderato e il controllo dello stato di un servizio. Se è necessario aggiornare il tipo di avvio, la versione della piattaforma, i dettagli del load balancer, la configurazione di rete o la definizione di attività, occorre creare un nuovo set di attività.

L'operazione API `UpdateTaskSet` aggiorna solo il parametro di dimensionamento per un set di attività.

L'operazione API `UpdateServicePrimaryTaskSet` modifica il set di attività impostato come principale in un servizio. Chiamando l'operazione API `DescribeServices`, vengono restituiti tutti i campi specificati per un set di attività principale. Se viene aggiornato il set di attività principale per un servizio, quando viene definito il nuovo set principale eventuali valori del parametro esistenti sul nuovo ma diversi da quelli impostati sul precedente set di attività in un servizio verranno aggiornati ai nuovi valori. Se non viene definito un set di attività principale per un servizio, i relativi campi nella descrizione del servizio sono nulli.

Considerazioni sull'implementazione esterna

Quando utilizzi il tipo di distribuzione esterna, tieni conto di quanto segue:

- I tipi di load balancer supportati sono un Application Load Balancer o un Network Load Balancer.
- Il tipo di avvio Fargate oppure i tipi di controller di implementazione `EXTERNAL` non supportano la strategia di pianificazione `DAEMON`.

Flusso di lavoro dell'implementazione esterna

Di seguito è riportato il flusso di lavoro di base per la gestione di una distribuzione esterna su Amazon ECS.

Come gestire un servizio Amazon ECS tramite un controller di implementazione esterno

1. Crea un servizio Amazon ECS. L'unico parametro obbligatorio è il nome del servizio. Durante la creazione di un servizio tramite un controller di distribuzione esterno, è possibile specificare i parametri indicati di seguito. Tutti gli altri, invece, vengono specificati al momento della creazione di un set di attività nel servizio.

   `serviceName`

   Tipo: stringa

Campo obbligatorio: sì

Il nome del servizio. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. I nomi dei servizi devono essere univoci all'interno di un cluster, ma puoi avere servizi dai nomi simili in più cluster all'interno di una Regione o in più Regioni.

`desiredCount`

Il numero di istanze della definizione di attività per il set specificato, da posizionare e mantenere in esecuzione nel servizio.

`deploymentConfiguration`

I parametri di distribuzione opzionali che determinano quante attività vengono eseguite durante un'implementazione e l'ordine di arresto e di avvio delle attività.

`tags`

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati applicati al servizio per aiutarti a catalogarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Quando un servizio viene eliminato, vengono eliminati anche i tag. Al servizio è possibile applicare un massimo di 50 tag. Per ulteriori informazioni, consulta Taggare le risorse Amazon ECS.

`key`

- Tipo: stringa

  Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

  Campo obbligatorio: no

  Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

`value`

- Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

`enableECSManagedTags`

Specifica se usare i tag gestiti di Amazon ECS per i processi all'interno del servizio. Per ulteriori informazioni, consulta Utilizza i tag per la fatturazione.

`propagateTags`

-Tipo: stringa

Valori validi: `TASK_DEFINITION` | `SERVICE`

Campo obbligatorio: no

Specifica se copiare i tag dalla definizione di attività o dal servizio nelle attività del servizio. Se non viene specificato alcun valore, i tag non vengono copiati. I tag possono essere copiati solo nelle attività all'interno del servizio durante la creazione del servizio. Per aggiungere tag a un processo dopo la creazione di un servizio o di un processo, utilizza l'operazione API `TagResource`.

`schedulingStrategy`

La strategia di pianificazione da utilizzare. I servizi che utilizzano un controller di distribuzione esterno supportano solo la strategia di pianificazione `REPLICA`.

`placementConstraints`

Serie di oggetti vincolo di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di 10 vincoli per attività (questo limite include i vincoli nella definizione di attività e quelli specificati in fase di runtime). Se utilizzi il tipo di avvio Fargate, i vincoli di posizionamento dei processi non sono supportati.

`placementStrategy`

Gli oggetti strategia di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di quattro regole di strategia per ogni servizio.

Di seguito è illustrato un esempio di definizione per la creazione di un servizio che utilizza un controller di distribuzione esterno.

```
{
    "cluster": "",
    "serviceName": "",
    "desiredCount": 0,
    "role": "",
    "deploymentConfiguration": {
        "maximumPercent": 0,
        "minimumHealthyPercent": 0
    },
    "placementConstraints": [
        {
            "type": "distinctInstance",
            "expression": ""
        }
    ],
    "placementStrategy": [
        {
            "type": "binpack",
            "field": ""
        }
    ],
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "EXTERNAL"
    },
    "tags": [
        {
            "key": "",
            "value": ""
        }
    ],
    "enableECSManagedTags": true,
    "propagateTags": "TASK_DEFINITION"
}
```

2. Creare un set di attività iniziali. Il set di attività contiene i seguenti i dettagli sul servizio:

`taskDefinition`

La definizione delle attività nel set da utilizzare.

`launchType`

- Tipo: stringa

Valori validi: `EC2` | `FARGATE` | `EXTERNAL`

Campo obbligatorio: no

Il tipo di avvio con cui eseguire il servizio. Se non viene specificato un tipo di avvio, per impostazione predefinita viene utilizzato il `capacityProviderStrategy` di default. Per ulteriori informazioni, consulta Tipi di avvio di Amazon ECS.

Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

`platformVersion`

- Tipo: stringa

Campo obbligatorio: no

La versione della piattaforma su cui sono in esecuzione le attività nel servizio. Viene specificata una versione della piattaforma solo per le attività con tipo di avvio Fargate. Se non è specificata, la versione più recente (`LATEST`) viene utilizzata di default.

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Quando specifichi la versione della piattaforma `LATEST` durante l'esecuzione di un'attività o la creazione di un servizio, ottieni la versione di piattaforma più aggiornata disponibile per le tue attività. Quando incrementi il servizio, tali attività riceveranno la versione della piattaforma specificata nell'implementazione corrente del servizio. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.

> **Note**
>
> Le versioni della piattaforma non sono specificate per i processi che utilizzano il tipo di avvio EC2.

`loadBalancers`

Un oggetto che rappresenta il load balancer da utilizzare con il tuo servizio. Quando si utilizza un controller di implementazione esterno, sono supportati solo Application Load Balancer e il Network Load Balancer. Se si utilizza un Application Load Balancer, è consentito un solo gruppo di destinazione di Application Load Balancer per ogni set di processi.

Il frammento di codice seguente mostra un oggetto `loadBalancer` di esempio da utilizzare.

```
"loadBalancers": [
        {
            "targetGroupArn": "",
            "containerName": "",
            "containerPort": 0
        }
]
```

> **Note**
>
> Quando si specifica un oggetto `loadBalancer`, è necessario specificare `targetGroupArn` e omettere i parametri `loadBalancerName`.

`networkConfiguration`

La configurazione di rete per il servizio. Questo parametro è obbligatorio per le definizioni di attività che utilizzano la modalità di rete `awsvpc` per ricevere la propria interfaccia di rete elastica e non è supportato per altre modalità. Per ulteriori informazioni sulla rete per il tipo di lancio Fargate, vedere. Opzioni di task networking di Amazon ECS per il tipo di lancio Fargate

`serviceRegistries`

I dettagli dei registri del servizio di individuazione da assegnare a questo servizio. Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.

`scale`

Percentuale a virgola mobile del numero desiderato di attività da posizionare e mantenere in esecuzione nel seti di attività. Il valore è specificato come percentuale totale del conteggio desiderato (`desiredCount`) di un servizio. Come valori sono accettati i numeri da 0 a 100.

Di seguito è riportato un esempio JSON per la creazione di un set di attività per un controller di distribuzione esterno.

```
{
    "service": "",
    "cluster": "",
    "externalId": "",
    "taskDefinition": "",
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": [
                ""
            ],
            "securityGroups": [
                ""
            ],
            "assignPublicIp": "DISABLED"
        }
    },
    "loadBalancers": [
        {
            "targetGroupArn": "",
            "containerName": "",
            "containerPort": 0
        }
    ],
    "serviceRegistries": [
        {
            "registryArn": "",
            "port": 0,
```

```
            "containerName": "",
            "containerPort": 0
        }
    ],
    "launchType": "EC2",
    "capacityProviderStrategy": [
        {
            "capacityProvider": "",
            "weight": 0,
            "base": 0
        }
    ],
    "platformVersion": "",
    "scale": {
        "value": null,
        "unit": "PERCENT"
    },
    "clientToken": ""
}
```

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 `100`e, una volta pronti a spostare tutto il traffico di produzione nel `tasksetB`, è possibile aggiornare `scale` per `tasksetA` su `0` per ridurlo.

Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS

Il servizio può essere configurato opzionalmente per utilizzare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività del servizio.

> **Note**
>
> Quando si utilizzano set di processi, tutti i processi del set devono essere configurati per utilizzare Elastic Load Balancing o non utilizzare Elastic Load Balancing.

I servizi Amazon ECS ospitati su AWS Fargate supportano Application Load Balancer, Network Load Balancer e Gateway Load Balancer. Utilizza la tabella seguente per scoprire quale tipo di bilanciamento del carico utilizzare.

Tipo di Load Balancer	Utilizzare in questi casi	
Application Load Balancer	Indirizza il traffico HTTP/HTTPS (o livello 7). Gli Application Load Balancer offrono diverse funzionalità che li rendono particolarmente 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. • Questi sono supportati dai processi ospitati su istanze Fargate e Amazon 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	

Tipo di Load Balancer	Utilizzare in questi casi
	provenienti dallo stesso servizio). • Gli Application Load Balancer supportano il routing basato su percorsi e le regole di priorità (in modo che più servizi possano utilizzare la stessa porta listener su un singolo Application Load Balancer).
Network Load Balancer	Indirizza il traffico TCP o UDP (o livello 4).
Gateway Load Balancer	Indirizza il traffico TCP o UDP (o livello 4). Utilizza dispositivi virtuali, come firewall, sistemi di rilevamento e prevenzione delle intrusioni e sistemi di ispezione approfondita dei pacchetti.

Ti consigliamo di utilizzare Application Load Balancers per i tuoi servizi Amazon ECS in modo da poter sfruttare queste ultime funzionalità, a meno che il tuo servizio non richieda una funzionalità disponibile solo con Network Load Balancer o Gateway Load Balancer. Per ulteriori informazioni sull'Elastic Load Balancing e le differenze tra questi tipi di load balancer, consulta la Guida per l'utente di Elastic Load Balancing.

Con il load balancer paghi solo in base all'uso effettivo. Per ulteriori informazioni, consulta Prezzi di Elastic Load Balancing.

Ottimizza i parametri di controllo dello stato del load balancer per Amazon ECS

I sistemi di bilanciamento del carico indirizzano le richieste solo verso gli obiettivi integri nelle zone di disponibilità del sistema di bilanciamento del carico. Ogni destinazione è registrata in un gruppo target. Il load balancer verifica lo stato di salute di ciascun target, utilizzando le impostazioni di controllo dello stato del gruppo target. Dopo aver registrato il bersaglio, deve superare un controllo dello stato di salute per essere considerato integro. Amazon ECS monitora il sistema di bilanciamento del carico. Il sistema di bilanciamento del carico invia periodicamente controlli di integrità al container Amazon ECS. L'agente Amazon ECS monitora e attende che il sistema di bilanciamento del carico fornisca un rapporto sullo stato del contenitore. Lo fa prima di considerare il container in buono stato.

Due parametri di controllo dello stato di Elastic Load Balancing influiscono sulla velocità di implementazione:

- Intervallo di controllo dello stato di salute: determina il periodo di tempo approssimativo, in secondi, tra i controlli sanitari di un singolo contenitore. Per impostazione predefinita, il sistema di bilanciamento del carico effettua i controlli ogni 30 secondi.

  Questo parametro è denominato:

  - `HealthCheckIntervalSeconds`nell'API Elastic Load Balancing
  - Intervallo sulla console Amazon EC2
- Numero di soglie di integrità: determina il numero di controlli sanitari consecutivi superati prima di considerare integro un contenitore non integro. Per impostazione predefinita, il sistema di bilanciamento del carico richiede il superamento di cinque controlli di integrità prima di segnalare che il contenitore di destinazione è integro.

  Questo parametro è denominato:

  - `HealthyThresholdCount`nell'API Elastic Load Balancing
  - Soglia integra sulla console Amazon EC2

Con le impostazioni predefinite, il tempo totale per determinare lo stato di salute di un contenitore è di due minuti e 30 secondi (`30 seconds * 5 = 150 seconds`).

Puoi accelerare il processo di controllo dello stato se il servizio si avvia e si stabilizza in meno di 10 secondi. Per accelerare il processo, riduci il numero di controlli sanitari e l'intervallo tra i controlli.

- `HealthCheckIntervalSeconds`(nome API Elastic Load Balancing) o Interval (nome della console Amazon EC2): 5

- `HealthyThresholdCount`(nome API Elastic Load Balancing) o Healthy threshold (nome della console Amazon EC2): 2

Con questa impostazione, il processo di controllo dello stato richiede 10 secondi rispetto all'impostazione predefinita di due minuti e 30 secondi.

Per ulteriori informazioni sui parametri del controllo dello stato di Elastic Load Balancing, consulta Controlli dello stato per i gruppi target nella Elastic Load Balancing User Guide.

Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS

Per consentire l'ottimizzazione, i client mantengono una connessione keep alive al servizio container. Ciò consente alle richieste successive di quel client di riutilizzare la connessione esistente. Quando si desidera interrompere il traffico verso un container, si invia una notifica al sistema di bilanciamento del carico. Il load balancer controlla periodicamente se il client ha chiuso la connessione keep alive. L'agente Amazon ECS monitora il sistema di bilanciamento del carico e attende che il sistema di bilanciamento del carico segnali che la connessione keep alive è chiusa (la destinazione è in uno stato). `UNUSED`

La quantità di tempo che il load balancer attende per spostare la destinazione verso lo stato corrisponde al ritardo di annullamento della registrazione. `UNUSED` È possibile configurare il seguente parametro di bilanciamento del carico per velocizzare le distribuzioni.

- `deregistration_delay.timeout_seconds`: 300 (impostazione predefinita)

Se disponi di un servizio con un tempo di risposta inferiore a un secondo, imposta il parametro sul valore seguente per fare in modo che il load balancer attenda solo 5 secondi prima di interrompere la connessione tra il client e il servizio di back-end:

- `deregistration_delay.timeout_seconds`: 5

> **Note**
>
> Non impostate il valore su 5 secondi quando disponete di un servizio con richieste di lunga durata, come caricamenti lenti di file o connessioni di streaming.

Reattività SIGTERM

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

La quantità di tempo che Amazon ECS attende per inviare il messaggio SIGKILL è determinata dalla seguente opzione dell'agente Amazon ECS:

- `ECS_CONTAINER_STOP_TIMEOUT`: 30 (impostazione predefinita)

  Per ulteriori informazioni sul parametro Container Agent, consulta Amazon ECS Container Agent on GitHub.

Per accelerare il periodo di attesa, imposta il parametro dell'agente Amazon ECS sul seguente valore:

> **Note**
>
> Se l'applicazione impiega più di 1 secondo, moltiplica il valore per 2 e usa quel numero come valore.

- `ECS_CONTAINER_STOP_TIMEOUT` 2:

In questo caso, Amazon ECS attende 2 secondi che il contenitore si spenga, quindi Amazon ECS invia un messaggio SIGKILL quando l'applicazione non si ferma.

Puoi anche modificare il codice dell'applicazione per intercettare il segnale SIGTERM e reagire ad esso. Di seguito è riportato un esempio in JavaScript:

```
process.on('SIGTERM', function() {
  server.close();
})
```

Questo codice fa sì che il server HTTP smetta di ascoltare eventuali nuove richieste, completi di rispondere a tutte le richieste in corso e quindi il processo Node.js termini. Questo perché il relativo ciclo di eventi non ha più nulla da fare. Detto questo, se il processo impiega solo 500 ms

per completare le richieste in corso, termina in anticipo senza dover attendere il timeout di arresto e ricevere un SIGKILL.

Usa un Application Load Balancer per Amazon ECS

Un Application Load Balancer seleziona il routing a livello di applicazione (HTTP/HTTPS), supporta il routing in base al percorso e può instradare le richieste a una o più porte su ogni istanza di container nel cluster. Gli Application Load Balancer supportano la mappatura dinamica delle porte dell'host. Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). All'avvio dell'attività, il contenitore NGINX viene registrato con Application Load Balancer come combinazione di ID di istanza e porta e il traffico viene distribuito all'ID dell'istanza e alla porta corrispondenti a quel contenitore. Con la mappatura dinamica possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta la Guida per l'utente per Application Load Balancer.

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le distribuzioni, consulta:

- Ottimizza i parametri di controllo dello stato del load balancer per Amazon ECS
- Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS

Considera quanto segue quando utilizzi Application Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.
- Il gruppo target deve avere il tipo di indirizzo IP impostato su IPv4.
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2.
- Se il servizio richiede l'accesso a più porte con bilanciamento del carico, come la porta 80 e la porta 443 per un servizio HTTP/HTTPS, puoi configurare due listener. Un listener è responsabile di

HTTPS che inoltra la richiesta al servizio e un altro listener responsabile del reindirizzamento delle richieste HTTP alla porta HTTPS appropriata. Per ulteriori informazioni, consulta Creazione di un listener per Application Load Balancer nella Guida per l'utente per Application Load Balancer.

- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. È possibile utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del bilanciamento del carico solo per il controller di distribuzione ECS mobile, non blu/verde o esterno. AWS CloudFormation AWS CodeDeploy Quando aggiungi, aggiorni o rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento UpdateServiceall'API di Amazon Elastic Container Service.
- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS.
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- Utilizza un gruppo target unico per ogni servizio.

  L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Application Load Balancer, consulta Creare un Application Load Balancer in Application Load Balancers

Usa un Network Load Balancer per Amazon ECS

Un Network Load Balancer seleziona il routing a livello di trasporto (TCP/SSL). È in grado di gestire milioni di richieste al secondo. Dopo aver ricevuto una connessione, il load balancer seleziona una destinazione dal gruppo di destinazione per la regola predefinita, utilizzando un algoritmo di routing per l'hash del flusso. Tenta quindi di aprire una connessione TCP per la destinazione selezionata sulla porta specificata nella configurazione del listener, Inoltra la richiesta senza modificare le

intestazioni. I Network Load Balancer supportano la mappatura dinamica delle porte dell'host. Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). Una volta avviata l'attività, il container NGINX viene registrato con il Network Load Balancer come combinazione di ID istanza e porta, mentre il traffico viene instradato all'ID istanza e alla porta corrispondenti a tale container. Con la mappatura dinamica possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta la Guida per l'utente dei Network Load Balancer.

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le implementazioni, consulta:

- Ottimizza i parametri di controllo dello stato del load balancer per Amazon ECS
- Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS

Considera quanto segue quando utilizzi Network Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.
- Non è possibile associare più di cinque gruppi target a un servizio.
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2.
- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. È possibile utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del bilanciamento del carico solo per il controller di distribuzione `ECS` mobile, non 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 UpdateServiceall'API di Amazon Elastic Container Service.

- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.
- Quando si utilizza un Gateway Load Balancer configurato con indirizzi IP come destinazioni e Client IP Preservation disattivata, le richieste vengono viste come provenienti dall'indirizzo IP privato del Gateway Load Balancer. Ciò significa che i servizi alla base di un Gateway Load Balancer sono effettivamente aperti al mondo non appena si autorizzano le richieste in entrata e i controlli di integrità nel gruppo di sicurezza target.
- Per le attività di Fargate, è necessario utilizzare la versione della piattaforma `1.4.0` (Linux) o `1.0.0` (Windows).
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS.
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- La conservazione dell'indirizzo IP del client Network Load Balancer è compatibile con i target Fargate.
- Utilizzate un gruppo target univoco per ogni servizio.

  L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Network Load Balancer, consulta Creare un Network Load Balancer in Network Load Balancer

> **Important**
>
> Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, necessaria per il tipo di avvio Fargate, è necessario scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2.

Non è possibile registrare le istanze in base all'ID istanza per i tipi di istanza seguenti: C1, CC1, CC2, CG1, CG2, CR1, G1, G2, HI1, HS1, M1, M2, M3 e T1. È possibile registrare le istanze di questi tipi in base all'indirizzo IP.

Usa un Gateway Load Balancer per Amazon ECS

Un Gateway Load Balancer funziona al terzo livello del modello Open Systems Interconnection (OSI), il livello di rete. È in ascolto per tutti i pacchetti IP attraverso tutte le porte e inoltra il traffico al gruppo di destinazione specificato nella regola del listener. Mantiene la viscosità dei flussi verso un'appliance di destinazione specifica utilizzando 5 tuple (per flussi TCP/UDP) o 3 tuple (per flussi non TCP/UDP). Ad esempio, se la tua definizione del container del processo specifica la porta 80 per una porta di container NGINX e la porta 0 per la porta dell'host, quest'ultima viene selezionata in modo dinamico nell'intervallo di porte temporaneo dell'istanza di container (ad esempio da 32768 a 61000 nell'AMI ottimizzata per Amazon ECS più recente). All'avvio dell'attività, il contenitore NGINX viene registrato con il Gateway Load Balancer come combinazione di ID di istanza e porta e il traffico viene distribuito all'ID dell'istanza e alla porta corrispondenti a quel contenitore. Con la mappatura dinamica possono essere presenti più attività di un unico servizio sulla stessa istanza di container. Per ulteriori informazioni, consulta Cos'è un Gateway Load Balancer in Gateway Load Balancers.

Per informazioni sulle migliori pratiche per impostare i parametri per velocizzare le implementazioni, consulta:

- Ottimizza i parametri di controllo dello stato del load balancer per Amazon ECS
- Ottimizza i parametri di drenaggio della connessione del sistema di bilanciamento del carico per Amazon ECS

Considera quanto segue quando utilizzi Gateway Load Balancer con Amazon ECS:

- Amazon ECS richiede il ruolo IAM collegato al servizio, che fornisce le autorizzazioni necessarie per eseguire e annullare la registrazione delle destinazioni nel tuo load balancer al momento dell'avvio e dell'arresto dei processi. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.
- Per i servizi con attività che utilizzano la modalità di rete `awsvpc`, quando crei un gruppo target per il servizio, devi scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2.

- La configurazione della sottorete del load balancer deve includere tutte le zone di disponibilità in cui risiedono le tue istanze di container.
- Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. Puoi utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del load balancer solo per il controller di distribuzione ECS mobile, non 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 UpdateServiceall'API di Amazon Elastic Container Service.
- Se un'attività di servizio non soddisfa i criteri di controllo dello stato del sistema di bilanciamento del carico, l'attività viene interrotta e riavviata. Questo processo continua finché il servizio raggiunge il numero desiderato di attività in esecuzione.
- Quando si utilizza un Gateway Load Balancer configurato con indirizzi IP come destinazioni, le richieste vengono viste come provenienti dall'indirizzo IP privato del Gateway Load Balancer. Ciò significa che i servizi alla base di un Gateway Load Balancer sono effettivamente aperti al mondo non appena si autorizzano le richieste in entrata e i controlli di integrità nel gruppo di sicurezza target.
- Per le attività di Fargate, è necessario utilizzare la versione della piattaforma `1.4.0` (Linux) o `1.0.0` (Windows).
- Se riscontri problemi con i servizi abilitati per il load balancer, consulta Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS.
- Le attività e il sistema di bilanciamento del carico devono trovarsi nello stesso VPC.
- Utilizza un gruppo target unico per ogni servizio.

  L'utilizzo dello stesso gruppo target per più servizi potrebbe causare problemi durante le distribuzioni dei servizi.

Per informazioni su come creare un Gateway Load Balancer, vedere Create a Gateway Load Balancer in Gateway Load Balancer

> **Important**
>
> Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, necessaria per il tipo di avvio Fargate, è necessario scegliere `ip` come tipo di destinazione e non `instance`. Il motivo è che i processi che usano la modalità di rete `awsvpc` sono associati a un'interfaccia di rete elastica e non a un'istanza Amazon EC2.
> Non è possibile registrare le istanze in base all'ID istanza per i tipi di istanza seguenti: C1, CC1, CC2, CG1, CG2, CR1, G1, G2, HI1, HS1, M1, M2, M3 e T1. È possibile registrare le istanze di questi tipi in base all'indirizzo IP.

Registrazione di più gruppi target con un servizio Amazon ECS

Il servizio Amazon ECS è in grado di servire traffico proveniente da più load balancer ed esporre più porte con carico bilanciato quando si specificano più gruppi di destinazione in una definizione del servizio.

Per creare un servizio che specifichi più gruppi target, devi creare il servizio utilizzando l'API Amazon ECS, l'SDK o un AWS CLI modello. AWS CloudFormation Dopo aver creato il servizio, è possibile visualizzare il servizio e i gruppi di destinazione registrati in esso con la AWS Management Console. Devi utilizzare `UpdateService` per modificare la configurazione del load balancer di un servizio esistente.

Più gruppi di destinazione possono essere specificati in una definizione del servizio utilizzando il formato seguente. Per la sintassi completa di una definizione di servizio, vedere Modello di definizione del servizio.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"container_name",
      "containerPort":container_port
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"container_name",
```

```
      "containerPort":container_port
   }
]
```

Considerazioni

Considera quanto segue durante la specifica di più gruppi di destinazione in una definizione del servizio:

- Per i servizi che utilizzano Application Load Balancer o un Network Load Balancer (load balancer di rete), non è possibile collegare più di cinque gruppi di destinazione a un servizio.
- La specifica di più gruppi di destinazione in una definizione di servizio è supportata solo nelle seguenti condizioni:
  - Il servizio deve utilizzare un sistema Application Load Balancer o un Network Load Balancer.
  - Il servizio deve utilizzare il tipo di controller di implementazione (`ECS`) con aggiornamento in sequenza.
- La specifica di più gruppi di destinazione è supportata per servizi contenenti processi che utilizzano entrambi i tipi di avvio Fargate e EC2.
- Durante la creazione di un servizio che specifica più gruppi di destinazione, è necessario creare il ruolo collegato ai servizi Amazon ECS. Il ruolo viene creato omettendo il parametro `role` nelle richieste API o la proprietà `Role` in AWS CloudFormation. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.

Definizioni del servizio di esempio

Di seguito sono riportati alcuni casi d'uso per la specifica di più gruppi di destinazione in una definizione del servizio. Per la sintassi completa di una definizione di servizio, vedere Modello di definizione del servizio.

Disporre di bilanciatori di carico separati per il traffico interno ed esterno

Nel seguente caso d'uso, un servizio utilizza due load balancer separati, uno per il traffico interno e un secondo per il traffico della connessione Internet, per lo stesso container e porta.

```
"loadBalancers":[
   //Internal ELB
   {
```

```
 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"nginx",
      "containerPort":8080
   },
   //Internet-facing ELB
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"nginx",
      "containerPort":8080
   }
]
```

Esposizione di più porte dallo stesso contenitore

Nel seguente caso d'uso, un servizio utilizza un load balancer, ma espone più porte dallo stesso container. Ad esempio, un container Jenkins potrebbe esporre la porta 8080 per l'interfaccia Web Jenkins e la porta 50000 per l'API.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"jenkins",
      "containerPort":8080
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"jenkins",
      "containerPort":50000
   }
]
```

Esposizione delle porte di più container

Nel seguente caso d'uso, un servizio utilizza un load balancer e due gruppi di destinazione per esporre porte da container separati.

```
"loadBalancers":[
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_1/1234567890123456",
      "containerName":"webserver",
      "containerPort":80
   },
   {

 "targetGroupArn":"arn:aws:elasticloadbalancing:region:123456789012:targetgroup/
target_group_name_2/6543210987654321",
      "containerName":"database",
      "containerPort":3306
   }
]
```

Ridimensiona automaticamente il tuo servizio Amazon ECS

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, consulta la Guida per l'utente di Application Auto Scaling.

Amazon ECS pubblica CloudWatch parametri con l'utilizzo medio di CPU e memoria del tuo servizio. Per ulteriori informazioni, consulta Metriche di utilizzo del servizio Amazon ECS. Puoi utilizzare queste e altre CloudWatch metriche per scalare il tuo servizio (aggiungere altre attività) per far fronte a una domanda elevata nelle ore di punta e scalare il servizio (eseguire meno attività) per ridurre i costi nei periodi di basso utilizzo.

Amazon ECS Service Auto Scaling supporta i seguenti tipi di scalabilità automatica:

- Scala il tuo servizio Amazon ECS utilizzando un valore metrico target: aumenta o diminuisci 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.
- Scala il tuo servizio Amazon ECS utilizzando incrementi predefiniti basati sugli allarmi CloudWatch : aumenta o diminuisci 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.

- Scala il tuo servizio Amazon ECS utilizzando una pianificazione—Aumentare o diminuire il numero di attività eseguite dal servizio in base alla data e all'ora.

Considerazioni

Quando usi le policy di dimensionamento, tieni in considerazione quanto segue:

- Amazon ECS invia i parametri a intervalli di 1 minuto a. CloudWatch Le metriche non sono disponibili finché i cluster e i servizi non le inviano a e non è possibile creare CloudWatch CloudWatch allarmi per metriche che non esistono.
- Le policy di dimensionamento supportano un periodo di attesa. Questo è il numero di secondi da attendere per rendere effettiva un'attività di dimensionamento precedente.
  - Per gli eventi di dimensionamento orizzontale, l'intenzione è di aumentare continuamente (ma non eccessivamente). Dopo che la scalabilità automatica dei servizi ha correttamente eseguito il dimensionamento orizzontale tramite una policy di dimensionamento, inizia a calcolare il periodo di attesa. La policy di dimensionamento non aumenterà di nuovo la capacità desiderata, a meno che non venga attivato un aumento orizzontale o che venga raggiunto il tempo di raffreddamento. Mentre è attivo il periodo di attesa di incremento, la capacità aggiunta dall'attività di incremento iniziale viene calcolata come parte della capacità desiderata per il successivo evento di incremento.
  - Per gli eventi di riduzione orizzontale, l'intenzione è di ridurre orizzontalmente in modo conservativo per proteggere la disponibilità dell'applicazione, in modo che le attività di riduzione siano bloccate fino alla scadenza del tempo di raffreddamento. Tuttavia, se un altro allarme attiva un'attività di aumento orizzontale durante il tempo di raffreddamento di riduzione orizzontale, Service Auto Scaling esegue immediatamente la riduzione orizzontale della destinazione. In questo caso, il periodo di attesa di riduzione si interrompe e non viene completato.
- Lo scheduler di servizi rispetta il numero desiderato in qualsiasi momento, ma fino a quando avrai policy di dimensionamento e allarmi su un servizio attivi, Service Auto Scaling potrebbe modificare un numero desiderato da te impostato manualmente.
- Se il numero desiderato di un servizio è impostato al di sotto del valore minimo di capacità e un allarme avvia un'attività di scalabilità orizzontale, Service Auto Scaling ridimensiona il conteggio desiderato fino al valore di capacità minimo e quindi continua a scalare in base alle esigenze, in base alla politica di scalabilità associata all'allarme. Tuttavia, un'attività di dimensionamento in riduzione non modifica il numero desiderato, poiché esso è già inferiore al valore di capacità minimo.

- Se il numero desiderato di un servizio è impostato al di sopra del valore di capacità massima e un allarme avvia una scalabilità dell'attività, Service Auto Scaling ridimensiona il conteggio desiderato fino al valore di capacità massima e quindi continua a scalare secondo necessità, in base alla politica di scalabilità associata all'allarme. Tuttavia, un'attività di aumento non modifica il numero desiderato, poiché esso è già superiore al valore di capacità massimo.
- Durante le attività di dimensionamento, il numero reale delle attività in esecuzione in un servizio è il valore che Service Auto Scaling utilizza come punto di partenza, in contrapposizione al numero desiderato. Questo è ciò che dovrebbe essere la capacità di elaborazione. In questo modo si impedisce un dimensionamento eccessivo (fuori controllo) che non può essere soddisfatto, ad esempio, se non ci sono sufficienti risorse di istanze di container per inserire le attività aggiuntive. Se la capacità dell'istanza di container è disponibile in un secondo momento, l'attività di dimensionamento in attesa potrebbe andare a buon fine e, quindi, le attività di dimensionamento potrebbero continuare dopo il periodo di attesa.
- Se desideri che il numero di attività venga dimensionato a zero quando non è necessario eseguire alcuna operazione, imposta una capacità minima pari a 0. Con le policy di dimensionamento con monitoraggio degli obiettivi, quando la capacità effettiva è 0 e il parametro indica che esiste una domanda di carico di lavoro, Service Auto Scaling attende l'invio di un punto dati prima del dimensionamento orizzontale. In questo caso, viene dimensionato in base alla quantità minima possibile come punto di partenza e quindi riprende il dimensionamento in base al numero effettivo di processi in esecuzione.
- Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta Scalabilità automatica e implementazioni dei servizi.
- Sono disponibili diverse opzioni di Application Auto Scaling per le attività di Amazon ECS. Il monitoraggio degli obiettivi è la modalità più semplice da utilizzare, in quanto è necessario soltanto impostare un valore target per un parametro, ad esempio l'utilizzo medio della CPU. L'autoscaler gestisce automaticamente il numero di attività necessarie per raggiungere tale valore. Il dimensionamento per fasi consente di reagire più rapidamente alle variazioni della domanda, poiché si definiscono le soglie specifiche per i parametri di dimensionamento e il numero di attività da aggiungere o rimuovere quando le soglie vengono superate. In particolare, consente di reagire molto rapidamente alle variazioni della domanda riducendo al minimo il tempo di superamento di una soglia di allarme.

Ottimizza la scalabilità automatica del servizio Amazon ECS

Un servizio Amazon ECS è una raccolta gestita di attività. Ogni servizio ha una definizione di attività associata, un numero di attività desiderato e una strategia di posizionamento opzionale. La scalabilità automatica del servizio Amazon ECS viene implementata tramite il servizio Application Auto Scaling. Application Auto Scaling utilizza le CloudWatch metriche come fonte per le metriche di scalabilità. Utilizza inoltre gli CloudWatch allarmi per impostare soglie su quando ampliare o disattivare il servizio. Puoi fornire le soglie per la scalabilità impostando un obiettivo metrico, denominato ridimensionamento del tracciamento degli obiettivi, o specificando delle soglie, denominate scalabilità dei passaggi. Una volta configurato, Application Auto Scaling calcola continuamente il numero di attività desiderato appropriato per il servizio. Inoltre, notifica ad Amazon ECS quando il numero di attività desiderato deve cambiare, ridimensionandolo orizzontalmente o ridimensionandolo.

Per utilizzare il service auto scaling in modo efficace, è necessario scegliere una metrica di scalabilità appropriata.

Un'applicazione deve essere scalata orizzontalmente se si prevede che la domanda sia superiore alla capacità attuale. Al contrario, un'applicazione può essere scalata per ridurre i costi quando le risorse superano la domanda.

Identifica una metrica

Per una scalabilità efficace, è fondamentale identificare una metrica che indichi l'utilizzo o la saturazione. Questa metrica deve presentare le seguenti proprietà per essere utile per la scalabilità.

- La metrica deve essere correlata alla domanda. Quando le risorse vengono mantenute stabili, ma la domanda cambia, anche il valore della metrica deve cambiare. La metrica dovrebbe aumentare o diminuire quando la domanda aumenta o diminuisce.
- Il valore della metrica deve essere scalato in proporzione alla capacità. Quando la domanda rimane costante, l'aggiunta di più risorse deve comportare una modifica proporzionale del valore della metrica. Pertanto, il raddoppio del numero di attività dovrebbe far diminuire la metrica del 50%.

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

Per avviare il processo di test di carico, crea dashboard per le metriche di utilizzo dell'applicazione. Queste metriche includono l'utilizzo della CPU, l'utilizzo della memoria, le operazioni di I/O, la

profondità della coda di I/O e il throughput di rete. Puoi raccogliere queste metriche con un servizio come Container Insights. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights. Durante questo processo, assicurati di raccogliere e tracciare le metriche relative ai tempi di risposta dell'applicazione o ai tassi di completamento del lavoro.

Inizia con una piccola richiesta o una percentuale di inserimento lavorativo ridotta. Mantieni questa frequenza costante per diversi minuti per consentire all'applicazione di riscaldarsi. Quindi, aumenta lentamente la velocità e mantienila costante per alcuni minuti. Ripeti questo ciclo, aumentando la frequenza ogni volta fino a quando i tempi di risposta o completamento dell'applicazione non sono troppo lenti per soddisfare gli obiettivi di livello di servizio (SLO).

Durante il test di carico, esamina ciascuna metrica di utilizzo. Le metriche che aumentano insieme al carico sono le migliori candidate a fungere da migliori metriche di utilizzo.

Successivamente, identifica la risorsa che raggiunge la saturazione. Allo stesso tempo, esamina anche le metriche di utilizzo per vedere quale si appiattisce prima ad un livello elevato o raggiunge un picco e poi blocca prima l'applicazione. Ad esempio, se l'utilizzo della CPU aumenta dallo 0% al 70-80% man mano che si aggiunge carico e poi rimane a quel livello dopo aver aggiunto ancora più carico, si può affermare con certezza che la CPU è satura. A seconda dell'architettura della CPU, potrebbe non raggiungere mai il 100%. Ad esempio, supponiamo che l'utilizzo della memoria aumenti man mano che aggiungi carico e che l'applicazione si blocchi improvvisamente quando raggiunge il limite di memoria dell'attività o dell'istanza Amazon EC2. In questa situazione, è probabile che la memoria sia stata completamente consumata. L'applicazione potrebbe consumare più risorse. Pertanto, scegli la metrica che rappresenta la risorsa che si esaurisce per prima.

Infine, riprova a testare il carico dopo aver raddoppiato il numero di attività o di istanze Amazon EC2. Supponiamo che la metrica chiave aumenti o diminuisca della metà rispetto a prima. In tal caso, la metrica è proporzionale alla capacità. Questa è una buona metrica di utilizzo per la scalabilità automatica.

Consideriamo ora questo scenario ipotetico. Supponiamo di testare un'applicazione e di scoprire che l'utilizzo della CPU alla fine raggiunge l'80% a 100 richieste al secondo. Quando viene aggiunto più carico, non aumenta più l'utilizzo della CPU. Tuttavia, fa sì che l'applicazione risponda più lentamente. Quindi, si esegue nuovamente il test di carico, raddoppiando il numero di attività ma mantenendo la frequenza al valore di picco precedente. Se ritieni che l'utilizzo medio della CPU scenda a circa il 40%, l'utilizzo medio della CPU è un buon candidato per una metrica di scalabilità. D'altra parte, se l'utilizzo della CPU rimane all'80% dopo l'aumento del numero di attività, l'utilizzo medio della CPU non è una buona metrica di scalabilità. In tal caso, sono necessarie ulteriori ricerche per trovare una metrica adeguata.

Modelli applicativi e proprietà di scalabilità comuni

Vengono eseguiti software di tutti i tipi. AWS Molti carichi di lavoro sono creati internamente, mentre altri si basano su popolari software open source. Indipendentemente dalla loro origine, abbiamo osservato alcuni modelli di progettazione comuni per i servizi. La scalabilità efficace dipende in gran parte dal modello.

L'efficiente server legato alla CPU

L'efficiente server legato alla CPU non utilizza quasi nessuna risorsa oltre alla CPU e al throughput di rete. Ogni richiesta può essere gestita dalla sola applicazione. Le richieste non dipendono da altri servizi come i database. L'applicazione è in grado di gestire centinaia di migliaia di richieste simultanee e può utilizzare in modo efficiente più CPU per farlo. Ogni richiesta è gestita da un thread dedicato con un basso sovraccarico di memoria, oppure esiste un ciclo di eventi asincrono che viene eseguito su ogni CPU che soddisfa le richieste. Ogni replica dell'applicazione è ugualmente in grado di gestire una richiesta. L'unica risorsa che potrebbe esaurirsi prima della CPU è la larghezza di banda della rete. Nei servizi legati alla CPU, l'utilizzo della memoria, anche al picco della velocità effettiva, è una frazione delle risorse disponibili.

Questo tipo di applicazione è adatto per il ridimensionamento automatico basato su CPU. L'applicazione gode della massima flessibilità in termini di scalabilità. Può essere scalato verticalmente fornendo istanze Amazon EC2 più grandi o vCPU Fargate. Inoltre, può essere scalato orizzontalmente aggiungendo altre repliche. L'aggiunta di più repliche o il raddoppio delle dimensioni dell'istanza dimezzano l'utilizzo medio della CPU rispetto alla capacità.

Se utilizzi la capacità di Amazon EC2 per questa applicazione, valuta la possibilità di collocarla su istanze ottimizzate per il calcolo come la famiglia or. `c5` `c6g`

L'efficiente server legato alla memoria

L'efficiente server legato alla memoria alloca una quantità significativa di memoria per richiesta. In caso di massima concorrenza, ma non necessariamente della velocità effettiva, la memoria si esaurisce prima dell'esaurimento delle risorse della CPU. La memoria associata a una richiesta viene liberata al termine della richiesta. Le richieste aggiuntive possono essere accettate purché sia disponibile memoria.

Questo tipo di applicazione è adatto per il ridimensionamento automatico basato sulla memoria. L'applicazione gode della massima flessibilità in termini di scalabilità. Può essere scalato sia verticalmente fornendogli risorse di memoria Amazon EC2 o Fargate più grandi. Inoltre, può essere

ridimensionato orizzontalmente aggiungendo altre repliche. L'aggiunta di più repliche o il raddoppio delle dimensioni dell'istanza possono dimezzare l'utilizzo medio della memoria rispetto alla capacità.

Se utilizzi la capacità di Amazon EC2 per questa applicazione, valuta la possibilità di collocarla su istanze ottimizzate per la memoria come la famiglia or. `r5` `r6g`

Alcune applicazioni legate alla memoria non liberano la memoria associata a una richiesta al termine, quindi una riduzione della concorrenza non si traduce in una riduzione della memoria utilizzata. Per questo, non è consigliabile utilizzare il ridimensionamento basato sulla memoria.

Il server basato sul lavoratore

Il server basato sul lavoratore elabora una richiesta per ogni singolo thread di lavoro una dopo l'altra. I thread di lavoro possono essere thread leggeri, come thread POSIX. Possono anche essere thread più pesanti, come i processi UNIX. Indipendentemente dal thread, c'è sempre una concorrenza massima che l'applicazione è in grado di supportare. Di solito il limite di concorrenza è impostato proporzionalmente alle risorse di memoria disponibili. Se viene raggiunto il limite di concorrenza, le richieste aggiuntive vengono inserite in una coda di backlog. Se la coda del backlog supera i limiti, le richieste aggiuntive in arrivo vengono immediatamente respinte. Le applicazioni più comuni che si adattano a questo modello includono il server web Apache e Gunicorn.

La concorrenza delle richieste è in genere la metrica migliore per scalare questa applicazione. Poiché esiste un limite di concorrenza per ogni replica, è importante eseguire la scalabilità orizzontale prima che venga raggiunto il limite medio.

Il modo migliore per ottenere le metriche relative alla concorrenza delle richieste consiste nel farle riportare dall'applicazione. CloudWatch Ogni replica dell'applicazione può pubblicare il numero di richieste simultanee come metrica personalizzata ad alta frequenza. Si consiglia di impostare la frequenza su almeno una volta al minuto. Dopo aver raccolto diversi report, puoi utilizzare la concorrenza media come metrica di scala. Questa metrica viene calcolata prendendo la concorrenza totale e dividendola per il numero di repliche. Ad esempio, se la concorrenza totale è 1000 e il numero di repliche è 10, la concorrenza media è 100.

Se la tua applicazione è alla base di un Application Load Balancer, puoi anche utilizzare la `ActiveConnectionCount` metrica per il load balancer come fattore nella metrica di scalabilità. La `ActiveConnectionCount` metrica deve essere divisa per il numero di repliche per ottenere un valore medio. Per la scalatura deve essere utilizzato il valore medio, anziché il valore di conteggio non elaborato.

Affinché questa progettazione funzioni al meglio, la deviazione standard della latenza di risposta deve essere piccola a basse frequenze di richiesta. Consigliamo che, durante i periodi di scarsa domanda, la maggior parte delle richieste riceva risposta in breve tempo e che non molte richieste impieghino molto più tempo della media per rispondere. Il tempo di risposta medio dovrebbe essere vicino al tempo di risposta del 95° percentile. In caso contrario, potrebbero verificarsi sovraccarichi dalla coda. Ciò porta a errori. Si consiglia di fornire repliche aggiuntive laddove necessario per mitigare il rischio di overflow.

Il server di attesa

Il server di attesa esegue alcune elaborazioni per ogni richiesta, ma il funzionamento dipende in larga misura da uno o più servizi downstream. Le applicazioni container spesso fanno un uso intensivo di servizi downstream come database e altri servizi API. La risposta di questi servizi può richiedere del tempo, in particolare in scenari ad alta capacità o ad alta concorrenza. Questo perché queste applicazioni tendono a utilizzare poche risorse della CPU e a sfruttare la massima concorrenza in termini di memoria disponibile.

Il servizio di attesa è adatto sia nel modello di server legato alla memoria che nel modello di server basato sui lavoratori, a seconda di come è progettata l'applicazione. Se la concorrenza dell'applicazione è limitata solo dalla memoria, l'utilizzo medio della memoria deve essere utilizzato come metrica di scalabilità. Se la concorrenza dell'applicazione si basa su un limite di lavoratori, è necessario utilizzare la concorrenza media come metrica di scalabilità.

Il server basato su Java

Se il server basato su Java è legato alla CPU e si adatta proporzionalmente alle risorse della CPU, potrebbe essere adatto all'efficiente modello di server associato alla CPU. In tal caso, l'utilizzo medio della CPU potrebbe essere appropriato come metrica di scalabilità. Tuttavia, molte applicazioni Java non sono legate alla CPU, il che le rende difficili da scalare.

Per prestazioni ottimali, si consiglia di allocare quanta più memoria possibile all'heap Java Virtual Machine (JVM). Le versioni recenti di JVM, tra cui Java 8 update 191 o versioni successive, impostano automaticamente la dimensione dell'heap il più grande possibile per adattarla al contenitore. Ciò significa che, in Java, l'utilizzo della memoria è raramente proporzionale all'utilizzo delle applicazioni. Con l'aumento della frequenza delle richieste e della concorrenza, l'utilizzo della memoria rimane costante. Per questo motivo, non è consigliabile scalare i server basati su Java in base all'utilizzo della memoria. Al contrario, in genere consigliamo di ridimensionare l'utilizzo della CPU.

In alcuni casi, i server basati su Java incontrano l'esaurimento dell'heap prima di esaurire la CPU. Se l'applicazione è soggetta all'esaurimento dell'heap in caso di elevata simultaneità, la metrica di scalabilità migliore è rappresentata dalle connessioni medie. Se la tua applicazione è soggetta all'esaurimento dell'heap a un throughput elevato, la frequenza media delle richieste è la metrica di scalabilità migliore.

Server che utilizzano altri runtime raccolti con dati inutili

Molte applicazioni server si basano su runtime che eseguono la raccolta dei dati indesiderati, ad esempio .NET e Ruby. Queste applicazioni server potrebbero rientrare in uno dei modelli descritti in precedenza. Tuttavia, come nel caso di Java, non è consigliabile scalare queste applicazioni in base alla memoria, poiché il loro utilizzo medio della memoria osservato spesso non è correlato alla velocità effettiva o alla concorrenza.

Per queste applicazioni, si consiglia di ridimensionare l'utilizzo della CPU se l'applicazione è vincolata alla CPU. Altrimenti, consigliamo di scalare in base al throughput medio o alla concorrenza media, in base ai risultati dei test di carico.

Job processor

Molti carichi di lavoro prevedono l'elaborazione asincrona dei processi. Includono applicazioni che non ricevono richieste in tempo reale, ma si iscrivono a una coda di lavoro per ricevere offerte di lavoro. Per questi tipi di applicazioni, la metrica di scalabilità corretta è quasi sempre la profondità della coda. L'aumento della coda indica che il lavoro in sospeso supera la capacità di elaborazione, mentre una coda vuota indica che c'è più capacità del lavoro da fare.

AWS i servizi di messaggistica, come Amazon SQS e Amazon Kinesis Data Streams, CloudWatch forniscono metriche che possono essere utilizzate per la scalabilità. Per Amazon SQS, `ApproximateNumberOfMessagesVisible` è la metrica migliore. Per Kinesis Data Streams, prendi in considerazione l'`MillisBehindLatest`utilizzo della metrica, pubblicata dalla Kinesis Client Library (KCL). È necessario calcolare la media di questa metrica tra tutti i consumatori prima di utilizzarla per la scalabilità.

Scalabilità automatica e implementazioni dei servizi

Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante un'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non vengano sospesi. Se desideri sospendere i processi di scalabilità orizzontale durante le implementazioni in corso, attieniti alla seguente procedura.

1. Chiama il comando describe-scalable-targets specificando l'ID 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. Chiama il comando register-scalable-target, specificando l'ID risorsa, lo spazio dei nomi e la dimensione scalabile. Specifica `true` sia per `DynamicScalingInSuspended` che per `DynamicScalingOutSuspended`.
3. Al termine dell'implementazione, potrai chiamare il comando register-scalable-target e riprendere il dimensionamento.

Per ulteriori informazioni, consulta Sospensione e ripresa del dimensionamento per Application Auto Scaling.

Scala il tuo servizio Amazon ECS utilizzando un valore metrico target

Con le policy di dimensionamento con monitoraggio degli obiettivi, puoi scegliere un parametro e impostare un valore obiettivo. Amazon ECS Service Auto Scaling crea e gestisce gli allarmi che controllano CloudWatch la politica di scalabilità e calcola la regolazione della scalabilità in base alla metrica e al valore target. La policy di dimensionamento aggiunge o rimuove le attività di servizio in base alle necessità, per mantenere il parametro al valore di destinazione specificato o vicino a esso. Oltre a mantenere il parametro vicino al valore di destinazione, una policy di dimensionamento di monitoraggio obiettivi si adatta alle fluttuazioni del parametro dovute a un modello di carico fluttuante e riduce al minimo le fluttuazioni rapide del numero di attività in esecuzione all'interno del servizio.

Considerazioni

Tieni presenti le seguenti informazioni quando usi le policy di tracciamento delle destinazioni:

- Una policy di dimensionamento di monitoraggio obiettivi presuppone che essa debba eseguire un dimensionamento orizzontale quando il parametro specificato supera il valore di destinazione. Non puoi utilizzare una policy di dimensionamento di monitoraggio obiettivi per il dimensionamento orizzontale quando il parametro specificato è inferiore al valore di destinazione.
- Una policy di dimensionamento di monitoraggio obiettivi non esegue il dimensionamento quando il parametro specificato non dispone di dati sufficienti. Non esegue la scalabilità in quanto la carenza di dati non viene interpretata come basso utilizzo.
- Potrebbero esserci delle differenze tra il valore di destinazione e i punti di dati dei parametri reali. Ciò avviene perché Service Auto Scaling agisce sempre con prudenza, arrotondando per

eccesso o per difetto quando determina la capacità da aggiungere o rimuovere. In questo modo si impedisce l'aggiunta di capacità insufficiente o la rimozione di capacità eccessiva.

- Per garantire la disponibilità delle applicazioni, il servizio aumenta in proporzione al parametro il più veloce possibile, ma si riduce in modo più graduale.
- Application Auto Scaling disattiva la riduzione dei processi mentre sono in corso le implementazioni di Amazon ECS. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta Scalabilità automatica e implementazioni dei servizi.
- Puoi avere più policy di dimensionamento con monitoraggio degli obiettivi per un servizio Amazon ECS, purché ciascuna di esse utilizzi parametri differenti. Lo scopo di Service Auto Scaling è sempre quello di assegnare la priorità alla disponibilità, quindi il suo comportamento varia a seconda che le policy di monitoraggio degli obiettivi siano pronte o meno per l'aumento o la riduzione orizzontale. Il servizio viene aumentato se una qualsiasi delle policy di monitoraggio delle destinazioni è pronta per l'aumento, ma viene ridotto solo se tutte le policy di monitoraggio delle destinazioni (con la parte di riduzione abilitata) sono pronte per la riduzione.
- Non modificare o eliminare gli CloudWatch allarmi gestiti da Service Auto Scaling per una politica di scalabilità di tracciamento degli obiettivi. Service Auto Scaling elimina gli allarmi automaticamente quando elimini la policy di dimensionamento.
- Il parametro `ALBRequestCountPerTarget` per le policy di scalabilità del monitoraggio delle destinazioni non è supportato per il tipo di implementazione blu/verde.

Per ulteriori informazioni sulle policy di dimensionamento di tracciamento target, consulta la sezione relativa alle policy di dimensionamento di tracciamento target nella Guida per l'utente di Application Auto Scaling.

Per configurare le politiche di scalabilità target per il tuo servizio Amazon ECS utilizzando la console Amazon ECS

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS.
2. È possibile configurare una politica di scalabilità quando si crea o si aggiorna un servizio. Per ulteriori informazioni, consultare uno dei seguenti argomenti:

    - Creazione di un servizio utilizzando parametri definiti— Creare un nuovo servizio

- Aggiornamento di un servizio Amazon ECS tramite la console— Aggiornare un servizio esistente

Per configurare le politiche di scalabilità target per il tuo servizio Amazon ECS utilizzando AWS CLI

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS.
2. Registra il servizio Amazon ECS come destinazione scalabile utilizzando il comando register-scalable-target.
3. Creare una policy di dimensionamento utilizzando il comando put-scaling-policy.

Scala il tuo servizio Amazon ECS utilizzando incrementi predefiniti basati sugli allarmi CloudWatch

Con le politiche di scalabilità graduale, puoi specificare gli CloudWatch allarmi che avviano il processo di scalabilità. Ad esempio, se desideri eseguire la scalabilità orizzontale quando l'utilizzo della CPU raggiunge un certo livello, crea un allarme utilizzando la metrica fornita. `CPUUtilization` Quando si crea una policy di dimensionamento a fasi, bisogna specificare uno dei seguenti tipi di adeguamento dimensionamento:

- Aggiungi: aumenta il numero di attività in base a un numero specifico di unità di capacità o a una percentuale specificata della capacità corrente.
- Rimuovi: riduce il numero di attività in base a un numero specifico di unità di capacità o a una percentuale specificata della capacità corrente.
- Imposta su: imposta il numero di attività sul numero specificato di unità di capacità.

Per esempio, supponiamo che la capacità di destinazione e quella soddisfatta siano 10 e che la policy di dimensionamento aggiunga 1. Quando l'allarme viene violato, il processo di ridimensionamento automatico aggiunge da 1 a 10 per ottenere 11, quindi Amazon ECS avvia 1 attività per il servizio.

Ti consigliamo vivamente di utilizzare le politiche di scalabilità di Target Tracking per adattare parametri come l'utilizzo medio della CPU o il numero medio di richieste per target. Le metriche che diminuiscono all'aumentare della capacità e aumentano quando la capacità diminuisce possono essere utilizzate per ridimensionare proporzionalmente o in base al numero di attività utilizzando il

monitoraggio del target. Questo aiuta a garantire che Service Auto Scaling segua da vicino la curva di domanda per le vostre applicazioni.

Per una panoramica delle politiche di scalabilità in fasi e del loro funzionamento, consulta le politiche di scalabilità Step nella Application Auto Scaling User Guide. Dopo aver letto questa introduzione, consulta le seguenti sezioni per scoprire come configurare la scalabilità a fasi per Amazon ECS utilizzando la console e. AWS Command Line Interface

Per configurare le politiche di scalabilità graduale per il tuo servizio Amazon ECS utilizzando la console Amazon ECS

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS.
2. È possibile configurare una politica di scalabilità quando si crea o si aggiorna un servizio. Per ulteriori informazioni, consultare uno dei seguenti argomenti:
   - Creazione di un servizio utilizzando parametri definiti— Creare un nuovo servizio
   - Aggiornamento di un servizio Amazon ECS tramite la console— Aggiornare un servizio esistente

Per configurare le politiche di scalabilità graduale per il tuo servizio Amazon ECS utilizzando AWS CLI

1. Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie autorizzazioni aggiuntive. Per ulteriori informazioni, consulta Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS.
2. Registra il servizio Amazon ECS come destinazione scalabile utilizzando il comando register-scalable-target.
3. Creare una policy di dimensionamento utilizzando il comando put-scaling-policy.
4. Crea un allarme che avvia la politica di scalabilità utilizzando il comando put-metric-alarm.

Scala il tuo servizio Amazon ECS utilizzando una pianificazione

Con il dimensionamento pianificato, puoi impostare il dimensionamento automatico per la tua applicazione in base a variazioni di carico prevedibili, creando azioni pianificate che aumentano o

diminuiscono la capacità 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 una capacità sufficiente per gestire il picco di traffico infrasettimanale ma in altri momenti non fornisce una capacità non necessaria eccedente.

È possibile utilizzare simultaneamente il dimensionamento pianificato e le policy di dimensionamento per ottenere i vantaggi degli approccia 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 la capacità. In questo modo è possibile garantire di disporre di capacità sufficiente per la gestione dei carichi dell'applicazione. Sebbene l'applicazione si dimensioni per soddisfare la domanda, la capacità corrente deve rientrare nei valori di capacità minima e massima impostati dall'operazione pianificata.

Puoi configurare la scalabilità della pianificazione utilizzando. AWS CLI Per ulteriori informazioni sulla scalabilità pianificata, vedere Scheduled Scaling nella Application Auto Scaling User Guide.

Interconnetti i servizi Amazon ECS

Le applicazioni eseguite nelle attività di Amazon ECS spesso devono ricevere connessioni da Internet o connettersi ad altre applicazioni eseguite nei servizi Amazon ECS. Se hai bisogno di connessioni esterne da Internet, consigliamo di utilizzare Elastic Load Balancing. Per ulteriori informazioni sul bilanciamento del carico integrato, consulta the section called “Usa il bilanciamento del carico per distribuire il traffico di servizio”.

Se hai bisogno di un'applicazione per connetterti ad altre applicazioni eseguite nei servizi Amazon ECS, Amazon ECS offre i modi seguenti per farlo senza un sistema di bilanciamento del carico:

- Amazon ECS Service Connect

  Consigliamo Service Connect, che fornisce la configurazione di Amazon ECS per l'individuazione dei servizi, la connettività e il monitoraggio del traffico. Con Service Connect, le tue applicazioni possono utilizzare nomi brevi e porte standard per connettersi ai servizi Amazon ECS nello stesso cluster o in altri cluster, anche tra VPC dello stesso. Regione AWS

  Quando usi Service Connect, Amazon ECS gestisce tutte le parti del service discovery: crea i nomi che possono essere scoperti, gestisce dinamicamente le voci per ogni attività all'inizio e all'interruzione delle attività, esegue un agente in ogni attività configurato per scoprire i nomi. L'applicazione può cercare i nomi utilizzando la funzionalità standard per i nomi DNS e stabilendo

connessioni. Se l'applicazione lo fa già, non è necessario modificarla per utilizzare Service Connect.

Fornisci la configurazione completa all'interno di ogni definizione di servizio e attività. Amazon ECS gestisce le modifiche a questa configurazione in ogni distribuzione del servizio, per garantire che tutte le attività di una distribuzione si comportino nello stesso modo. Ad esempio, un problema comune relativo al DNS as service discovery è il controllo di una migrazione. Se si modifica un nome DNS in modo che punti ai nuovi indirizzi IP sostitutivi, potrebbe essere necessario il tempo TTL massimo prima che tutti i client inizino a utilizzare il nuovo servizio. Con Service Connect, la distribuzione del client aggiorna la configurazione sostituendo le attività del client. È possibile configurare l'interruttore automatico di distribuzione e altre configurazioni di distribuzione per influire sulle modifiche di Service Connect allo stesso modo di qualsiasi altra distribuzione.

Per ulteriori informazioni, consulta Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi.

- Individuazione dei servizi di Amazon ECS

  Un altro approccio alla service-to-service comunicazione è la comunicazione diretta tramite Service Discovery. In questo approccio, puoi utilizzare l'integrazione del rilevamento dei AWS Cloud Map servizi con Amazon ECS. Utilizzando Service Discovery, Amazon ECS sincronizza l'elenco delle attività avviate con AWS Cloud Map, che mantiene un nome host DNS che si risolve negli indirizzi IP interni di una o più attività di quel particolare servizio. Altri servizi in Amazon VPC possono utilizzare questo nome host DNS per inviare traffico direttamente a un altro contenitore utilizzando il suo indirizzo IP interno.

  Questo approccio alla service-to-service comunicazione offre una bassa latenza. Non ci sono componenti aggiuntivi tra i contenitori. Il traffico viaggia direttamente da un container all'altro.

  Questo approccio è adatto quando si utilizza la modalità `awsvpc` di rete, in cui ogni attività ha il proprio indirizzo IP univoco. La maggior parte dei software supporta solo l'uso di `A` record DNS, che si risolvono direttamente in indirizzi IP. Quando si utilizza la modalità di `awsvpc` rete, l'indirizzo IP per ogni operazione è un `A` record. Tuttavia, se utilizzi la modalità di `bridge` rete, è possibile che più contenitori condividano lo stesso indirizzo IP. Inoltre, le mappature dinamiche delle porte fanno sì che ai contenitori vengano assegnati in modo casuale i numeri di porta su quel singolo indirizzo IP. A questo punto, un `A` record non è più sufficiente per l'individuazione del servizio. È inoltre necessario utilizzare un `SRV` record. Questo tipo di record può tenere traccia sia degli indirizzi IP che dei numeri di porta, ma richiede la configurazione appropriata delle applicazioni. Alcune applicazioni predefinite utilizzate potrebbero non supportare `SRV` i record.

Un altro vantaggio della modalità di `awsvpc` rete è che si dispone di un gruppo di sicurezza unico per ogni servizio. È possibile configurare questo gruppo di sicurezza per consentire le connessioni in entrata solo dai servizi upstream specifici che devono comunicare con quel servizio.

Lo svantaggio principale della service-to-service comunicazione diretta tramite Service Discovery è che è necessario implementare una logica aggiuntiva per ripetere i tentativi e gestire gli errori di connessione. I record DNS hanno un periodo time-to-live (TTL) che controlla per quanto tempo vengono memorizzati nella cache. L'aggiornamento del record DNS e la scadenza della cache richiedono del tempo per consentire alle applicazioni di recuperare la versione più recente del record DNS. Pertanto, l'applicazione potrebbe finire per risolvere il record DNS in modo che punti a un altro contenitore che non è più presente. L'applicazione deve gestire i nuovi tentativi e disporre di una logica per ignorare i backend non validi.

Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS

Tabella di compatibilità della modalità di rete

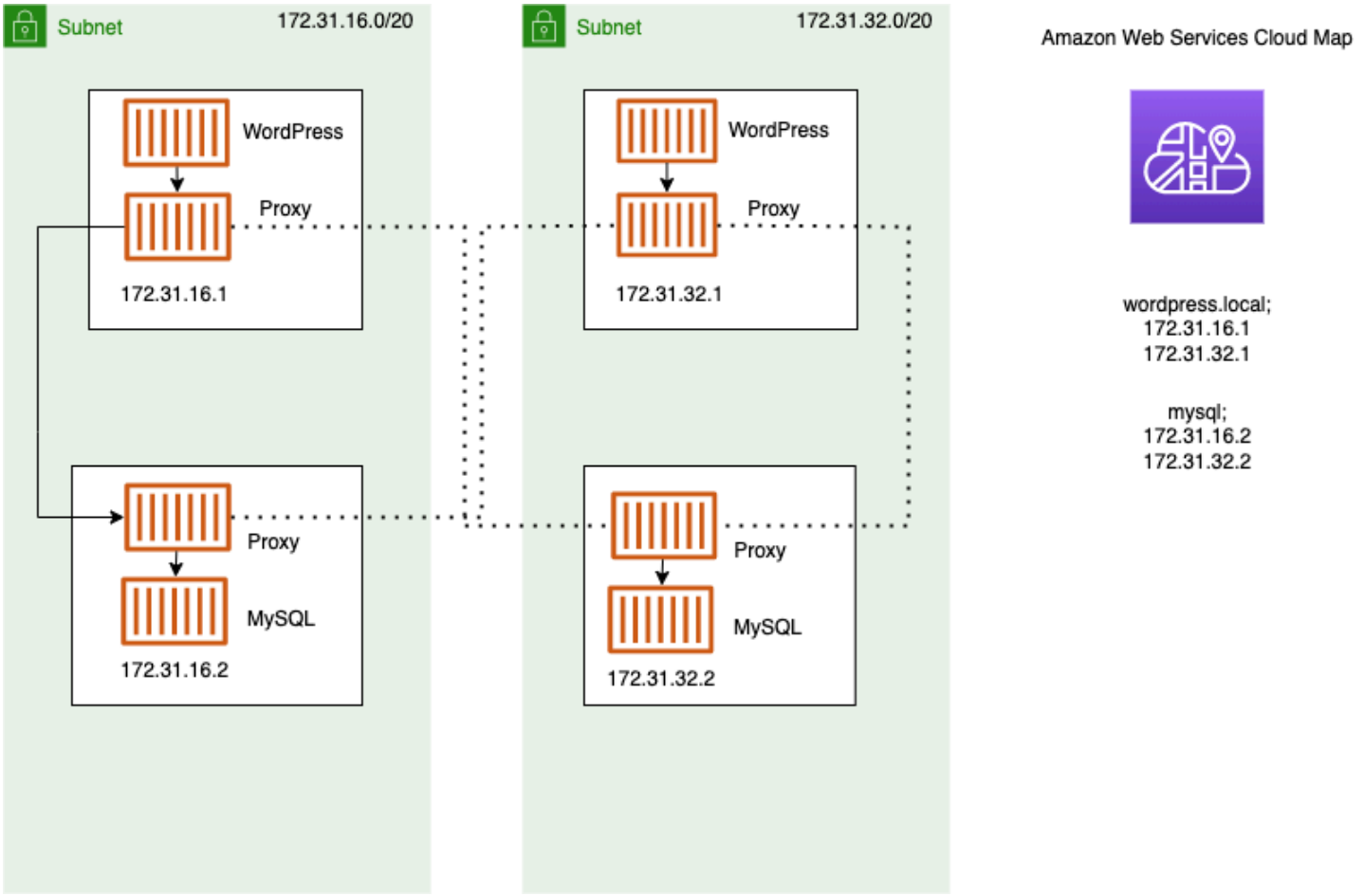
La tabella seguente illustra la compatibilità tra queste opzioni e le modalità di rete delle attività. Nella tabella, "client" si riferisce all'applicazione che effettua le connessioni dall'interno di un'attività Amazon ECS.

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

Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi

Amazon ECS Service Connect fornisce la gestione della service-to-service comunicazione come configurazione Amazon ECS. Crea sia l'individuazione dei servizi che una rete di servizi in Amazon ECS. Ciò fornisce la configurazione completa all'interno di ogni servizio che gestisci tramite distribuzioni di servizi, un modo unificato di fare riferimento ai tuoi servizi all'interno di namespace che non dipendono dalla configurazione DNS VPC e metriche e log standardizzati per monitorare tutte le tue applicazioni. Service Connect interconnette solo i servizi.

Il diagramma seguente mostra un esempio di rete Service Connect con 2 sottoreti nel VPC e 2 servizi. Un servizio client che viene eseguito WordPress con 1 attività in ciascuna sottorete. Un servizio server che esegue MySQL con 1 attività in ogni sottorete. Entrambi i servizi sono altamente disponibili e resistenti ai problemi relativi alle attività e alle zone di disponibilità, poiché ogni servizio esegue più attività distribuite su 2 sottoreti. Le frecce piene mostrano una connessione WordPress da MySQL. Ad esempio, un comando `mysql --host=mysql` CLI che viene eseguito dall'interno del WordPress contenitore nell'operazione con l'indirizzo IP. `172.31.16.1` Il comando utilizza il nome breve `mysql` sulla porta predefinita per MySQL. Questo nome e questa porta si connettono al proxy Service Connect nella stessa attività. Il proxy utilizzato nell' WordPress operazione utilizza il bilanciamento del carico a tutto tondo e tutte le informazioni relative agli errori precedenti nel rilevamento dei valori anomali per scegliere a quale task MySQL connettersi. Come mostrato dalle frecce piene nel diagramma, il proxy si connette al secondo proxy nell'attività MySQL con l'indirizzo IP `172.31.16.2`. Il secondo proxy si connette al server MySQL locale nella stessa attività. Entrambi i proxy riportano le prestazioni di connessione visibili nei grafici nelle console Amazon ECS e CloudWatch Amazon, in modo da poter ottenere i parametri delle prestazioni da tutti i tipi di applicazioni allo stesso modo.



I seguenti termini sono utilizzati insieme a Service Connect.

nome porta

La configurazione della definizione di attività di Amazon ECS che assegna un nome a una particolare mappatura delle porte. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

alias del cliente

La configurazione del servizio Amazon ECS che assegna il numero di porta utilizzato nell'endpoint. Inoltre, l'alias del client può assegnare il nome DNS dell'endpoint, sovrascrivendo il nome del rilevamento. Se nel servizio Amazon ECS non viene fornito un nome di rilevamento, l'alias del client sostituisce il nome della porta come nome dell'endpoint. Per esempi di endpoint, consulta la definizione di endpoint. È possibile assegnare più alias client a un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

nome del rilevamento

Il nome intermedio opzionale che è possibile creare per una porta specificata dalla definizione di attività. Questo nome viene utilizzato per creare un servizio. AWS Cloud Map Se questo nome non viene fornito, viene utilizzato il nome della porta indicato nella definizione di attività. È possibile assegnare più nomi di rilevamento a una porta specifica di un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

AWS Cloud Map i nomi dei servizi devono essere univoci all'interno di un namespace. A causa di questa limitazione, è possibile avere una sola configurazione di Service Connect senza un nome di rilevamento per una particolare definizione di attività in ogni spazio dei nomi.

endpoint

L'URL per connettersi a un'API o a un sito Web. L'URL contiene il protocollo, un nome DNS e la porta. Per ulteriori informazioni sugli endpoint in generale, consulta la sezione Endpoint nel glossario AWS all'interno della documentazione Riferimenti generali di Amazon Web Services.

Service Connect crea endpoint che si connettono ai servizi Amazon ECS e configura le attività nei servizi Amazon ECS per connettersi agli endpoint. L'URL contiene il protocollo, un nome DNS e la porta. Il protocollo e il nome della porta vengono selezionati nella definizione dell'attività, poiché la porta deve corrispondere all'applicazione che si trova all'interno dell'immagine di container. Nel servizio, si seleziona ogni porta in base al nome e si può assegnare il nome DNS. Se non si specifica un nome DNS nella configurazione del servizio Amazon ECS, per impostazione predefinita viene utilizzato il nome della porta indicato nella definizione di attività. Ad esempio, un endpoint Service Connect potrebbe essere `http://blog:80`, `grpc://checkout:8080` o `http://_db.production.internal:99`.

Servizio Service Connect

La configurazione di un singolo endpoint in un servizio Amazon ECS. Questa è una parte della configurazione Service Connect, costituita da una singola riga nella configurazione di Service Connect e del nome del rilevamento nella console o da un oggetto nell'elenco `services` nella configurazione JSON di un servizio Amazon ECS. Questa configurazione viene utilizzata solo da Amazon ECS Service Connect.

Per ulteriori informazioni, consulta ServiceConnectService in Amazon Elastic Container Service API Reference.

spazio dei nomi

Il nome breve o il nome completo Amazon Resource Name (ARN) dello spazio dei AWS Cloud Map nomi da utilizzare con Service Connect. Lo spazio dei nomi deve trovarsi nello stesso spazio del servizio Regione AWS e del cluster Amazon ECS. Il tipo di namespace in AWS Cloud Map non influisce su Service Connect.

Service Connect utilizza lo spazio dei AWS Cloud Map nomi come raggruppamento logico di attività Amazon ECS che comunicano tra loro. Ogni servizio Amazon ECS può appartenere a un solo spazio dei nomi. I servizi all'interno di un namespace possono essere distribuiti su diversi cluster Amazon ECS all'interno dello stesso nello stesso Regione AWS . Account AWS Puoi organizzare liberamente i servizi in base a qualsiasi criterio.

servizi client

Un servizio che esegue un'applicazione client di rete. Questo servizio deve avere uno spazio dei nomi configurato. Ogni attività del servizio può rilevare e connettersi a tutti gli endpoint nello spazio dei nomi tramite un container proxy Service Connect.

Se uno dei container dell'attività deve connettersi a un endpoint da un servizio in uno spazio dei nomi, scegli un servizio client. Se un'applicazione front-end, un proxy inverso o un sistema di bilanciamento del carico riceve traffico esterno tramite altri metodi come Elastic Load Balancing, potrebbe utilizzare questo tipo di configurazione di Service Connect.

servizio client-server

Un servizio Amazon ECS che esegue un'applicazione di rete o del servizio Web. Questo servizio deve avere uno spazio dei nomi e almeno un endpoint configurati. Ogni attività del servizio è raggiungibile utilizzando gli endpoint. Il container proxy Service Connect ascolta il nome e la porta dell'endpoint per indirizzare il traffico verso i container delle app nell'attività.

Se uno dei container espone e ascolta il traffico di rete su una porta, scegli un servizio client-server. Queste applicazioni non devono connettersi ad altri servizi client-server nello stesso spazio dei nomi, ma è necessaria la configurazione del client. Un backend, un middleware, un livello aziendale o la maggior parte dei microservizi possono utilizzare questo tipo di configurazione Service Connect. Se desideri che un'applicazione front-end, un proxy inverso o un sistema di bilanciamento del carico riceva traffico da altri servizi configurati con Service Connect nello stesso spazio dei nomi, questi servizi devono utilizzare questo tipo di configurazione Service Connect.

La funzionalità Service Connect crea una rete virtuale di servizi correlati. La stessa configurazione del servizio può essere utilizzata su più spazi dei nomi diversi per eseguire set di applicazioni indipendenti ma identici. Service Connect definisce il container del proxy nel servizio Amazon ECS. In questo modo, la stessa definizione di attività può essere utilizzata per eseguire applicazioni identiche in spazi dei nomi diversi con configurazioni Service Connect diverse. Ogni attività eseguita dal servizio esegue un contenitore proxy all'interno dell'attività.

Service Connect è adatto per connessioni tra servizi Amazon ECS all'interno dello stesso spazio dei nomi. Per le seguenti applicazioni, è necessario utilizzare un metodo di interconnessione aggiuntivo per connettersi a un servizio Amazon ECS configurato con Service Connect:

- Attività configurate in altri namespace
- Attività non configurate per Service Connect
- Altre applicazioni esterne ad Amazon ECS

Queste applicazioni possono connettersi tramite il proxy Service Connect, ma non possono risolvere i nomi degli endpoint Service Connect.

Affinché queste applicazioni risolvano gli indirizzi IP delle attività di Amazon ECS, è necessario utilizzare un altro metodo di interconnessione.

Prezzi

I prezzi di Amazon ECS Service Connect dipendono dall'utilizzo AWS Fargate o meno dell'infrastruttura Amazon EC2 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 direttamente Amazon EC2. Per ulteriori informazioni, consulta Prezzi di Amazon ECS.

AWS Cloud Map l'utilizzo è completamente gratuito, quando Service Connect lo utilizza.

Componenti Amazon ECS Service Connect

Quando usi Amazon ECS Service Connect, configuri ogni servizio Amazon ECS per eseguire un'applicazione server che riceve richieste di rete (servizio client-server) o per eseguire un'applicazione client che effettua le richieste (servizio client).

Quando ti prepari a iniziare a usare Service Connect, inizia con un servizio client-server. È possibile aggiungere una configurazione Service Connect a un nuovo servizio o a un servizio esistente. Amazon ECS crea un endpoint Service Connect nello spazio dei nomi. Inoltre, Amazon ECS crea una nuova implementazione nel servizio per sostituire le attività attualmente in esecuzione.

Le attività esistenti e altre applicazioni potranno continuare a connettersi agli endpoint esistenti e alle applicazioni esterne. Se un servizio client-server aggiunge attività scalando orizzontalmente, le nuove connessioni dei client verranno bilanciate tra tutte le attività. Se un servizio client-server viene aggiornato, le nuove connessioni dei client verranno bilanciate tra le attività della nuova versione.

Le attività esistenti non possono essere risolte e connettersi al nuovo endpoint. Solo le nuove attività con una configurazione Service Connect nello stesso namespace e che iniziano a essere eseguite dopo questa distribuzione possono risolversi e connettersi a questo endpoint.

Ciò significa che l'operatore dell'applicazione client determina quando la configurazione della propria app cambia, anche se l'operatore dell'applicazione server può modificare la configurazione in qualsiasi momento. L'elenco degli endpoint nel namespace può cambiare ogni volta che viene distribuito un servizio nel namespace. Le attività esistenti e le attività sostitutive continuano a comportarsi come dopo la distribuzione più recente.

Considera i seguenti esempi:

Innanzitutto, si supponga di creare un'applicazione disponibile sulla rete Internet pubblica in un unico AWS CloudFormation modello e in un unico AWS CloudFormation stack. La scoperta pubblica e la raggiungibilità dovrebbero essere create per ultimo AWS CloudFormation, incluso il servizio client frontend. I servizi devono essere creati in questo ordine per evitare un periodo di tempo in cui il servizio client di front-end è attivo e disponibile al pubblico, ma un back-end non lo è. Ciò elimina l'invio di messaggi di errore al pubblico durante quel periodo di tempo. Nel AWS CloudFormation, devi utilizzare il `dependsOn` per indicare AWS CloudFormation che non è possibile creare più servizi Amazon ECS in parallelo o contemporaneamente. È necessario aggiungere `dependsOn` al servizio client di front-end per ogni servizio client-server di back-end a cui si connettono le attività del client.

In secondo luogo, supponiamo che esista un servizio di front-end senza la configurazione di Service Connect. Le attività si connettono a un servizio di back-end esistente. Aggiungi prima una configurazione di Service Connect client-server al servizio di back-end utilizzando lo stesso nome nel DNS o `clientAlias` utilizzato dal front-end. Questo crea una nuova distribuzione, quindi tutti gli AWS SDK, il rilevamento del rollback AWS Management Console AWS CLI, e altri metodi servono per ripristinare il servizio di backend alla distribuzione e alla configurazione precedenti. Se sei soddisfatto delle prestazioni e del comportamento del servizio di back-end, aggiungi una configurazione di Service Connect client o client-server al servizio di front-end. Solo le attività della nuova implementazione utilizzano il proxy Service Connect aggiunto a quelle nuove attività. In caso di problemi con questa configurazione, puoi eseguire il rollback e ripristinare la configurazione precedente utilizzando il rilevamento del rollback di distribuzione oppure AWS Management Console gli AWS SDK e altri metodi per ripristinare e ripristinare il servizio di backend alla distribuzione e alla

configurazione precedenti. AWS CLI Se si utilizza un altro sistema di individuazione dei servizi basato sul DNS anziché su Service Connect, qualsiasi applicazione frontend o client inizia a utilizzare nuovi endpoint e modifica la configurazione degli endpoint dopo la scadenza della cache DNS locale, in genere impiegando diverse ore.

Rete

Per impostazione predefinita, il proxy Service Connect ascolta la mappatura delle porte `containerPort` dalla definizione dell'attività. Le regole del gruppo di sicurezza devono consentire il traffico in entrata (in ingresso) verso questa porta dalle sottoreti in cui verranno eseguiti i client.

Anche se si imposta un numero di porta nella configurazione del servizio Service Connect, ciò non cambia la porta per il servizio client-server su cui è in ascolto il proxy Service Connect. Quando imposti questo numero di porta, Amazon ECS modifica la porta dell'endpoint a cui si connettono i servizi client, sul proxy Service Connect all'interno di tali attività. Il proxy nel servizio client si connette al proxy nel servizio client-server utilizzando la `containerPort`.

Se desideri modificare la porta su cui il proxy Service Connect è in ascolto, modifica `ingressPort0verride` nella configurazione di Service Connect del servizio client-server. Se si modifica questo numero di porta, è necessario consentire il traffico in entrata su questa porta utilizzata dal traffico diretto a questo servizio.

Il traffico inviato dalle applicazioni ai servizi Amazon ECS configurati per Service Connect richiede che Amazon VPC e le sottoreti dispongano di regole della tabella di instradamento e di regole dell'ACL di rete che consentano i numeri di porta `containerPort` e `ingressPort0verride` che si stanno utilizzando.

Puoi utilizzare Service Connect per inviare traffico tra VPC. Gli stessi requisiti per le regole della tabella di routing, gli ACL di rete e i gruppi di sicurezza si applicano a entrambi i VPC.

Ad esempio, due cluster creano attività in diversi VPC. Un servizio in ogni cluster è configurato per utilizzare lo stesso spazio dei nomi. Le applicazioni di questi due servizi possono risolvere ogni endpoint nello spazio dei nomi senza alcuna configurazione DNS del VPC. Tuttavia, i proxy non possono connettersi a meno che il peering VPC, le tabelle di routing VPC o subnet e gli ACL di rete VPC consentano il traffico sui numeri di porta e. `containerPort` `ingressPort0verride`

Per le attività che utilizzano la modalità `bridge` di rete, è necessario creare un gruppo di sicurezza con una regola in entrata che consenta il traffico sull'intervallo di porte dinamiche superiore. Quindi, assegna il gruppo di sicurezza a tutte le istanze EC2 nel cluster Service Connect.

Proxy Service Connect

Se crei o aggiorni un servizio con la configurazione Service Connect, Amazon ECS aggiunge un nuovo contenitore a ogni nuova attività non appena viene avviata. Questo modello di utilizzo di un container separato è denominato `sidecar`. Questo container non è presente nella definizione dell'attività e non è possibile configurarlo. Amazon ECS gestisce la configurazione del contenitore nel servizio. Ciò consente di riutilizzare le stesse definizioni di attività tra più servizi, namespace e attività senza Service Connect.

Risorse proxy

- Per le definizioni delle attività, è necessario impostare i parametri di CPU e memoria.

  Si consiglia di aggiungere 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 di memoria minima che è possibile impostare è di 512 MiB di memoria. Su Amazon EC2, è richiesta la memoria per la definizione delle attività.
- Per il servizio, si imposta la configurazione del registro nella configurazione Service Connect.
- Se prevedi che le attività di questo servizio ricevano più di 500 richieste al secondo al massimo del carico, ti consigliamo di aggiungere 512 unità CPU alla CPU dell'attività in questa definizione di attività per il container del proxy Service Connect.
- Se prevedi di creare più di 100 servizi Service Connect nello spazio dei nomi o 2.000 attività in totale su tutti i servizi Amazon ECS all'interno dello spazio dei nomi, ti consigliamo di aggiungere 128 MiB di memoria alla memoria dell'attività per il container del proxy Service Connect. È necessario eseguire questa operazione in ogni definizione di attività utilizzata da tutti i servizi Amazon ECS nello spazio dei nomi.

Configurazione del proxy

Le applicazioni si connettono al proxy nel container sidecar nella stessa attività in cui si trova l'applicazione. Amazon ECS configura l'attività e i contenitori in modo che le applicazioni si connettano al proxy solo quando l'applicazione si connette ai nomi degli endpoint nello stesso spazio dei nomi. Tutto il resto del traffico non utilizza il proxy. L'altro traffico include indirizzi IP nello stesso VPC, endpoint di AWS servizio e traffico esterno.

Bilanciamento del carico

Service Connect configura il proxy per utilizzare la strategia round-robin per il bilanciamento del carico tra le attività in un endpoint Service Connect. Il proxy locale, che si trova nell'attività da cui proviene la connessione, seleziona una delle attività del servizio client-server che fornisce l'endpoint.

Ad esempio, si consideri un'attività eseguita WordPress in un servizio configurato come servizio client in uno spazio dei nomi chiamato locale. È presente un altro servizio con 2 attività che eseguono il database MySQL. Questo servizio è configurato per fornire un endpoint denominato `mysql` tramite Service Connect nello stesso spazio dei nomi. Nell' WordPress operazione, l' WordPress applicazione si connette al database utilizzando il nome dell'endpoint. Le connessioni a questo nome vanno al proxy che viene eseguito in un contenitore laterale durante la stessa attività. Quindi, il proxy può connettersi a una delle attività MySQL utilizzando la strategia round-robin.

Strategie di bilanciamento del carico: round-robin

Rilevamento di anomalie

Questa funzionalità utilizza i dati acquisiti dal proxy sulle connessioni non riuscite precedenti per evitare l'invio di nuove connessioni agli host che le contenevano. Service Connect configura la funzionalità di rilevamento delle anomalie del proxy per fornire controlli dell'integrità passivi.

Utilizzando l'esempio precedente, il proxy può connettersi a entrambe le attività MySQL. Se il proxy ha effettuato più connessioni a un'attività MySQL specifica e negli ultimi 30 secondi 5 di queste, o un numero maggiore, non sono andate a buon fine, il proxy evita di eseguire tale attività MySQL per un periodo compreso tra 30 e 300 secondi.

Tentativi

Service Connect configura il proxy in modo da tentare nuovamente le connessioni che passano attraverso il proxy e falliscono. Il secondo tentativo evita di utilizzare l'host delle connessioni precedenti. In questo modo si garantisce che ogni connessione attraverso Service Connect non si interrompa per motivi isolati.

Numero di tentativi: 2

Timeout

Service Connect configura il proxy in modo che attenda un tempo massimo per la risposta delle applicazioni client-server. Il valore di timeout predefinito è di 15 secondi, ma può essere aggiornato.

Parametri opzionali:

idleTimeout - La quantità di tempo in secondi in cui una connessione rimane attiva mentre è inattiva. Un valore di disabilita. `0` `idleTimeout`

L'`idleTimeout`impostazione predefinita per`HTTP/HTTP2/GRPC`è 5 minuti.

L'`idleTimeout`impostazione predefinita per `TCP` è 1 ora.

per RequestTimeout - Il periodo di attesa che l'upstream risponda con una risposta completa per richiesta. Il valore di `0` si spegne. `perRequestTimeout` Questo può essere impostato solo quando il contenitore dell'applicazione `appProtocol` for è`HTTP/HTTP2/GRPC`. L'impostazione predefinita è 15 secondi.

> **Note**
>
> Se `idleTimeout` è impostato su un tempo inferiore a`perRequestTimeout`, la connessione si chiuderà quando `idleTimeout` viene raggiunto il`perRequestTimeout`.

Considerazioni

Quando utilizzi Service Connect, considera quanto segue:

- Le attività eseguite in Fargate devono utilizzare la versione della piattaforma Fargate Linux `1.4.0` o superiore per utilizzare Service Connect.
- La versione dell'agente Amazon ECS sull'istanza del contenitore deve essere uguale `1.67.2` o superiore.
- Per utilizzare Service Connect, le istanze di container devono eseguire la versione `20230428` dell'AMI di Amazon Linux 2023 ottimizzata per Amazon ECS, o versioni successive, o la versione `2.0.20221115` dell'AMI di Amazon Linux 2 ottimizzata per Amazon ECS. Oltre all'agente del container Amazon ECS, queste versioni contengono l'agente Service Connect. Per ulteriori informazioni sull'agente Service Connect, consulta Amazon ECS Service Connect Agent on GitHub.
- Le istanze di container devono disporre dell'autorizzazione `ecs:Poll` per la risorsa `arn:aws:ecs:`*`region`*`:`*`0123456789012`*`:task-set/`*`cluster`*`/*`. Se utilizzi il `ecsInstanceRole`, non è necessario aggiungere altre autorizzazioni. La policy gestita da `AmazonEC2ContainerServiceforEC2Role` dispone delle autorizzazioni necessarie. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

- Solo i servizi che utilizzano implementazioni in sequenza sono supportati con Service Connect.
- Le attività che utilizzano la modalità di `bridge` rete e utilizzano Service Connect non supportano il parametro di definizione del `hostname` contenitore.
- Per utilizzare Service Connect, le definizioni di attività devono impostare il limite di memoria dell'attività. Per ulteriori informazioni, consulta Proxy Service Connect.
- Le definizioni delle attività che impostano i limiti di memoria del contenitore non sono supportate.

  È possibile impostare i limiti di memoria del container sui container, ma è necessario impostare il limite di memoria delle attività su un numero maggiore della somma dei limiti di memoria del container. La CPU e la memoria aggiuntive nei limiti delle attività che non vengono allocate nei limiti del container negli altri contenitori vengono utilizzate dal container del proxy Service Connect e da altri container che non impostano limiti di container. Per ulteriori informazioni, consulta Proxy Service Connect.
- È possibile configurare Service Connect per utilizzare qualsiasi spazio dei AWS Cloud Map nomi nella stessa regione nella stessa. Account AWS
- Ogni servizio può appartenere a un solo namespace.
- Sono supportate solo le attività create dai servizi.
- Tutti gli endpoint devono essere univoci all'interno di uno spazio dei nomi.
- Tutti i nomi di rilevamento devono essere univoci all'interno di uno spazio dei nomi.
- È necessario ridistribuire i servizi esistenti prima che le applicazioni possano risolvere nuovi endpoint. I nuovi endpoint aggiunti allo spazio dei nomi dopo l'implementazione più recente non verranno aggiunti alla configurazione dell'attività. Per ulteriori informazioni, consulta the section called “Componenti Service Connect”.
- Service Connect non elimina i namespace quando vengono eliminati i cluster. È necessario eliminare i namespace in. AWS Cloud Map
- Per impostazione predefinita, il traffico di Application Load Balancer viene instradato tramite l'agente Service Connect in modalità rete. `awsvpc` Se desideri che il traffico non di servizio aggiri l'agente Service Connect, utilizza il `ingressPortOverride` parametro nella configurazione del servizio Service Connect.

Service Connect non supporta quanto segue:

- Container Windows
- HTTP 1.0

- Attività autonome
- Servizi che utilizzano i tipi di distribuzione blu/verde ed esterna
- Le istanze di container `External` per Amazon ECS Anywhere non sono supportate con Service Connect.
- PPv2

Regioni con Service Connect

Amazon ECS Service Connect è disponibile nelle seguenti AWS regioni:

Nome della regione	Regione
US East (Ohio)	us-east-2
US East (N. Virginia)	us-east-1
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacifico (Melbourne)	ap-southeast-4

Nome della regione	Regione
Asia Pacifico (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
Cina (Pechino)	cn-north-1 (Nota: TLS for Service Connect non è disponibile in questa regione.)
Cina (Ningxia)	cn-northwest-1 (Nota: TLS for Service Connect non è disponibile in questa regione.)
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Spagna)	eu-south-2
Europa (Stoccolma)	eu-north-1
Europa (Zurigo)	eu-central-2
Israele (Tel Aviv)	il-central-1
Medio Oriente (Bahrein)	me-south-1
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Sud America (São Paulo)	sa-east-1

Panoramica della configurazione di Amazon ECS Service Connect

Quando si utilizza Service Connect, è necessario configurare dei parametri nelle risorse.

Risorse Amazon ECS che devono essere configurate per Service Connect

Posizione dei parametri	Tipo di app	Descrizione	Richiesto
Definizione di attività	Client	Non sono disponibili modifiche per Service Connect nelle definizioni delle attività del client.	N/D
Definizione di attività	Client-server	I server devono aggiungere i campi `name` alle porte nelle `portMappings` dei container. Per ulteriori informazioni, consulta `portMappings`	Sì
Definizione di attività	Client-server	I server possono opzionalmente fornire un protocollo applicativo (ad esempio, HTTP) per ricevere parametri specifici del protocollo per le loro applicazioni server (ad esempio, `HTTP 5xx`).	No
Definizioni di servizi	Client	I servizi client devono aggiungere una `serviceConnectConfiguration` per configurare lo spazio dei nomi a cui aderire. Questo spazio dei nomi deve contenere tutti i servizi del server che questo servizio deve individuare. Per ulteriori informazioni, consulta `serviceConnectConfiguration`.	Sì
Definizione di servizio	Client-server	I servizi server devono aggiungere una `serviceConnectConfiguration` per configurare i nomi DNS, i numeri di porta e lo spazio dei nomi da cui è disponibile il servizio. Per ulteriori informazioni, consulta `serviceConnectConfiguration`.	Sì
Cluster	Client	I cluster possono aggiungere uno spazio dei nomi Service Connect predefinito.	No

Posizione dei parametri	Tipo di app	Descrizione	Richiesto
		Quando Service Connect è configurato in un servizio, i nuovi servizi nel cluster ereditano lo spazio dei nomi.	
Cluster	Client-server	Non sono disponibili modifiche per Service Connect nei cluster che si applicano ai servizi server. Le definizioni e i servizi delle attività del server devono impostare la rispettiva configurazione.	N/D

Panoramica della procedura per configurare Service Connect

I passaggi seguenti forniscono una panoramica su come configurare Service Connect.

> **Important**
>
> - Service Connect crea AWS Cloud Map servizi nel tuo account. La modifica di queste risorse AWS Cloud Map mediante la registrazione/deregistrazione manuale delle istanze, la modifica degli attributi dell'istanza o l'eliminazione di un servizio può portare a un comportamento imprevisto per il traffico dell'applicazione o le implementazioni successive.
> - Service Connect non supporta i collegamenti nella definizione dell'attività.

1. Aggiungi i nomi delle porte alle mappature delle porte nelle definizioni di attività. Per ottenere parametri aggiuntivi, puoi inoltre identificare il protocollo di livello 7 dell'applicazione.
2. Crea un cluster con uno spazio AWS Cloud Map dei nomi o crea lo spazio dei nomi separatamente. Per un'organizzazione semplice, crea un cluster con il nome che desideri per lo spazio dei nomi e specifica lo stesso nome per lo spazio dei nomi. In questo caso, Amazon ECS crea un nuovo spazio dei nomi HTTP con la configurazione necessaria. Service Connect non utilizza né crea zone ospitate DNS in Amazon Route 53.
3. Configura i servizi per creare endpoint Service Connect all'interno dello spazio dei nomi.
4. Implementa i servizi per creare gli endpoint. Amazon ECS aggiunge un container del proxy Service Connect a ogni attività e crea gli endpoint Service Connect in AWS Cloud Map. Questo container

non è configurato nella definizione dell'attività e la definizione dell'attività può essere riutilizzata senza modifiche per creare più servizi nello stesso spazio dei nomi o in più spazi dei nomi.

5. Implementa le app client come servizi per connetterti agli endpoint. Amazon ECS li connette agli endpoint Service Connect tramite il proxy Service Connect in ogni attività.

   Le applicazioni utilizzano il proxy solo per connettersi agli endpoint Service Connect. Non è disponibile alcuna configurazione aggiuntiva per l'utilizzo del proxy. Il proxy esegue il bilanciamento del carico round-robin, il rilevamento di anomalie ed effettua un nuovo tentativo. Per ulteriori informazioni sul proxy, consulta Proxy Service Connect.
6. Monitora il traffico tramite il proxy Service Connect in Amazon CloudWatch.

Configurazione del cluster

È possibile impostare uno spazio dei nomi predefinito per Service Connect quando si crea o si aggiorna il cluster. Se si specifica un nome per lo spazio dei nomi che non esiste nella stessa Regione AWS e nello stesso account, viene creato un nuovo spazio dei nomi HTTP.

Se crei un cluster e specifichi uno spazio dei nomi Service Connect predefinito, il cluster attende lo stato `PROVISIONING` mentre Amazon ECS crea lo spazio dei nomi. Puoi vedere un `attachment` nello stato del cluster che mostra lo stato dello spazio dei nomi. Per impostazione predefinita, gli allegati non vengono visualizzati in. È necessario aggiungerli `--include ATTACHMENTS` per visualizzarli. AWS CLI

Configurazione del servizio

Service Connect è progettato per richiedere la configurazione minima. È necessario impostare un nome per ogni mappatura delle porte che si desidera utilizzare con Service Connect nella definizione di attività. Nel servizio, per creare un servizio client è necessario attivare Service Connect e selezionare uno spazio dei nomi. Per creare un servizio client-server, è necessario aggiungere una singola configurazione del servizio Service Connect che corrisponda al nome di una delle mappature delle porte. Amazon ECS riutilizza il numero di porta e il nome della porta dalla definizione dell'attività per definire il servizio e l'endpoint Service Connect. Per sovrascrivere questi valori, puoi utilizzare gli altri parametri Discovery, DNS e Port nella console o `discoveryName` e `clientAliases`, rispettivamente, nell'API Amazon ECS.

L'esempio seguente mostra ogni tipo di configurazione Service Connect utilizzata insieme nello stesso servizio Amazon ECS. Sono forniti commenti nella shell (interprete di comandi), tuttavia tieni presente che la configurazione JSON utilizzata per i servizi Amazon ECS non supporta i commenti.

```
{
    ...
    serviceConnectConfiguration: {
        enabled: true,
        namespace: "internal",
        #config for client services can end here, only these two parameters are
 required.
        services: [{
                portName: "http"
            }, #minimal client - server service config can end here.portName must match
 the "name"
            parameter of a port mapping in the task definition. {
                discoveryName: "http-second"
                #name the discoveryName to avoid a Task def port name collision with
 the minimal config in the same Cloud Map namespace
                portName: "http"
            },
            {
                clientAliases: [{
                        dnsName: "db",
                        port: 81
                    }] #use when the port in Task def is not the port that client apps
 use.Client apps can use http: //db:81 to connect
                    discoveryName: "http-three"
                portName: "http"
            },
            {
                clientAliases: [{
                    dnsName: "db.app",
                    port: 81
                }] #use when the port in Task def is not the port that client apps
 use.duplicates are fine as long as the discoveryName is different.
                discoveryName: "http-four"
                portName: "http",
                ingressPortOverride: 99 #If App should also accept traffic directly on
 Task def port.
            }
        ]
    }
}
```

Crittografa il traffico Amazon ECS Service Connect

Amazon ECS Service Connect supporta la crittografia automatica del traffico con certificati Transport Layer Security (TLS) per i servizi Amazon ECS. Quando indirizzi i tuoi servizi Amazon ECS verso un AWS Private Certificate Authority (AWS Private CA), Amazon ECS fornisce automaticamente certificati TLS per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect. Amazon ECS genera, ruota e distribuisce i certificati TLS utilizzati per la crittografia del traffico.

La crittografia automatica del traffico con Service Connect utilizza funzionalità di crittografia leader del settore per proteggere le comunicazioni tra servizi e soddisfare i requisiti di sicurezza. Supporta certificati AWS Private Certificate Authority TLS con crittografia. `256-bit ECDSA` `2048-bit RSA` Per impostazione predefinita, TLS 1.3 è supportato, ma TLS 1.0 - 1.2 non lo sono. Hai anche il pieno controllo sui certificati privati e sulle chiavi di firma per aiutarti a soddisfare i requisiti di conformità.

> **Note**
>
> Per utilizzare TLS 1.3, è necessario abilitarlo sul listener sulla destinazione.
> Solo il traffico in entrata e in uscita che passa attraverso l'agente Amazon ECS è crittografato.

AWS Private Certificate Authority certificati e Service Connect

Sono necessarie autorizzazioni IAM aggiuntive per emettere certificati. Amazon ECS fornisce una policy di fiducia delle risorse gestite che delinea il set di autorizzazioni. Per ulteriori informazioni su questa politica, consulta Amazonecs Security. InfrastructureRole PolicyFor ServiceConnect TransportLayer

AWS Private Certificate Authority modalità per Service Connect

AWS Private Certificate Authority può funzionare in due modalità: generica e di breve durata.

- Scopo generale - Certificati che possono essere configurati con qualsiasi data di scadenza.
- Di breve durata - Certificati con una validità massima di sette giorni.

Sebbene Amazon ECS supporti entrambe le modalità, consigliamo di utilizzare certificati di breve durata. Per impostazione predefinita, i certificati ruotano ogni cinque giorni e l'esecuzione in modalità di breve durata offre risparmi significativi sui costi rispetto agli usi generici.

Service Connect non supporta la revoca dei certificati e sfrutta invece certificati di breve durata con rotazione frequente dei certificati. Hai l'autorità per modificare la frequenza di rotazione, disabilitare

o eliminare i segreti utilizzando la rotazione gestita in Secrets Manager, ma ciò può comportare le seguenti possibili conseguenze.

- Frequenza di rotazione più breve - Una frequenza di rotazione più breve comporta costi più elevati a AWS Private CA causa di Secrets Manager AWS KMS e Auto Scaling che subiscono un maggiore carico di lavoro per la rotazione.
- Frequenza di rotazione più lunga - Le comunicazioni delle applicazioni falliscono se la frequenza di rotazione supera i sette giorni.
- Eliminazione del segreto - L'eliminazione del segreto comporta un errore di rotazione e influisce sulle comunicazioni con le applicazioni dei clienti.

Se la rotazione segreta fallisce, viene pubblicato un `RotationFailed` evento in. AWS CloudTrail Puoi anche impostare un CloudWatch allarme per`RotationFailed`.

> **Important**
>
> Non aggiungere regioni di replica ai segreti. In questo modo si impedisce ad Amazon ECS di eliminare il segreto, poiché Amazon ECS non dispone dell'autorizzazione per rimuovere le regioni dalla replica. Se hai già aggiunto la replica, esegui il seguente comando.
>
> ```
> aws secretsmanager remove-regions-from-replication \
>  --secret-id SecretId \
>  --remove-replica-regions region-name
> ```

Autorità di certificazione subordinate

È possibile importare qualsiasi certificato AWS Private CA, root o subordinato, a Service Connect TLS per emettere certificati di entità finale per i servizi. L'emittente fornito viene considerato il firmatario e la fonte della fiducia ovunque. È possibile emettere certificati di entità finale per diverse parti dell'applicazione da diverse CA subordinate. Quando utilizzi AWS CLI, fornisci l'Amazon Resource Name (ARN) della CA per stabilire la catena di fiducia.

Autorità di certificazione locali

Per utilizzare la CA locale, è necessario creare e configurare una CA subordinata in. AWS Private Certificate Authority Ciò garantisce che tutti i certificati TLS emessi per i carichi di lavoro Amazon

ECS condividano la catena di fiducia con i carichi di lavoro eseguiti in locale e siano in grado di connettersi in modo sicuro.

> **Important**
>
> Aggiungi il tag richiesto nel tuo. `AmazonECSManaged : true` AWS Private CA

infrastruttura come codice

Quando si utilizzano Service Connect TLS con gli strumenti Infrastructure as Code (IaC), è importante configurare correttamente le dipendenze per evitare problemi, come i servizi bloccati. La AWS KMS chiave, se fornita, il ruolo IAM e AWS Private CA le dipendenze devono essere eliminati dopo il servizio Amazon ECS.

Service Connect e AWS Key Management Service

È possibile utilizzare AWS Key Management Serviceper 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 valori *di input dell'utente* con i tuoi.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "id",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/role-name"
```

```
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyPair"
      ],
      "Resource": "*"
    }
  ]
}
```

Per ulteriori informazioni sulle politiche chiave, consulta Creazione di una politica chiave nella Guida per gli AWS Key Management Service sviluppatori.

> **Note**
>
> Service Connect supporta solo chiavi di crittografia AWS KMS simmetriche. Non è possibile utilizzare nessun altro tipo di AWS KMS chiave per crittografare le risorse Service Connect. Per informazioni su come determinare se una AWS KMS chiave è una chiave di crittografia simmetrica, vedi Identificazione delle chiavi simmetriche e asimmetriche. AWS KMS

Per ulteriori informazioni sulle chiavi di crittografia AWS Key Management Service simmetriche, consulta Chiavi di crittografia simmetriche nella Guida per gli sviluppatori. AWS KMS AWS Key Management Service

Abilitazione del TLS per Amazon ECS Service Connect

La crittografia del traffico viene abilitata quando si crea o si aggiorna un servizio Service Connect.

Per abilitare la crittografia del traffico per un servizio in uno spazio dei nomi esistente utilizzando il AWS Management Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel riquadro di navigazione seleziona Namespaces (Spazio dei nomi).
3. Scegli il Namespace con il servizio per cui desideri abilitare la crittografia del traffico.
4. Scegli il servizio per cui desideri abilitare la crittografia del traffico.
5. Scegli Update Service nell'angolo in alto a destra e scorri verso il basso fino alla sezione Service Connect.

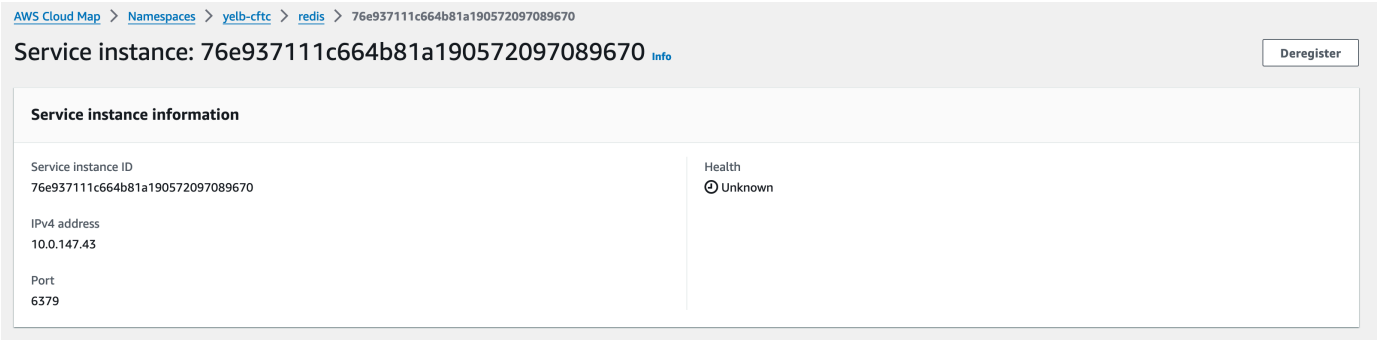
6. Scegli Attiva la crittografia del traffico tra le informazioni sul servizio per abilitare TLS.
7. Per il ruolo TLS di Service Connect, scegli un ruolo esistente o creane uno nuovo.
8. Per l'autorità di certificazione Signer, scegli un'autorità di certificazione esistente o creane una nuova.
9. Per Choose an AWS KMS key, scegli una chiave AWS proprietaria e gestita oppure puoi scegliere una chiave diversa. Puoi anche scegliere di crearne una nuova.

Per un esempio di utilizzo di TLS AWS CLI per configurare il servizio,Configurazione di Amazon ECS Service Connect con AWS CLI.

La verifica che TLS sia abilitato per Amazon ECS Service Connect

Service Connect avvia TLS nell'agente Service Connect e lo termina nell'agente di destinazione. Di conseguenza, il codice dell'applicazione non vede mai le interazioni TLS. Utilizza i passaggi seguenti per verificare che TLS sia abilitato.

1. Assicurati che l'immagine dell'applicazione abbia la `openssl` CLI.
2. Abilita ECS Exec sui tuoi servizi per connetterti alle tue attività tramite SSM. In alternativa, puoi avviare un'istanza Amazon EC2 nello stesso Amazon VPC del servizio.
3. Recupera l'IP e la porta di un'attività da un servizio che desideri verificare. Ad esempio, se il `redis` servizio ha il TLS attivato, puoi recuperare l'IP dell'attività accedendo AWS Cloud Map, trovando il servizio e controllando l'IP e la porta di un'istanza.



4. Accedi a una qualsiasi delle tue attività utilizzando `execute-command` like nell'esempio seguente. In alternativa, accedi all'istanza Amazon EC2 creata nella Fase 2.

```
$ aws ecs execute-command --cluster cluster-name \
    --task < TASK_ID>  \
    --container app \
    --interactive \
```

```
    --command "/bin/sh"
```

> **Note**
>
> La chiamata diretta al nome DNS non rivela il certificato.

5. Nella shell connessa, utilizza la `openssl` CLI per verificare e visualizzare il certificato allegato all'attività.

   Esempio:

   ```
   openssl s_client -connect 10.0.147.43:6379 < /dev/null 2> /dev/null \
   | openssl x509 -noout -text
   ```

   Risposta di esempio:

   ```
   Certificate:
       Data:
           Version: 3 (0x2)
           Serial Number:
               <serial-number>
           Signature Algorithm: ecdsa-with-SHA256
           Issuer: <issuer>
           Validity
               Not Before: Jan 23 21:38:12 2024 GMT
               Not After : Jan 30 22:38:12 2024 GMT
           Subject: <subject>
           Subject Public Key Info:
               Public Key Algorithm: id-ecPublicKey
                   Public-Key: (256 bit)
                   pub:
                       <pub>
                   ASN1 OID: prime256v1
                   NIST CURVE: P-256
           X509v3 extensions:
               X509v3 Subject Alternative Name:
                   DNS:redis.yelb-cftc
               X509v3 Basic Constraints:
                   CA:FALSE
               X509v3 Authority Key Identifier:
                   keyid:<key-id>
   ```

```
            X509v3 Subject Key Identifier:
                1D:<id>
            X509v3 Key Usage: critical
                Digital Signature, Key Encipherment
            X509v3 Extended Key Usage:
                TLS Web Server Authentication, TLS Web Client Authentication
    Signature Algorithm: ecdsa-with-SHA256
        <hash>
```

Configurazione di Amazon ECS Service Connect con AWS CLI

Puoi creare un servizio Amazon ECS per un'attività Fargate che utilizza Service Connect con. AWS CLI

Prerequisiti

Di seguito sono riportati i prerequisiti di Service Connect:

- Verifica che la regione supporti Service Connect. Per ulteriori informazioni, consulta Regions with Service Connect.
- Verificare che sia installata e configurata AWS CLI la versione più recente di. Per ulteriori informazioni, consulta l'argomento relativo all'installazione di AWS Command Line Interface.
- Il tuo AWS utente dispone delle autorizzazioni richieste specificate nell'esempio di policy AmazonECS_ FullAccess IAM.
- Sono disponibili un VPC, una sottorete, una tabella di instradamento e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.
- Hai un ruolo di esecuzione delle attività con il nome `ecsTaskExecutionRole` e la policy gestita da `AmazonECSTaskExecutionRolePolicy` è associata al ruolo. Questo ruolo consente a Fargate di scrivere i log delle applicazioni NGINX e i log del proxy Service Connect su Amazon Logs. CloudWatch Per ulteriori informazioni, consulta Creazione del ruolo di esecuzione attività.

Fase 1: Creazione del cluster

Utilizza la procedura seguente per creare un cluster e uno spazio dei nomi Amazon ECS.

Per creare il cluster e AWS Cloud Map lo spazio dei nomi Amazon ECS

1. Crea un cluster Amazon ECS denominato `tutorial` da utilizzare. Il parametro `--service-connect-defaults` imposta lo spazio dei nomi predefinito del cluster. Nell'output di esempio, uno AWS Cloud Map spazio dei nomi con il nome `service-connect` non esiste in questo account e Regione AWS quindi lo spazio dei nomi viene creato da Amazon ECS. Lo spazio dei nomi è creato in AWS Cloud Map nell'account ed è visibile insieme a tutti gli altri spazi dei nomi, quindi utilizza un nome che ne indichi lo scopo.

```
aws ecs create-cluster --cluster-name tutorial --service-connect-defaults
 namespace=service-connect
```

Output:

```
{
    "cluster": {
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",
        "clusterName": "tutorial",
        "serviceConnectDefaults": {
            "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-EXAMPLE"
        },
        "status": "PROVISIONING",
        "registeredContainerInstancesCount": 0,
        "runningTasksCount": 0,
        "pendingTasksCount": 0,
        "activeServicesCount": 0,
        "statistics": [],
        "tags": [],
        "settings": [
            {
                "name": "containerInsights",
                "value": "disabled"
            }
        ],
        "capacityProviders": [],
        "defaultCapacityProviderStrategy": [],
        "attachments": [
            {
                "id": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
                "type": "sc",
                "status": "ATTACHING",
```

```
                "details": []
            }
        ],
        "attachmentsStatus": "UPDATE_IN_PROGRESS"
    }
}
}
```

2. Verifica che il cluster sia stato creato:

```
aws ecs describe-clusters --clusters tutorial
```

Output:

```
{
    "clusters": [
        {
            "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",
            "clusterName": "tutorial",
            "serviceConnectDefaults": {
                "namespace": "arn:aws:servicediscovery:us-
west-2:123456789012:namespace/ns-EXAMPLE"
            },
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 0,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [],
            "settings": [],
            "capacityProviders": [],
            "defaultCapacityProviderStrategy": []
        }
    ],
    "failures": []
}
```

3. (Facoltativo) Verifica che lo spazio dei nomi sia stato creato in. AWS Cloud MapÈ possibile utilizzare la configurazione AWS Management Console o la AWS CLI configurazione normale così come viene creata in. AWS Cloud Map

Ad esempio, utilizza la AWS CLI.

```
aws servicediscovery --region us-west-2 get-namespace --id ns-EXAMPLE
```

Output:

```
{
    "Namespace": {
        "Id": "ns-EXAMPLE",
        "Arn": "arn:aws:servicediscovery:us-west-2:123456789012:namespace/ns-
EXAMPLE",
        "Name": "service-connect",
        "Type": "HTTP",
        "Properties": {
            "DnsProperties": {
                "SOA": {}
            },
            "HttpProperties": {
                "HttpName": "service-connect"
            }
        },
        "CreateDate": 1661749852.422,
        "CreatorRequestId": "service-connect"
    }
}
```

Fase 2: Creare il servizio per il server

La funzionalità Service Connect è progettata per l'interconnessione di più applicazioni su Amazon ECS. Almeno una di queste applicazioni deve fornire un servizio Web a cui connettersi. In questa fase viene creata:

- La definizione dell'attività che utilizza l'immagine del container NGINX ufficiale non modificata e include la configurazione di Service Connect.
- La definizione del servizio Amazon ECS che configura Service Connect per fornire il rilevamento del servizio e il proxy service mesh per il traffico verso questo servizio. La configurazione riutilizza lo spazio dei nomi predefinito della configurazione del cluster per ridurre la quantità di operazioni di configurazione del servizio effettuata per ciascun servizio.

- Il servizio Amazon ECS. Esegue un'attività utilizzando la definizione di attività e inserisce un container aggiuntivo per il proxy Service Connect. Il proxy Service Connect è in ascolto sulla porta dalla mappatura della porta del container nella definizione di attività. In un'applicazione client in esecuzione su Amazon ECS, il proxy nell'attività client ascolta le connessioni in uscita al nome della porta di definizione dell'attività, al nome della porta di individuazione del servizio o al nome alias del client del servizio e al numero di porta dall'alias del client.

Per creare il servizio Web con Service Connect

1. Registra una definizione di attività che sia compatibile con Fargate e utilizzi la modalità di rete `awsvpc`. Completare la procedura riportata di seguito.

   a. Crea un file denominato `service-connect-nginx.json` con il contenuto della seguente definizione di attività.

      Questa definizione di attività configura Service Connect aggiungendo i parametri `name` e `appProtocol` alla mappatura delle porte. Il nome della porta rende questa porta più identificabile nella configurazione del servizio quando vengono utilizzate più porte. Il nome della porta viene utilizzato anche per impostazione predefinita come nome individuabile per essere utilizzato da altre applicazioni nello spazio dei nomi.

      La definizione dell'attività contiene il ruolo IAM dell'attività perché il servizio ha abilitato l'esecuzione ECS.

      > **Important**
      >
      > Questa definizione di attività utilizza `logConfiguration` a per inviare l'output nginx da `stdout` e `stderr` verso Amazon Logs. CloudWatch Questo ruolo di esecuzione delle attività non dispone delle autorizzazioni aggiuntive necessarie per creare il CloudWatch gruppo di log Logs. Crea il gruppo di log in CloudWatch Logs utilizzando o. AWS Management Console AWS CLI Se non desideri inviare i log di nginx a Logs, puoi rimuovere CloudWatch il. `logConfiguration`
      > Sostituisci l' Account AWS id nel ruolo di esecuzione dell'attività con il tuo id. Account AWS

```
{
```

```
    "family": "service-connect-nginx",
    "executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
    "taskRoleArn": "arn:aws:iam::123456789012:role/ecsTaskRole",
    "networkMode": "awsvpc",
    "containerDefinitions": [
        {
        "name": "webserver",
        "image": "public.ecr.aws/docker/library/nginx:latest",
        "cpu": 100,
        "portMappings": [
            {
                "name": "nginx",
                "containerPort": 80,
                "protocol": "tcp",
                "appProtocol": "http"
            }
        ],
        "essential": true,
        "logConfiguration": {
            "logDriver": "awslogs",
            "options": {
                "awslogs-group": "/ecs/service-connect-nginx",
                "awslogs-region": "region",
                "awslogs-stream-prefix": "nginx"
            }
        }
        }
    ],
    "cpu": "256",
    "memory": "512"
}
```

b. Registra la definizione di attività usando il file `service-connect-nginx.json`:

```
aws ecs register-task-definition --cli-input-json file://service-connect-
nginx.json
```

2. Crea un servizio:

a. Crea un file denominato `service-connect-nginx-service.json` con il contenuto del servizio Amazon ECS che stai per creare. Questo esempio usa la definizione di attività

creata nella fase precedente. È obbligatorio l'uso di un parametro `awsvpcConfiguration` perché la definizione di attività di esempio usa la modalità di rete `awsvpc`.

Quando crei il servizio ECS, specifica il tipo di avvio Fargate e la versione della piattaforma `LATEST` che supporta Service Connect. I `securityGroups` e le `subnets` devono appartenere a un cloud VPC che soddisfi i requisiti per l'utilizzo di Amazon ECS. Puoi ottenere gli ID del gruppo di sicurezza e della sottorete dalla console Amazon VPC.

Questo servizio configura Service Connect aggiungendo il parametro `serviceConnectConfiguration`. Lo spazio dei nomi non è richiesto perché il cluster ha uno spazio dei nomi predefinito configurato. Le applicazioni client in esecuzione in ECS nello spazio dei nomi si connettono a questo servizio utilizzando il `portName` e la porta negli `clientAliases`. Ad esempio, questo servizio è raggiungibile tramite `http://nginx:80/`, poiché nginx fornisce una pagina di benvenuto nella posizione root `/`. Le applicazioni esterne che non sono in esecuzione in Amazon ECS o che non si trovano nello stesso spazio dei nomi possono raggiungere questa applicazione tramite il proxy Service Connect utilizzando l'indirizzo IP dell'attività e il numero di porta indicato nella definizione dell'attività. Per la tua `tls` configurazione, aggiungi il certificato `arn` per il tuo `awsPcaAuthorityArn` ruolo IAM e `roleArn` per il tuo ruolo IAM. `kmsKey`

Questo servizio utilizza un `logConfiguration` per inviare l'output del proxy Service Connect da `stdout` e `stderr` verso Amazon CloudWatch Logs. Questo ruolo di esecuzione delle attività non dispone delle autorizzazioni aggiuntive necessarie per creare il gruppo di CloudWatch log Logs. Crea il gruppo di log in CloudWatch Logs utilizzando o. AWS Management Console AWS CLI Si consiglia di creare questo gruppo di log e di archiviare i log proxy in CloudWatch Logs. Se non desideri inviare i log del proxy a Logs, puoi CloudWatch rimuovere il. `logConfiguration`

```
{
    "cluster": "tutorial",
    "deploymentConfiguration": {
        "maximumPercent": 200,
        "minimumHealthyPercent": 0
    },
    "deploymentController": {
        "type": "ECS"
    },
    "desiredCount": 1,
```

```
    "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. Crea un servizio utilizzando il `service-connect-nginx-service.json` file:

```
aws ecs create-service --cluster tutorial --cli-input-json file://service-
connect-nginx-service.json
```

Output:

```
{
    "service": {
        "serviceArn": "arn:aws:ecs:us-west-2:123456789012:service/tutorial/
service-connect-nginx-service",
        "serviceName": "service-connect-nginx-service",
        "clusterArn": "arn:aws:ecs:us-west-2:123456789012:cluster/tutorial",
        "loadBalancers": [],
        "serviceRegistries": [],
        "status": "ACTIVE",
        "desiredCount": 1,
        "runningCount": 0,
        "pendingCount": 0,
        "launchType": "FARGATE",
        "platformVersion": "LATEST",
        "platformFamily": "Linux",
        "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-definition/
service-connect-nginx:1",
        "deploymentConfiguration": {
            "deploymentCircuitBreaker": {
                "enable": false,
                "rollback": false
            },
            "maximumPercent": 200,
            "minimumHealthyPercent": 0
        },
        "deployments": [
            {
                "id": "ecs-svc/3763308422771520962",
```

```
                "status": "PRIMARY",
                "taskDefinition": "arn:aws:ecs:us-west-2:123456789012:task-
definition/service-connect-nginx:1",
                "desiredCount": 1,
                "pendingCount": 0,
                "runningCount": 0,
                "failedTasks": 0,
                "createdAt": 1661210032.602,
                "updatedAt": 1661210032.602,
                "launchType": "FARGATE",
                "platformVersion": "1.4.0",
                "platformFamily": "Linux",
                "networkConfiguration": {
                    "awsvpcConfiguration": {
                        "assignPublicIp": "ENABLED",
                        "securityGroups": [
                            "sg-EXAMPLE"
                        ],
                        "subnets": [
                            "subnet-EXAMPLEf",
                            "subnet-EXAMPLE",
                            "subnet-EXAMPLE"
                        ]
                    }
                },
                "rolloutState": "IN_PROGRESS",
                "rolloutStateReason": "ECS deployment ecs-
svc/3763308422771520962 in progress.",
                "failedLaunchTaskCount": 0,
                "replacedTaskCount": 0,
                "serviceConnectConfiguration": {
                    "enabled": true,
                    "namespace": "service-connect",
                    "services": [
                        {
                            "portName": "nginx",
                            "clientAliases": [
                                {
                                    "port": 80
                                }
                            ]
                        }
                    ],
                    "logConfiguration": {
```

```
                        "logDriver": "awslogs",
                        "options": {
                            "awslogs-group": "/ecs/service-connect-proxy",
                            "awslogs-region": "us-west-2",
                            "awslogs-stream-prefix": "service-connect-proxy"
                        },
                        "secretOptions": []
                    }
                },
                "serviceConnectResources": [
                    {
                        "discoveryName": "nginx",
                        "discoveryArn": "arn:aws:servicediscovery:us-
west-2:123456789012:service/srv-EXAMPLE"
                    }
                ]
            }
        ],
        "roleArn": "arn:aws:iam::123456789012:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
        "version": 0,
        "events": [],
        "createdAt": 1661210032.602,
        "placementConstraints": [],
        "placementStrategy": [],
        "networkConfiguration": {
            "awsvpcConfiguration": {
                "assignPublicIp": "ENABLED",
                "securityGroups": [
                    "sg-EXAMPLE"
                ],
                "subnets": [
                    "subnet-EXAMPLE",
                    "subnet-EXAMPLE",
                    "subnet-EXAMPLE"
                ]
            }
        },
        "schedulingStrategy": "REPLICA",
        "enableECSManagedTags": true,
        "propagateTags": "SERVICE",
        "enableExecuteCommand": true
    }
}
```

La `serviceConnectConfiguration` fornita viene visualizzata all'interno della prima implementazione dell'output. Quando apporti modifiche al servizio ECS in modi che richiedono modifiche alle attività, Amazon ECS crea una nuova implementazione.

Fase 3: Verifica della riuscita della connessione

Per verificare che Service Connect sia configurato e funzionante, completa questa procedura per connetterti al servizio Web da un'applicazione esterna. Quindi, guarda le metriche aggiuntive nelle creazioni CloudWatch del proxy Service Connect.

Connessione al servizio Web da un'applicazione esterna

- Connessione all'indirizzo IP dell'attività e alla porta del container utilizzando l'indirizzo IP dell'attività

  Usa il AWS CLI per ottenere l'ID dell'attività, utilizzando il`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 la procedura prevede di accedere ad Amazon EC2 AWS Management Console o AWS CLI ottenere i dettagli dell'interfaccia di rete elastica.

  In questo esempio, un'istanza Amazon EC2 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 applicative (HTTP, HTTP2, gRPC o connessione TCP) nelle metriche. CloudWatch Quando usi la CloudWatch console, visualizza le dimensioni metriche

aggiuntive di DiscoveryName, (, DiscoveryName, ClusterName) ServiceName, TargetDiscovery Name e (TargetDiscoveryName, ServiceName, ClusterName) nello spazio dei nomi Amazon ECS. Per ulteriori informazioni su questi parametri e sulle dimensioni, consulta Visualizza i parametri disponibili nella Amazon CloudWatch Logs User Guide.

Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS

Il servizio Amazon ECS può facoltativamente essere configurato per l'uso della funzione di individuazione dei servizi di Amazon ECS. Service Discovery utilizza azioni AWS Cloud Map API per gestire i namespace HTTP e DNS per i tuoi servizi Amazon ECS. Per ulteriori informazioni, consulta Che cos'è AWS Cloud Map? nella Guida per gli sviluppatori di AWS Cloud Map .

L'individuazione dei servizi è disponibile nelle seguenti regioni: AWS

Nome della regione	Regione
US East (N. Virginia)	us-east-1
Stati Uniti orientali (Ohio)	us-east-2
Stati Uniti occidentali (California settentrionale)	us-west-1
US West (Oregon)	us-west-2
Africa (Cape Town)	af-south-1
Asia Pacifico (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Hyderabad)	ap-south-2
Asia Pacifico (Tokyo)	ap-northeast-1
Asia Pacifico (Seoul)	ap-northeast-2
Asia Pacifico (Osaka-Locale)	ap-northeast-3
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

Nome della regione	Regione
Asia Pacifico (Giacarta)	ap-southeast-3
Asia Pacifico (Melbourne)	ap-southeast-4
Canada (Central)	ca-central-1
Canada occidentale (Calgary)	ca-west-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
Europe (Frankfurt)	eu-central-1
Europa (Zurigo)	eu-central-2
Europa (Irlanda)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3
Europa (Milano)	eu-south-1
Europa (Stoccolma)	eu-north-1
Israele (Tel Aviv)	il-central-1
Europa (Spagna)	eu-south-2
Medio Oriente (Emirati Arabi Uniti)	me-central-1
Medio Oriente (Bahrein)	me-south-1
Sud America (São Paulo)	sa-east-1
AWS GovCloud (Stati Uniti orientali)	us-gov-east-1
AWS GovCloud (Stati Uniti occidentali)	us-gov-west-1

Concetti relativi all'individuazione dei servizi

L'individuazione dei servizi è costituita dai componenti seguenti:

- Spazio dei nomi di individuazione dei servizi: un raggruppamento logico di servizi di individuazione dei servizi che condividono lo stesso nome di dominio, ad esempio `example.com`. Questo è il nome di dominio a cui desideri instradare il traffico. Puoi creare uno spazio dei nomi con una chiamata al `aws servicediscovery create-private-dns-namespace` comando o nella console Amazon ECS. Puoi utilizzare il comando `aws servicediscovery list-namespaces` per visualizzare le informazioni di riepilogo degli spazi dei nomi creati dall'account corrente. Per ulteriori informazioni sui comandi di scoperta dei servizi, consulta `create-private-dns-namespace` e `list-namespaces` consulta la AWS Cloud Map (service discovery) AWS CLI Reference Guide.
- Servizio di individuazione dei servizi: è incluso nello spazio dei nomi di individuazione dei servizi ed è costituito dal nome del servizio e dalla configurazione DNS per lo spazio dei nomi. Fornisce il componente principale seguente:
  - Registro dei servizi: consente di cercare un servizio tramite azioni DNS o AWS Cloud Map API e recuperare uno o più endpoint disponibili che possono essere utilizzati per connettersi al servizio.
- Istanza di individuazione dei servizi: esiste all'interno del servizio di individuazione dei servizi e include gli attributi associati a ogni servizio Amazon ECS nella directory del servizio.
  - Attributi dell'istanza: i metadati seguenti vengono aggiunti come attributi personalizzati per ciascun servizio Amazon ECS configurato per l'utilizzo dell'individuazione dei servizi:
    - **`AWS_INSTANCE_IPV4`**— Per un `A` record, l'indirizzo IPv4 che Route 53 restituisce in risposta alle query DNS e AWS Cloud Map restituisce quando si scoprono i dettagli dell'istanza, ad esempio. `192.0.2.44`
    - **`AWS_INSTANCE_PORT`**: il valore della porta associato al servizio di individuazione dei servizi.
    - **`AVAILABILITY_ZONE`**: la zona di disponibilità in cui il processo è stato avviato. Per i processi che utilizzano il tipo di avvio EC2, questa è la zona di disponibilità in cui è presente l'istanza di container. Per i processi che utilizzano il tipo di avvio Fargate, questa è la zona di disponibilità in cui è presente l'interfaccia di rete elastica.
    - **`REGION`**: la regione in cui è presente il processo.
    - **`ECS_SERVICE_NAME`**: il nome del servizio Amazon ECS a cui appartiene il processo.
    - **`ECS_CLUSTER_NAME`**: il nome del cluster Amazon ECS a cui appartiene il processo.
    - **`EC2_INSTANCE_ID`**: l'ID dell'istanza di container in cui è stata posizionato il processo Questo attributo personalizzato non viene aggiunto se l'attività sta utilizzando il tipo di avvio Fargate.

  - **ECS_TASK_DEFINITION_FAMILY**: la famiglia della definizione di attività utilizzata dal processo.
  - **ECS_TASK_SET_EXTERNAL_ID**: se viene creato un set di processi per un'implementazione esterna ed è associato a un registro di individuazione dei servizi, l'attributo `ECS_TASK_SET_EXTERNAL_ID` includerà l'ID esterno del set di processi.
- Controlli dell'integrità di Amazon ECS: Amazon ECS esegue periodicamente controlli dell'integrità a livello di container. Se un endpoint non supera il controllo dello stato, viene rimosso dal routing DNS e contrassegnato come non integro.

Considerazioni relative all'individuazione dei servizi

Quando usi l'individuazione dei servizi, tieni presenti le considerazioni seguenti:

- L'individuazione dei servizi è supportata per i processi su Fargate che utilizzano la piattaforma versione 1.1.0 o successiva. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux 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, viene creata automaticamente una zona ospitata privata di Route 53.
- Per la corretta risoluzione DNS, gli attributi DNS del VPC devono essere configurati. Per ulteriori informazioni su come configurare gli attributi, consulta Supporto DNS nel VPC nella Guida per l'utente di Amazon VPC.
- I record DNS creati per un servizio DNS si registrano sempre con l'indirizzo IP privato per il processo piuttosto che l'indirizzo IP pubblico, anche quando vengono utilizzati gli spazi di nomi pubblici.
- L'individuazione dei servizi richiede che i processi specifichino la modalità di rete `awsvpc`, `bridge` o `host` (`none` non è supportata).
- Se la definizione di attività del servizio usa la modalità di rete `awsvpc`, puoi creare qualsiasi combinazione di record `A` o SRV per ciascuna attività di servizio. Se usi record SRV, è necessaria una porta.
- Se la definizione di attività di servizio usa la modalità di rete `bridge` o `host`, un record SRV è il solo tipo di record DNS supportato. Crea un record SRV per ciascuna operazione di servizio. Il

record SRV deve specificare un nome di container e una combinazione di porte di container dalla definizione di attività.

- È possibile eseguire query sui record DNS per un servizio di individuazione dei servizi all'interno del VPC. Viene usato questo formato: `<service discovery service name>.<service discovery namespace>`.
- Quando esegui una query DNS sul nome del servizio, i record `A` restituiscono un set di indirizzi IP corrispondenti alle attività. I record SRV restituiscono un set di indirizzi IP e le porte per ogni attività.
- Se dispone di otto o meno record integri, Route 53 risponde alle query DNS con tutti i record integri.
- Quando tutti i record non sono integri, Route 53 risponde alle query DNS con un massimo di otto record non integri.
- Puoi configurare il rilevamento dei servizi per un servizio protetto da un load balancer, ma il traffico del rilevamento dei servizi viene sempre instradato verso l'attività e non verso il load balancer.
- L'individuazione dei servizi non supporta l'utilizzo di Classic Load Balancer.
- Per il servizio di individuazione dei servizi consigliamo di utilizzare i controlli dell'integrità a livello dei container gestiti da Amazon ECS.
  - HealthCheckCustomConfig—Amazon ECS gestisce i controlli sanitari per tuo conto. Amazon ECS usa le informazioni ottenute dal container e dai controlli dell'integrità, insieme allo stato dell'attività, per aggiornare l'integrità con AWS Cloud Map. Questo comportamento viene specificato tramite il parametro `--health-check-custom-config` durante la creazione del servizio di individuazione dei servizi. Per ulteriori informazioni, consulta l'AWS Cloud Map API HealthCheckCustomConfigReference.
- Le AWS Cloud Map risorse create quando si utilizza il service discovery devono essere pulite manualmente.
- Le attività e le istanze vengono registrate `UNHEALTHY` fino a quando i controlli dello stato del contenitore non restituiscono un valore. Se i controlli sanitari hanno esito positivo, lo stato viene aggiornato a`HEALTHY`. Se i controlli di integrità del contenitore falliscono, l'istanza di rilevamento del servizio viene annullata.

Prezzi del servizio di individuazione dei servizi

I clienti che utilizzano l'individuazione dei servizi di Amazon ECS pagano per le risorse Route 53 e le operazioni API di individuazione AWS Cloud Map . Sono inclusi i costi per la creazione delle zone

ospitate di Route 53 e per le query nel registro del servizio. Per ulteriori informazioni, consulta Prezzi di AWS Cloud Map nella Guida per gli sviluppatori di AWS Cloud Map .

Amazon ECS esegue controlli di integrità a livello di container e li espone a operazioni API di controllo dello stato AWS Cloud Map personalizzate. Questa opzione al momento viene fornita gratuitamente ai clienti. Se configuri controlli dell'integrità di rete aggiuntivi per le attività esposte pubblicamente, potrebbero esserti addebitati i relativi costi.

Creazione di un servizio Amazon ECS che utilizza Service Discovery

Scopri come creare un servizio contenente un'attività Fargate che utilizza il rilevamento dei servizi con. AWS CLI

Per un elenco della scoperta di Regioni AWS tale servizio di supporto, vedereUsa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.

Per informazioni sulle regioni che supportano Fargate, consulta the section called “AWS Regioni di Fargate”.

Prerequisiti

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

- La versione più recente di AWS CLI è installata e configurata. Per ulteriori informazioni, consulta l'argomento relativo all'installazione di AWS Command Line Interface.
- Le fasi descritte in Configurazione per l'uso di Amazon ECS sono complete.
- AWS L'utente dispone delle autorizzazioni richieste specificate nell'esempio di policy AmazonECS_FullAccess IAM.
- Hai creato almeno un VPC e un gruppo di sicurezza. Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.

Fase 1: Creare le risorse Service Discovery in AWS Cloud Map

Utilizza la procedura seguente per creare lo spazio dei nomi di individuazione dei servizi e il servizio di individuazione dei servizi.

1. Crea uno spazio dei nomi privato di individuazione dei servizi Cloud Map. In questo esempio viene creato uno spazio dei nomi denominato `tutorial`. Sostituisci *`vpc-abcd1234`* con l'ID di uno dei VPC esistenti.

```
aws servicediscovery create-private-dns-namespace \
      --name tutorial \
      --vpc vpc-abcd1234
```

L'output di questo comando è il seguente:

```
{
    "OperationId": "h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-je6chs2e"
}
```

2. Usando `OperationId` dall'output della fase precedente, verifica che lo spazio dei nomi privato sia stato creato correttamente. Prendi nota dell'ID dello spazio dei nomi perché lo utilizzerai nei comandi successivi.

```
aws servicediscovery get-operation \
      --operation-id h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-je6chs2e
```

L'output è il seguente.

```
{
    "Operation": {
        "Id": "h2qe3s6dxftvvt7riu6lfy2f6c3jlhf4-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": "httpd:2.4",
                "portMappings": [
                    {
                        "containerPort": 80,
                        "hostPort": 80,
                        "protocol": "tcp"
                    }
                ],
                "essential": true,
                "entryPoint": [
                    "sh",
                    "-c"
                ],
                "command": [
                    "/bin/sh -c \"echo '<html> <head> <title>Amazon ECS Sample
 App</title> <style>body {margin-top: 40px; background-color: #333;} </style>
 </head><body> <div style=color:white;text-align:center> <h1>Amazon ECS Sample
 App</h1> <h2>Congratulations!</h2> <p>Your application is now running on a
 container in Amazon ECS.</p> </div></body></html>' >  /usr/local/apache2/
htdocs/index.html && httpd-foreground\""
                ]
            }
        ],
        "requiresCompatibilities": [
            "FARGATE"
        ],
        "cpu": "256",
        "memory": "512"
}
```

b. Registrazione della definizione dell'attività tramite `fargate-task.json`.

```
aws ecs register-task-definition \
      --cli-input-json file://fargate-task.json
```

3. Crea un servizio ECS completando la seguente procedura:

a. Creazione di un file denominato `ecs-service-discovery.json` con il contenuto del servizio ECS che stai per creare. Questo esempio usa la definizione di attività creata nella fase precedente. È obbligatorio l'uso di un parametro `awsvpcConfiguration` perché la definizione di attività di esempio usa la modalità di rete `awsvpc`.

   Quando crei il servizio ECS, specifica il tipo di avvio Fargate e la versione della piattaforma `LATEST` che supporta l'individuazione dei servizi. Quando viene creato il servizio di individuazione dei servizi in AWS Cloud Map , l'ARN restituito è `registryArn`. `securityGroups` e `subnets` devono appartenere al VPC utilizzato per creare lo spazio dei nomi Cloud Map. Puoi ottenere gli ID del gruppo di sicurezza e della sottorete 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
```

Passaggio 3: Verifica Service Discovery in AWS Cloud Map

Puoi verificare che sia stato creato tutto correttamente eseguendo una query sulle informazioni di individuazione dei servizi. Dopo aver configurato l'individuazione del servizio, puoi utilizzare le operazioni AWS Cloud Map API o effettuare una chiamata `dig` da un'istanza all'interno del tuo VPC. Completare la procedura riportata di seguito.

1. Usando l'ID servizio di individuazione dei servizi, elenca le istanze di individuazione dei servizi. Prendi nota dell'ID istanza (contrassegnato in grassetto) per la pulizia delle risorse.

   ```
   aws servicediscovery list-instances \
         --service-id srv-utcrh6wavdkggqtk
   ```

   L'output è il seguente.

   ```
   {
       "Instances": [
           {
               "Id": "16becc26-8558-4af1-9fbd-f81be062a266",
               "Attributes": {
                   "AWS_INSTANCE_IPV4": "172.31.87.2"
                   "AWS_INSTANCE_PORT": "80",
                   "AVAILABILITY_ZONE": "us-east-1a",
                   "REGION": "us-east-1",
                   "ECS_SERVICE_NAME": "ecs-service-discovery",
                   "ECS_CLUSTER_NAME": "tutorial",
                   "ECS_TASK_DEFINITION_FAMILY": "tutorial-task-def"
               }
           }
       ]
   }
   ```

2. Utilizza lo spazio dei nomi, il servizio di individuazione dei servizi e parametri aggiuntivi quali il nome del cluster ECS per ottenere i dettagli sulle istanze di individuazione dei servizi.

   ```
   aws servicediscovery discover-instances \
         --namespace-name tutorial \
         --service-name myapplication \
         --query-parameters ECS_CLUSTER_NAME=tutorial
   ```

3. Puoi eseguire query sui record DNS creati nella zona ospitata di Route 53 per il servizio di individuazione dei servizi con i seguenti comandi AWS CLI :

   a. Con l'ID spazio dei nomi, ottieni le informazioni sullo spazio dei nomi che includono l'ID della zona ospitata di Route 53.

   ```
   aws servicediscovery \
         get-namespace --id ns-uejictsjen2i4eeg
   ```

   L'output è il seguente.

   ```
   {
       "Namespace": {
           "Id": "ns-uejictsjen2i4eeg",
           "Arn": "arn:aws:servicediscovery:region:aws_account_id:namespace/ns-
   uejictsjen2i4eeg",
           "Name": "tutorial",
           "Type": "DNS_PRIVATE",
           "Properties": {
                "DnsProperties": {
                   "HostedZoneId": "Z35JQ4ZFDRYPLV"
               }
           },
           "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 Z35JQ4ZFDRYPLV
   ```

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"
}
```

2. 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 xhu73bsertlyffhm3faqi7kumsmx274n-jh0zimzv
```

```
{
  "Operation": {
        "Id": "xhu73bsertlyffhm3faqi7kumsmx274n-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"
        }
    }
}
```

3. Eliminazione del servizio di individuazione dei servizi utilizzando l'ID servizio.

```
aws servicediscovery delete-service \
      --id srv-utcrh6wavdkggqtk
```

4. Eliminazione dello spazio dei nomi di individuazione dei servizi utilizzando l'ID spazio dei nomi.

```
aws servicediscovery delete-namespace \
      --id ns-uejictsjen2i4eeg
```

L'output è il seguente.

```
{
    "OperationId": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj"
}
```

5. Usando l'`OperationId` dall'output della fase precedente, verifica che lo spazio dei nomi di individuazione dei servizi sia stato eliminato correttamente.

```
aws servicediscovery get-operation \
      --operation-id c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj
```

L'output è il seguente.

```
{
    "Operation": {
        "Id": "c3ncqglftesw4ibgj5baz6ktaoh6cg4t-jh0ztysj",
        "Type": "DELETE_NAMESPACE",
        "Status": "SUCCESS",
        "CreateDate": 1525984602.211,
        "UpdateDate": 1525984602.558,
        "Targets": {
            "NAMESPACE": "ns-rymlehshst7hhukh",
            "ROUTE_53_CHANGE_ID": "CJP2A2M86XW3O"
        }
    }
}
```

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
```

Proteggi le tue attività di Amazon ECS dall'interruzione a causa di eventi di scalabilità

Puoi utilizzare la protezione scale-in delle attività di Amazon ECS per evitare che le tue attività vengano interrotte da eventi di scalabilità in base all'auto scalabilità del servizio o alle distribuzioni.

Alcune applicazioni richiedono un meccanismo per salvaguardare le attività strategiche dall'interruzione dovuta a eventi scalabili nei periodi di scarso utilizzo o durante le implementazioni dei servizi. Per esempio:

- Si tratta di un'applicazione asincrona di elaborazione delle code, ad esempio un processo di transcodifica video in cui alcune attività devono essere eseguite per ore anche quando l'utilizzo cumulativo del servizio è basso.
- Hai un'applicazione di gioco che esegue server di gioco come attività Amazon ECS che devono continuare a funzionare anche se tutti gli utenti si sono disconnessi per ridurre la latenza di avvio di un riavvio del server.
- Quando si implementa una nuova versione del codice, è necessario che le attività continuino a essere eseguite perché sarebbe costoso rielaborarle.

Per evitare che le attività appartenenti al servizio vengano terminate in caso di un evento di dimensionamento, imposta l'attributo `protectionEnabled` su `true`. Per impostazione predefinita, le attività sono protette per 2 ore. 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).

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.

Meccanismi di protezione scalabile delle attività

Puoi impostare e ottenere una protezione scalabile delle attività utilizzando l'endpoint dell'agente container Amazon ECS o l'API Amazon ECS.

- Endpoint dell'agente del container Amazon ECS

  Consigliamo di utilizzare l'endpoint dell'agente container Amazon ECS per attività che possono determinare autonomamente la necessità di essere protette. Utilizza questo approccio per carichi di lavoro basati su code o di elaborazione dei processi

  Quando un container inizia il lavoro di elaborazione, ad esempio consumando un messaggio SQS, è possibile impostare l'attributo `ProtectionEnabled` tramite il percorso dell'endpoint di protezione scalabile dell'attività `$ECS_AGENT_URI/task-protection/v1/state` dall'interno del container. Amazon ECS non terminerà questa attività durante gli eventi di scalabilità. Una volta che l'attività ha terminato il suo funzionamento, puoi cancellare l'`ProtectionEnabled`attributo utilizzando lo stesso endpoint, in modo da renderla idonea alla cessazione durante i successivi eventi di scalabilità.

  Per ulteriori informazioni sull'endpoint Amazon ECS Container Agent, consulta. Endpoint di protezione scalabile per attività Amazon ECS
- API Amazon ECS

  Puoi utilizzare l'API Amazon ECS per impostare e recuperare la protezione scalabile delle attività se l'applicazione dispone di un componente che monitora lo stato delle attività attive. Utilizza `UpdateTaskProtection` per contrassegnare una o più attività come protette. Utilizza per recuperare `GetTaskProtection` lo stato di protezione.

  Un esempio di questo approccio potrebbe essere se l'applicazione ospita sessioni del server di gioco come attività Amazon ECS. Quando un utente accede a una sessione sul server (attività), è possibile contrassegnare l'attività come protetta. Dopo la disconnessione dell'utente, puoi annullare

la protezione specifica per l'attività o annullare periodicamente la protezione per attività simili che non hanno più sessioni attive, a seconda se è necessario mantenere i server inattivi.

Per ulteriori informazioni, consulta UpdateTaskProtezione e GetTaskprotezione nel riferimento all'API di Amazon Elastic Container Service.

È possibile combinare entrambi gli approcci. Ad esempio, utilizza l'endpoint dell'agente Amazon ECS per impostare la protezione delle attività dall'interno di un container e utilizza l'API Amazon ECS per rimuovere la protezione delle attività dal servizio di controller esterno.

Considerazioni

Considera i seguenti punti prima di utilizzare la protezione scalabile delle attività:

- Ti consigliamo di utilizzare l'endpoint dell'agente del container Amazon ECS, in quanto l'agente Amazon ECS dispone di meccanismi di ripetizione integrati e di un'interfaccia più semplice.
- Puoi reimpostare il periodo di scadenza della protezione per il dimensionamento delle attività richiamando `UpdateTaskProtection` su un'attività che ha già la protezione attivata.
- Determina il tempo necessario a un'attività per completare il lavoro richiesto e imposta la proprietà `expiresInMinutes` di conseguenza. Se imposti la scadenza della protezione più a lungo del necessario, dovrai sostenere costi e ritardi nell'implementazione di nuove attività.
- La protezione per il dimensionamento delle attività è supportata sull'agente del container Amazon ECS `1.65.0` o versioni successive.

  Questa caratteristica può essere supportata sulle istanze Amazon EC2 utilizzando versioni precedenti dell'agente di container Amazon ECS aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta Aggiornamento dell'agente del container Amazon ECS.
- Considerazioni sull'implementazione:
  - Se il servizio utilizza un aggiornamento in sequenza, verranno create nuove attività ma le attività che eseguono una versione precedente non verranno interrotte fino a quando `protectionEnabled` non viene annullato o scade. È possibile regolare il parametro `maximumPercentage` nella configurazione di implementazione su un valore che consenta di creare nuove attività quando le vecchie attività sono protette.
  - Se viene applicato un aggiornamento blu/verde, l'implementazione blu contenente le attività protette non verrà rimossa se le attività presentano l'opzione `protectionEnabled`. Il traffico verrà dirottato verso le nuove attività che si presentano e le attività precedenti verranno rimosse solo quando `protectionEnabled` vengono cancellate o scadute. A seconda del timeout degli

CloudFormation aggiornamenti, la CodeDeploy distribuzione potrebbe scadere e le attività Blue precedenti potrebbero essere ancora presenti.

- Se lo utilizzi CloudFormation, lo update-stack ha un timeout di 3 ore. Pertanto, se imposti la protezione delle attività per più di 3 ore, la CloudFormation distribuzione potrebbe causare errori e ripristini.

  Durante il periodo in cui le vecchie attività sono protette, lo CloudFormation stack viene visualizzato. `UPDATE_IN_PROGRESS` Se la protezione per il dimensionamento delle attività viene rimossa o scade entro la finestra di 3 ore, l'implementazione avrà esito positivo e passerà allo stato `UPDATE_COMPLETE`. Se l'implementazione rimane bloccata in `UPDATE_IN_PROGRESS` per più di 3 ore, non riuscirà e mostrerà lo stato `UPDATE_FAILED`, quindi verrà ripristinato il vecchio set di attività.
- Amazon ECS invia gli eventi di servizio se le attività protette impediscono a un'implementazione (in sequenza o blu/verde) di raggiungere lo stato stazionario, in modo che sia possibile intraprendere azioni correttive. Durante il tentativo di aggiornare lo stato di protezione di un'attività, se viene ricevuto un messaggio di errore `DEPLOYMENT_BLOCKED`, allora significa che il servizio ha più attività protette rispetto al numero di attività desiderato per il servizio. Per risolvere questo errore, effettuare una delle seguenti operazioni:
  - Attendi la scadenza della protezione dell'attività corrente. Quindi imposta la protezione delle attività.
  - Determina quali attività possono essere arrestate. Quindi, usa `UpdateTaskProtection` con l'opzione `protectionEnabled` impostata su `false` per queste attività.
  - Aumenta il numero di attività desiderate del servizio portandolo a un numero maggiore del numero di attività protette.

Autorizzazioni IAM richieste per la protezione scalabile delle attività

L'attività deve avere il ruolo di attività di Amazon ECS con le seguenti autorizzazioni:

- `ecs:GetTaskProtection`: consente all'agente del container Amazon ECS di effettuare chiamate `GetTaskProtection`.
- `ecs:UpdateTaskProtection`: consente all'agente del container Amazon ECS di effettuare chiamate `UpdateTaskProtection`.

Endpoint di protezione scalabile per attività Amazon ECS

L'agente del container Amazon ECS inserisce automaticamente la variabile di ambiente `ECS_AGENT_URI` nei container delle attività di Amazon ECS per fornire un metodo per interagire con l'endpoint API dell'agente del container.

Consigliamo di utilizzare l'endpoint dell'agente container Amazon ECS per attività che possono determinare autonomamente la necessità di essere protette.

Quando un contenitore inizia a elaborare il lavoro, è possibile impostare l'`protectionEnabled`attributo utilizzando il percorso `$ECS_AGENT_URI/task-protection/v1/state` dell'endpoint di protezione scale-in dell'attività dall'interno del contenitore.

Utilizzate una richiesta PUT a questo URI dall'interno di un contenitore per impostare la protezione scalabile delle attività. Una richiesta GET a questo URI restituisce lo stato di protezione corrente di un'attività.

Parametri della richiesta di protezione scalabile in base alla dimensione dell'attività

È possibile impostare la protezione scalabile delle attività utilizzando l'endpoint `${ECS_AGENT_URI}/task-protection/v1/state` con i seguenti parametri di richiesta.

`ProtectionEnabled`

`true`Specificare di contrassegnare un'attività per la protezione. `false`Specificare se rimuovere la protezione e rendere l'attività idonea alla cessazione.

Tipo: Booleano

Campo obbligatorio: sì

`ExpiresInMinutes`

Il numero di minuti in cui l'attività è protetta. È possibile specificare da un minimo di 1 minuto a un massimo di 2.880 minuti (48 ore). Durante questo periodo di tempo, l'attività non verrà terminata da eventi di dimensionamento derivanti dal servizio Dimensionamento automatico o dalle implementazioni. Trascorso questo periodo di tempo, il parametro `protectionEnabled` viene impostato su `false`.

Se non si specifica l'ora, l'attività viene protetta automaticamente per 120 minuti (2 ore).

Tipo: integer

Campo obbligatorio: no

Gli esempi seguenti mostrano come impostare la protezione delle attività con durate diverse.

Esempio di come proteggere un'attività con il periodo di tempo predefinito

Questo esempio mostra come proteggere un'attività con il periodo di tempo predefinito di 2 ore.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-
protection/v1/state --data '{"ProtectionEnabled":true}'
```

Esempio di protezione di un'attività per 60 minuti

Questo esempio mostra come proteggere un'attività per 60 minuti utilizzando il parametro `expiresInMinutes`.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-
protection/v1/state --data '{"ProtectionEnabled":true,"ExpiresInMinutes":60}'
```

Esempio di protezione di un'attività per 24 ore

Questo esempio mostra come proteggere un'attività per 24 ore utilizzando il parametro `expiresInMinutes`.

```
curl --request PUT --header 'Content-Type: application/json' ${ECS_AGENT_URI}/task-
protection/v1/state --data '{"ProtectionEnabled":true,"ExpiresInMinutes":1440}'
```

La richiesta PUT restituisce la seguente risposta.

```
{
  "protection": {
    "ExpirationDate": "2023-12-20T21:57:44.837Z",
    "ProtectionEnabled": true,
    "TaskArn": "arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"
  }
}
```

Parametri di risposta alla protezione scalabili in base alle attività

La risposta in formato JSON dell'endpoint di protezione per il dimensionamento delle attività `${ECS_AGENT_URI}/task-protection/v1/state` restituisce le seguenti informazioni.

`ExpirationDate`

L'ora in cui scadrà la protezione per l'attività. Se l'attività non è protetta, questo valore è nullo.

`ProtectionEnabled`

Lo stato di protezione dell'attività. Se la protezione scalabile è abilitata per un'attività, il valore è `true`. In caso contrario è `false`.

`TaskArn`

L'Amazon Resource Name (ARN) completo dell'attività a cui appartiene il container.

L'esempio che segue mostra i dettagli restituiti per un'attività protetta.

```
curl --request GET ${ECS_AGENT_URI}/task-protection/v1/state
```

```
{
    "protection":{
        "ExpirationDate":"2023-12-20T21:57:44Z",
        "ProtectionEnabled":true,
        "TaskArn":"arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0"
    }
}
```

Le seguenti informazioni vengono restituite quando si verifica un errore.

`Arn`

Il nome della risorsa Amazon (ARN) completo dell'attività.

`Detail`

I dettagli relativi all'errore.

`Reason`

Il motivo dell'errore.

L'esempio che segue mostra i dettagli restituiti per un'attività che non è protetta.

```
{
```

```
    "failure":{
        "Arn":"arn:aws:ecs:us-west-2:111122223333:task/1234567890abcdef0",
        "Detail":null,
        "Reason":"TASK_NOT_VALID"
    }
}
```

Le seguenti informazioni vengono restituite quando si verifica un'eccezione.

`requestID`

L'ID della AWS richiesta per la chiamata API Amazon ECS che genera un'eccezione.

`Arn`

Il nome della risorsa Amazon (ARN) completo dell'attività o del servizio.

`Code`

Il codice di errore.

`Message`

Messaggio di errore.

> **Note**
>
> Se viene visualizzato un errore `RequestError` o `RequestTimeout`, è probabile che si tratti di un problema di rete. Prova a utilizzare gli endpoint VPC per Amazon ECS.

L'esempio che segue mostra i dettagli restituiti quando si verifica un errore.

```
{
    "requestID":"12345-abc-6789-0123-abc",
    "error":{
        "Arn":"arn:aws:ecs:us-west-2:555555555555:task/my-cluster-
name/1234567890abcdef0",
        "Code":"AccessDeniedException",
        "Message":"User: arn:aws:sts::444455556666:assumed-role/my-ecs-task-
role/1234567890abcdef0 is not authorized to perform: ecs:GetTaskProtection on resource:
 arn:aws:ecs:us-west-2:555555555555:task/test/1234567890abcdef0 because no identity-
based policy allows the ecs:GetTaskProtection action"
    }
```

```
}
```

L'errore seguente viene visualizzato se l'agente Amazon ECS non è in grado di ottenere una risposta dall'endpoint Amazon ECS per motivi quali problemi di rete o il piano di controllo Amazon ECS inattivo.

```
{
  "error": {
    "Arn": "arn:aws:ecs:us-west-2:555555555555:task/my-cluster-name/1234567890abcdef0",
    "Code": "RequestCanceled",
    "Message": "Timed out calling Amazon ECS Task Protection API"
  }
}
```

Il seguente errore si verifica quando l'agente Amazon ECS riceve un'eccezione di limitazione da Amazon ECS.

```
{
  "requestID": "12345-abc-6789-0123-abc",
  "error": {
    "Arn": "arn:aws:ecs:us-west-2:555555555555:task/my-cluster-name/1234567890abcdef0",
    "Code": "ThrottlingException",
    "Message": "Rate exceeded"
  }
}
```

Logica di accelerazione del servizio Amazon ECS

Il pianificatore di servizi Amazon ECS ora include la logica che limita la frequenza dei tentativi di riavvio dei processi in caso di avvii non riusciti ripetuti.

Se le attività relative a un servizio non rientrano ripetutamente nello `RUNNING` stato impostato (passando direttamente da uno `STOPPED` stato `PENDING` a a uno), il tempo che intercorre tra i successivi tentativi di riavvio viene aumentato in modo incrementale fino a un massimo di 27 minuti. Questo periodo massimo è soggetto a modifiche in futuro. Questo comportamento riduce l'effetto delle attività fallite nelle risorse del cluster Amazon ECS o sui costi dell'infrastruttura Fargate. Se il servizio attiva la logica di limitazione, riceverai il seguente messaggio di evento relativo al servizio:

```
(service service-name) is unable to consistently start tasks successfully.
```

Amazon ECS non impedisce mai a un servizio guasto di riprovare. Inoltre, non prova a modificarlo in alcun modo se non aumentando il tempo tra i riavvii. La logica di limitazione del servizio non fornisce parametri regolabili dall'utente.

Se aggiorni il servizio per l'utilizzo di una nuova definizione di attività, il servizio torna immediatamente a uno stato normale non limitato. Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console.

Di seguito sono riportate alcune cause comuni che avviano questa logica. Ti consigliamo di intraprendere un'azione manuale per risolvere il problema:

- Una mancanza di risorse che possono ospitare l'attività, quali porte, memoria o unità di CPU nel cluster. In questo caso, viene anche visualizzato il messaggio di evento relativo al servizio delle risorse insufficienti.
- L'agente del container di Amazon ECS non è in grado di recuperare l'immagine Docker dell'attività. Ciò potrebbe essere dovuto al nome di un'immagine del container non corretto, a un'immagine o un tag non corretti o alla mancanza di autenticazione o delle autorizzazioni dei registri privati. In questo caso, viene anche visualizzato `CannotPullContainerError` negli errori di attività interrotte.
- Lo spazio su disco insufficiente nell'istanza di container per poter creare il container. In questo caso, viene anche visualizzato `CannotCreateContainerError` negli errori di attività interrotte. Per ulteriori informazioni, consulta Risolvi i problemi relativi al Docker in Amazon `API error (500): devmapper` ECS.

> **Important**
>
> Le attività interrotte dopo che hanno raggiunto lo stato `RUNNING` non attivano la logica di limitazione o il messaggio di evento relativo al servizio associato. Ad esempio, se i controlli dell'integrità di Elastic Load Balancing non riusciti per un servizio fanno sì che un processo venga contrassegnato come non integro e Amazon ECS annulla la registrazione dell'attività e la interrompe. A questo punto, le attività non vengono limitate. Anche se il comando del container di un'attività termina immediatamente con un codice di uscita diverso zero, l'attività è già passata allo stato `RUNNING`. Le attività che non riescono immediatamente a causa di errori di comando non provocano la limitazione o il messaggio di evento relativo al servizio.

Parametri di definizione del servizio Amazon ECS

Una definizione di servizio definisce come eseguire il servizio Amazon ECS. I seguenti parametri possono essere specificati in una definizione di servizio.

Tipo di avvio

`launchType`

- Tipo: stringa

  Valori validi: `EC2` | `FARGATE` | `EXTERNAL`

  Campo obbligatorio: no

  Il tipo di avvio con cui eseguire il servizio. Se non viene specificato un tipo di avvio, per impostazione predefinita viene utilizzato il `capacityProviderStrategy` di default. Per ulteriori informazioni, consulta Tipi di avvio di Amazon ECS.

  Se viene specificato un `launchType`, il parametro `capacityProviderStrategy` deve essere omesso.

Strategia del provider di capacità

`capacityProviderStrategy`

  Tipo: matrice di oggetti

  Campo obbligatorio: no

  La strategia del fornitore di capacità da utilizzare per il servizio.

  Una strategia del fornitore di capacità consiste di uno o più fornitori di capacità e dei valori `base` e `weight` da assegnare a essi. Un provider di capacità deve essere associato al cluster da utilizzare in una strategia del provider di capacità. L' PutClusterCapacityProviders API viene utilizzata per associare un provider di capacità a un cluster. È possibile utilizzare solo i provider di capacità con uno stato `ACTIVE` o uno stato `UPDATING`.

  Se viene specificato un `capacityProviderStrategy`, il parametro `launchType` deve essere omesso. Se non viene specificato `capacityProviderStrategy` o `launchType`, viene utilizzato `defaultCapacityProviderStrategy` per il cluster.

Se desideri specificare un provider di capacità che utilizzi un gruppo con scalabilità automatica, il provider di capacità deve essere già stato creato. È possibile creare nuovi fornitori di capacità con l'operazione CreateCapacityProvider API.

Per utilizzare un provider di capacità AWS Fargate, specificare `FARGATE` o i provider di `FARGATE_SPOT` capacità. I provider di capacità AWS Fargate sono disponibili per tutti gli account e devono essere associati solo a un cluster per essere utilizzati.

L'operazione PutClusterCapacityProviders API viene utilizzata per aggiornare l'elenco dei fornitori di capacità disponibili per un cluster dopo la creazione del cluster.

`capacityProvider`

Tipo: stringa

Campo obbligatorio: sì

Il nome breve o il nome della risorsa Amazon (ARN) del provider di capacità.

`weight`

Tipo: integer

Intervallo valido: numeri interi compresi tra 0 e 1.000.

Campo obbligatorio: no

Il valore del peso indica la percentuale relativa del numero totale di attività avviate che devono utilizzare il provider di capacità specificato.

Ad esempio, supponiamo di avere una strategia che contiene due provider di capacità, ognuno con un peso pari a uno. Una volta soddisfatta la base, le attività si dividono equamente tra i due provider di capacità. Usando la stessa logica, assumiamo di specificare un peso di 1 per capacityProviderA e un peso di 4 per capacityProviderB. Quindi, per ogni attività che viene eseguita utilizzando capacityProviderA, quattro attività utilizzano capacityProviderB.

`base`

Tipo: integer

Intervallo valido: numeri interi compresi tra 0 e 100.000.

Campo obbligatorio: no

Il valore di base indica il numero minimo di attività da eseguire nel provider di capacità specificato. Solo un provider di capacità in una strategia di provider di capacità può avere una base definita.

Definizione di attività

`taskDefinition`

- Tipo: stringa

Campo obbligatorio: no

I valori `family` e `revision` (`family:revision`) o il nome della risorsa Amazon (ARN) completo della definizione di attività da eseguire nel servizio. Se non viene specificato un `revision`, viene utilizzata l'ultima revisione `ACTIVE` della famiglia specificata.

Una definizione di attività deve essere specificata quando si utilizza il controller di distribuzione di aggiornamento in sequenza (`ECS`).

Sistema operativo della piattaforma

`platformFamily`

Tipo: stringa

Obbligatorio: condizionale

Di default: Linux

Questo parametro è richiesto per i servizi Amazon ECS ospitati su Fargate.

Questo parametro viene ignorato per i servizi Amazon ECS ospitati su Amazon EC2.

Il sistema operativo sui container che esegue il servizio. I valori validi sono `LINUX`, `WINDOWS_SERVER_2019_FULL`, `WINDOWS_SERVER_2019_CORE`, `WINDOWS_SERVER_2022_FULL` e `WINDOWS_SERVER_2022_CORE`.

Il valore `platformFamily` per ogni processo specificato per il servizio deve corrispondere al valore `platformFamily` del servizio. Ad esempio, se hai impostato `platformFamily` su

`WINDOWS_SERVER_2019_FULL`, il valore `platformFamily` per tutte le attività deve essere `WINDOWS_SERVER_2019_FULL`.

Versione della piattaforma

`platformVersion`

- Tipo: stringa

Campo obbligatorio: no

La versione della piattaforma su cui sono in esecuzione le attività nel servizio. Viene specificata una versione della piattaforma solo per le attività con tipo di avvio Fargate. Se non è specificata, la versione più recente (`LATEST`) viene utilizzata di default.

AWS Le versioni della piattaforma Fargate vengono utilizzate per fare riferimento a un ambiente di runtime specifico per l'infrastruttura di attività Fargate. Quando specifichi la versione della piattaforma `LATEST` durante l'esecuzione di un'attività o la creazione di un servizio, ottieni la versione di piattaforma più aggiornata disponibile per le tue attività. Quando incrementi il servizio, tali attività riceveranno la versione della piattaforma specificata nell'implementazione corrente del servizio. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.

> **Note**
>
> Le versioni della piattaforma non sono specificate per i processi che utilizzano il tipo di avvio EC2.

Cluster

`cluster`

- Tipo: stringa

Campo obbligatorio: no

Il nome breve o il nome della risorsa Amazon (ARN) completo del cluster in cui eseguire il servizio. Se non specifichi un cluster, viene utilizzato il cluster di `default`.

Nome servizio

`serviceName`

Tipo: stringa

Campo obbligatorio: sì

Il nome del servizio. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. I nomi dei servizi devono essere univoci all'interno di un cluster, ma puoi avere servizi dai nomi simili in più cluster all'interno di una Regione o in più Regioni.

Strategia di pianificazione

`schedulingStrategy`

-Tipo: stringa

Valori validi: `REPLICA` | `DAEMON`

Campo obbligatorio: no

La strategia di pianificazione da utilizzare. Se non viene specificata alcuna strategia di pianificazione, viene utilizzata la strategia `REPLICA`. Per ulteriori informazioni, consulta Servizi Amazon ECS.

Sono disponibili due strategie del pianificatore del servizio:

- `REPLICA`: la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le decisione riguardo al posizionamento delle attività. Per ulteriori informazioni, consulta Strategia di replica.
- `DAEMON`: la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Quando si utilizza questa strategia, non è necessario specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzare le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta Strategia daemon.

> **Note**
>
> I processi Fargate non supportano la strategia di pianificazione `DAEMON`

Conteggio desiderato

`desiredCount`

Tipo: integer

Campo obbligatorio: no

Il numero di istanze della definizione di attività specificata da posizionare e mantenere in esecuzione nel servizio.

Questo parametro è obbligatorio se si utilizza la strategia di pianificazione `REPLICA`. Se il servizio utilizza la strategia di pianificazione `DAEMON`, questo parametro è facoltativo.

Configurazione dell'implementazione

`deploymentConfiguration`

Tipo: oggetto

Campo obbligatorio: no

I parametri di distribuzione opzionali che determinano quante attività vengono eseguite durante l'implementazione e l'ordine di arresto e di avvio delle attività.

`maximumPercent`

Tipo: integer

Campo obbligatorio: no

Se il servizio utilizza il tipo di implementazione con aggiornamento in sequenza (`ECS`), il parametro `maximumPercent` rappresenta un limite superiore sul numero di attività del servizio che sono consentite nello stato `RUNNING`, `STOPPING` o `PENDING` durante

un'implementazione. Viene espresso come percentuale del `desiredCount`arrotondato per difetto al valore intero più vicino. Questo parametro può essere utilizzato per definire le dimensioni del batch di implementazione. Ad esempio, se il servizio utilizza il pianificatore di servizi `REPLICA` e ha un `desiredCount` di quattro attività e un valore `maximumPercent` pari al 200%, il pianificatore può avviare 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%.

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 sta utilizzando i tipi di distribuzione blu/verde (`CODE_DEPLOY`) o `EXTERNAL` con attività che utilizzano il tipo di avvio EC2, il valore della percentuale massima viene impostato sul valore predefinito e utilizzato per definire il limite superiore per il numero di attività nel servizio che rimangono nello stato `RUNNING` mentre le istanze di container sono in stato `DRAINING`. Se le attività del servizio utilizzano il tipo di avvio Fargate, il valore della percentuale massima non viene utilizzato anche se viene restituito nella descrizione del servizio.

`minimumHealthyPercent`

Tipo: integer

Campo obbligatorio: no

Se il servizio utilizza il tipo di implementazione aggiornamento in sequenza (`ECS`), il parametro `minimumHealthyPercent` rappresenta un limite inferiore sul numero di attività del servizio che devono restare nello stato `RUNNING` durante un'implementazione. Viene espresso come percentuale del `desiredCount`arrotondato per eccesso al valore intero più vicino. Questo parametro può essere utilizzato per eseguire l'implementazione senza utilizzare capacità aggiuntiva del cluster. Ad esempio, se il servizio ha un `desiredCount` di quattro attività e un `minimumHealthyPercent` del 50%, il pianificatore di servizi può arrestare due attività esistenti per liberare capacità del cluster prima di avviare due nuove attività.

Per i servizi che non utilizzano un load balancer, considera quanto riportato di seguito:

- Un servizio è considerato integro se tutti i container essenziali all'interno delle attività del servizio superano i controlli dello stato.
- Se un'attività non dispone di container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà 40 secondi dopo che un'attività raggiunge uno stato `RUNNING` prima di contarla per la percentuale minima di integrità totale.
- Se un'attività ha uno o più container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà che l'attività raggiunga uno stato integro prima di contarla per la percentuale minima di integrità totale. Un'attività è considerata sana quando tutti i container essenziali al suo interno hanno superato i controlli dello stato. La quantità di tempo che il pianificatore del servizio può attendere è determinata dalle impostazioni di controllo dello stato del container. Per ulteriori informazioni, consulta Controllo dello stato.

Per i servizi che utilizzano un load balancer, considera quanto riportato di seguito:

- Se un'attività non dispone di container essenziali con un controllo dell'integrità definito, il pianificatore del servizio attenderà che il controllo dell'integrità del gruppo di destinazione del load balancer restituisca uno stato integro prima di contarla nella percentuale minima di integrità totale.
- Se un'attività dispone di un container essenziale con un controllo dell'integrità definito, il pianificatore del servizio attenderà che l'attività raggiunga uno stato integro e il controllo dell'integrità del gruppo di destinazione del load balancer restituisca uno stato integro prima di contare l'attività nella percentuale minima di integrità totale.

Il valore predefinito di un servizio di replica per `minimumHealthyPercent` è 100%. Il `minimumHealthyPercent` valore predefinito per un servizio che utilizza la pianificazione del `DAEMON` servizio è 0% per AWS CLI gli AWS SDK e le API e il 50% per. AWS Management Console

Il numero minimo di attività integre durante un'implementazione è uguale a `desiredCount` moltiplicato per `minimumHealthyPercent`/100, arrotondato per eccesso al numero intero più vicino.

Se un servizio sta utilizzando i tipi di implementazione blu/verde (`CODE_DEPLOY`) o `EXTERNAL` con processi in esecuzione che utilizzano il tipo di avvio EC2, il valore della percentuale minima di integrità viene impostato sul valore di default e utilizzato per definire il limite inferiore per il numero di processi nel servizio che rimangono nello stato `RUNNING` mentre le istanze di container sono in stato `DRAINING`. Se un servizio utilizza i tipi di implementazione blu/verde (`CODE_DEPLOY`) o `EXTERNAL` ed esegue processi che utilizzano il tipo di avvio Fargate, il

valore della percentuale minima di integrità non viene utilizzato, anche se viene restituito nella descrizione del servizio.

Controller di implementazione

`deploymentController`

Tipo: oggetto

Campo obbligatorio: no

Il tipo di controller di distribuzione da utilizzare per il servizio. Se non viene specificato alcun controller di implementazione, viene utilizzato il controller `ECS`. Per ulteriori informazioni, consulta Servizi Amazon ECS.

`type`

- Tipo: stringa

Valori validi: `ECS` | `CODE_DEPLOY` | `EXTERNAL`

Obbligatorio: sì

Il tipo di controller di distribuzione da utilizzare. Esistono tre tipi di controller di distribuzione:

`ECS`

Il tipo di distribuzione aggiornamento in sequenza (`ECS`) prevede la sostituzione della versione attualmente in esecuzione del container con quella più recente. Il numero di container che Amazon ECS aggiunge o rimuove dal servizio durante un aggiornamento in sequenza è controllato regolando il numero minimo e massimo di integrità consentite durante l'implementazione di un servizio, come specificato in deploymentConfiguration.

`CODE_DEPLOY`

Il tipo di distribuzione blue/green (`CODE_DEPLOY`) utilizza il modello di distribuzione blue/green powered by CodeDeploy, che consente di verificare una nuova implementazione di un servizio prima di inviargli il traffico di produzione.

`EXTERNAL`

Utilizza il tipo di implementazione esterna quando desideri usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implementazione per un servizio Amazon ECS.

Posizionamento dei processi

`placementConstraints`

Tipo: matrice di oggetti

Campo obbligatorio: no

Serie di oggetti vincolo di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di 10 vincoli per attività. Questo limite include i vincoli nella definizione dell'attività e quelli specificati in fase di runtime. Se utilizzi il tipo di avvio Fargate, i vincoli di posizionamento delle attività non sono supportati.

`type`

- Tipo: stringa

  Campo obbligatorio: no

  Il tipo di vincolo. Utilizza `distinctInstance` per accertarti che ogni attività in un determinato gruppo sia in esecuzione su un'istanza di container diversa. Utilizza `memberOf` per limitare la selezione a un gruppo di candidati validi. Il valore `distinctInstance` non è supportato nelle definizioni di attività.

`expression`

- Tipo: stringa

  Campo obbligatorio: no

  Un'espressione del linguaggio di query del cluster da applicare al vincolo. Non puoi specificare un'espressione se il tipo di vincolo è `distinctInstance`. Per ulteriori informazioni, consulta Crea espressioni per definire istanze di container per le attività di Amazon ECS.

`placementStrategy`

Tipo: matrice di oggetti

Campo obbligatorio: no

Gli oggetti strategia di posizionamento da utilizzare per le attività del servizio. Puoi specificare un massimo di quattro regole di strategia per ogni servizio.

`type`

- Tipo: stringa

  Valori validi: `random` | `spread` | `binpack`

  Campo obbligatorio: no

  Il tipo di strategia di posizionamento. La strategia di posizionamento `random` posiziona in modo casuale le attività sui candidati disponibili. La strategia di posizionamento `spread` distribuisce le attività in modo uniforme tra i candidati disponibili in base al parametro `field`. La strategia `binpack` posiziona le attività sui candidati disponibili che dispongono della quantità minima di risorsa specificata con il parametro `field`. Ad esempio, se adotti il binpacking della memoria, un'attività verrà posizionata sull'istanza con la quantità minima di memoria residua (ma ancora sufficiente a eseguire l'attività).

`field`

- Tipo: stringa

  Campo obbligatorio: no

  Il campo a cui applicare la strategia di posizionamento. Per la strategia di posizionamento `spread`, i valori validi sono `instanceId` (o `host`, che ha lo stesso effetto) oppure qualsiasi attributo personalizzato o di piattaforma applicato a un'istanza di container, come `attribute:ecs.availability-zone`. Per la strategia di posizionamento `binpack`, i valori validi sono `cpu` e `memory`. Per la strategia di posizionamento `random` questo campo non viene utilizzato.

Tag

`tags`

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati applicati al servizio per aiutarti a catalogarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili. Quando un servizio viene eliminato, vengono eliminati anche i tag. Al servizio è possibile applicare un massimo di 50 tag. Per ulteriori informazioni, consulta Taggare le risorse Amazon ECS.

`key`

- Tipo: stringa

  Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

  Campo obbligatorio: no

  Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

`value`

- Tipo: stringa

  Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

  Campo obbligatorio: no

  La parte facoltativa di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

`enableECSManagedTags`

Tipo: Booleano

Valori validi: `true` | `false`

Campo obbligatorio: no

Specifica se usare i tag gestiti di Amazon ECS per i processi nel servizio. Se non viene specificato alcun valore, il valore predefinito è `false`. Per ulteriori informazioni, consulta Utilizza i tag per la fatturazione.

`propagateTags`

- Tipo: stringa

Valori validi: `TASK_DEFINITION` | `SERVICE`

Campo obbligatorio: no

Specifica se copiare i tag dalla definizione di attività o dal servizio nelle attività del servizio. Se non viene specificato alcun valore, i tag non vengono copiati. I tag possono essere copiati solo nelle

attività all'interno del servizio durante la creazione del servizio. Per aggiungere tag a un processo dopo la creazione di un servizio o di un processo, utilizza l'operazione API `TagResource`.

Configurazione della rete

`networkConfiguration`

Tipo: oggetto

Campo obbligatorio: no

La configurazione di rete per il servizio. Questo parametro è obbligatorio per le definizioni di attività che utilizzano la modalità di rete `awsvpc` per ricevere la propria interfaccia di rete elastica e non è supportato per altre modalità. Con il tipo di avvio Fargate, la modalità di rete `awsvpc` è obbligatoria. Per ulteriori informazioni sul networking per il tipo di lancio di Amazon EC2, consulta. Opzioni di task networking di Amazon ECS per il tipo di lancio EC2 Per ulteriori informazioni sulla rete per il tipo di lancio Fargate, vedere Fargate Task Networking.

`awsvpcConfiguration`

Tipo: oggetto

Campo obbligatorio: no

Un oggetto che rappresenta le sottoreti e i gruppi di sicurezza per un'attività o un servizio.

`subnets`

Tipo: matrice di stringhe

Campo obbligatorio: sì

Le sottoreti associate all'attività o al servizio. Esiste un limite di 16 sottoreti che possono essere specificate in base a `awsvpcConfiguration`.

`securityGroups`

Tipo: matrice di stringhe

Campo obbligatorio: no

I gruppi di sicurezza associati all'attività o al servizio. Se non specifichi un gruppo di sicurezza, viene utilizzato il gruppo di sicurezza predefinito dell'ambiente VPC.

Esiste un limite di cinque gruppi di sicurezza che possono essere specificati in base a `awsvpcConfiguration`.

`assignPublicIP`

- Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Se l'interfaccia di rete elastica dell'attività riceve un indirizzo IP pubblico. Se non viene specificato alcun valore, viene utilizzato il valore predefinito `DISABLED`.

`healthCheckGracePeriodSeconds`

Tipo: integer

Campo obbligatorio: no

Il periodo di tempo, in secondi, durante il quale il pianificatore di servizi di Amazon ECS deve ignorare i controlli dell'integrità delle destinazioni Elastic Load Balancing non integre, i controlli dell'integrità del container e i controlli dell'integrità di Route 53 dopo che un processo entra nella stato `RUNNING`. Ciò è valido solo se il tuo servizio è configurato per utilizzare un load balancer. Se il servizio ha definito un load balancer e non si specifica un valore del periodo di tolleranza del controllo integrità, viene utilizzato il valore predefinito di `0`.

Se le attività del servizio non si avviano subito e rispondono ai controlli dello stato, puoi specificare un periodo di grazia per il controllo dello stato di un massimo di 2,147,483,647 secondi, durante i quali il pianificatore del servizio ECS ignora lo stato di questi controlli. Questo periodo di grazia può evitare che il pianificatore del servizio ECS contrassegni le attività come non integre e le interrompa prima che abbiano il tempo di iniziare.

Se non utilizzi un Elastic Load Balancing, ti consigliamo di utilizzare `startPeriod` nei parametri di controllo dell'integrità di definizione del processo. Per ulteriori informazioni, consulta Determinare lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container.

`loadBalancers`

Tipo: matrice di oggetti

Campo obbligatorio: no

Un oggetto che rappresenta il load balancer da utilizzare con il tuo servizio. Per i servizi che utilizzano un Application Load Balancer o un Network Load Balancer, esiste un limite di cinque gruppi di destinazione che puoi collegare a un servizio.

Dopo aver creato un servizio, la configurazione del load balancer non può essere modificata dalla AWS Management Console. Puoi utilizzare AWS Copilot AWS CLI o SDK per modificare la configurazione del bilanciamento del carico solo per il controller di distribuzione ECS mobile, non blu/verde o esterno. AWS CloudFormation AWS CodeDeploy Quando aggiungi, aggiorni o rimuovi una configurazione del load balancer, Amazon ECS avvia una nuova implementazione con la configurazione aggiornata di Elastic Load Balancing. Questo causa la registrazione e l'annullamento della registrazione dai load balancer. Si consiglia di verificarlo in un ambiente di test prima di aggiornare la configurazione di Elastic Load Balancing. Per informazioni su come modificare la configurazione, consulta il riferimento UpdateServiceall'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

  Campo obbligatorio: no

  Il nome della risorsa Amazon (ARN) completo del gruppo o dei gruppi di destinazione dell'Elastic Load Balancer associati a un servizio.

  L'ARN del gruppo target viene specificato solo quando di utilizza un Application Load Balancer o un Network Load Balancer.

`loadBalancerName`

- Tipo: stringa

  Campo obbligatorio: no

  Il nome del load balancer da associare al servizio.

  Se utilizzi un Application Load Balancer o un Network Load Balancer, il parametro relativo al nome del load balancer deve essere omesso.

`containerName`

- Tipo: stringa

Campo obbligatorio: no

Il nome del container (così come appare nella definizione di quest'ultimo) da associare al load balancer.

`containerPort`

Tipo: integer

Campo obbligatorio: no

La porta del container da associare al load balancer. Questa porta deve corrispondere a un `containerPort` nella definizione di attività utilizzata dalle attività nel servizio. Per le attività che utilizzano il tipo di avvio EC2, l'istanza di container deve consentire il traffico in ingresso sulla `hostPort` della mappatura delle porte.

`role`

- Tipo: stringa

Campo obbligatorio: no

Il nome breve o l'ARN completo del ruolo IAM che consente ad Amazon ECS di effettuare chiamate al load balancer per tuo conto. Questo parametro è consentito solo se utilizzi un load balancer con un singolo gruppo di destinazione e se la definizione di attività non utilizza la modalità di rete `awsvpc`. Se specifichi il parametro `role`, devi anche specificare un oggetto load balancer con il parametro `loadBalancers`.

Se il ruolo specificato ha un percorso diverso da `/`, devi specificare l'ARN del ruolo completo (scelta consigliata) o anteporre il percorso al nome del ruolo come prefisso. Ad esempio, se un ruolo con il nome `bar` ha il percorso `/foo/`, devi specificare `/foo/bar` come nome del ruolo. Per ulteriori informazioni, consulta Nomi descrittivi e percorsi nella Guida per l'utente IAM.

> **Important**
>
> Se il tuo account ha già creato il ruolo collegato al servizio Amazon ECS, tale ruolo viene utilizzato di default per il tuo servizio, a meno che tu non specifichi un ruolo qui. Il ruolo

collegato al servizio è obbligatorio se la definizione di attività utilizza la modalità di rete awsvpc; in tal caso, non è necessario specificare un ruolo qui. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.

`serviceConnectConfiguration`

Tipo: oggetto

Campo obbligatorio: no

La configurazione di questo servizio per rilevare e connettersi ai servizi, ed essere rilevati e connessi da altri servizi all'interno di uno spazio nomi.

Per ulteriori informazioni, consulta Usa Service Connect per connettere i servizi Amazon ECS con nomi brevi.

`enabled`

Tipo: Booleano

Campo obbligatorio: sì

Specifica se utilizzare Service Connect con questo servizio.

`namespace`

-Tipo: stringa

Campo 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. Lo spazio dei nomi deve trovarsi nella stessa Regione AWS del cluster e del servizio Amazon ECS. Il tipo di spazio dei nomi non influisce su Service Connect. Per ulteriori informazioni AWS Cloud Map, consulta Working with Services nella Developer Guide.AWS Cloud Map

`services`

Tipo: matrice di oggetti

Campo obbligatorio: no

Un array degli oggetti del servizio Service Connect. Questi sono nomi e alias (detti anche endpoint) utilizzati da altri servizi Amazon ECS per connettersi a questo servizio.

Questo campo non è obbligatorio per un servizio Amazon ECS "client" che un membro di uno spazio dei nomi utilizza solo per connettersi ad altri servizi nello spazio dei nomi. Un esempio può essere un'applicazione front-end che accetta richieste in arrivo da un sistema di bilanciamento del carico collegato al servizio o con altri mezzi.

Un oggetto seleziona una porta dalla definizione dell'attività, assegna un nome al AWS Cloud Map servizio e una serie di alias (noti anche come endpoint) e porte alle applicazioni client per fare riferimento a questo servizio.

`portName`

Tipo: stringa

Campo obbligatorio: sì

`portName` deve corrispondere al `name` di una delle `portMappings` da tutti i container nella definizione di attività di questo servizio Amazon ECS.

`discoveryName`

-Tipo: stringa

Campo obbligatorio: no

`discoveryName`È il nome del nuovo AWS Cloud Map servizio creato da Amazon ECS per questo servizio Amazon ECS. Deve essere univoco nello spazio dei nomi AWS Cloud Map .

Se questo campo non è specificato, viene utilizzato `portName`.

`clientAliases`

Tipo: matrice di oggetti

Campo obbligatorio: no

L'elenco degli alias client per questo servizio Service Connect. Si usano per assegnare nomi che possono essere utilizzati da applicazioni client. Il numero massimo di alias client che si può avere in questo elenco è 1.

Ogni alias ("endpoint") è un nome DNS completo e un numero di porta che altre attività Amazon ECS ("client") possono utilizzare per connettersi a questo servizio.

Ogni combinazione di nome e porta deve essere univoca nello spazio dei nomi.

Questi nomi sono configurati all'interno di ogni attività del servizio client, non in AWS Cloud Map. Le richieste DNS per risolvere questi nomi non lasciano l'attività e non vengono conteggiate ai fini della quota di richieste DNS al secondo per interfaccia di rete elastica.

`port`

Tipo: integer

Campo obbligatorio: sì

Il numero della porta di ascolto per il proxy Service Connect. Questa porta è disponibile all'interno di tutte le attività nello stesso spazio dei nomi.

Per evitare di modificare le applicazioni nei servizi client Amazon ECS, imposta la stessa porta utilizzata dall'applicazione client per impostazione predefinita.

`dnsName`

- Tipo: stringa

Campo obbligatorio: no

Il `dnsName` è il nome utilizzato nelle applicazioni di attività client per connettersi a questo servizio. Il nome deve essere un'etichetta DNS valida.

Se questo campo non viene specificato, il valore predefinito sarà `discoveryName.namespace`. Se il valore `discoveryName` non viene specificato, viene utilizzato il valore `portName` della definizione dell'attività.

Per evitare di modificare le applicazioni nei servizi client Amazon ECS, imposta lo stesso nome utilizzato dall'applicazione client per impostazione predefinita. Ad esempio, alcuni nomi comuni sono `database`, `db` o il nome minuscolo di un database, ad esempio `mysql` o `redis`.

`ingressPortOverride`

Tipo: integer

Campo obbligatorio: no

(Facoltativo) Il numero della porta di ascolto per il proxy Service Connect.

Utilizza il valore di questo campo per ignorare il proxy per il traffico sul numero di porta specificato nella `portMapping` denominata nella definizione di attività di questa applicazione, quindi utilizzalo nei tuoi gruppi di sicurezza Amazon VPC per consentire il traffico nel proxy per questo servizio Amazon ECS.

In modalità `awsvpc`, il valore predefinito è il numero di porta del container specificato nella `portMapping` nella definizione di attività di questa applicazione. In modalità `bridge`, il valore predefinito è la porta temporanea dinamica del proxy Service Connect.

`logConfiguration`

Tipo: oggetto LogConfiguration

Campo obbligatorio: no

Questo definisce dove vengono pubblicati i log del proxy di Service Connect. Usa i log per il debug durante eventi imprevisti. Questa configurazione imposta il parametro `logConfiguration` nel container del proxy Service Connect in ogni attività di questo servizio Amazon ECS. Il container del proxy non è specificato nella definizione di attività.

Ti consigliamo di utilizzare la stessa configurazione di log dei container delle applicazioni della definizione di attività per questo servizio Amazon ECS. Infatti FireLens, questa è la configurazione del registro del contenitore dell'applicazione. Non è il contenitore del FireLens log router che utilizza l'immagine del `fluentd` contenitore `fluent-bit` or.

`serviceRegistries`

Tipo: matrice di oggetti

Campo obbligatorio: no

I dettagli della configurazione dell'individuazione per il tuo servizio. Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.

`registryArn`

- Tipo: stringa

Campo obbligatorio: no

Il nome della risorsa Amazon (ARN) del registro del servizio. Il registro dei servizi attualmente supportato è AWS Cloud Map. Per ulteriori informazioni, consulta Utilizzo dei servizi nella Guida per gli sviluppatori di AWS Cloud Map .

`port`

Tipo: integer

Campo obbligatorio: no

Il valore della porta utilizzato se l'individuazione dei servizi ha specificato un record SRV. Questo campo è obbligatorio se vengono utilizzati entrambi la modalità di rete `awsvpc` e i record SRV.

`containerName`

-Tipo: stringa

Campo obbligatorio: no

Il valore che indica il nome del container da utilizzare per l'individuazione del servizio. Questo valore è specificato nella definizione dell'attività. Se la definizione di attività specificata dall'attività del servizio utilizza le modalità di rete `bridge` o `host`, devi specificare una combinazione di `containerName` e `containerPort` nella definizione di attività. Se la definizione di attività specificata dall'attività del servizio utilizza la modalità di rete `awsvpc` e utilizzi un record tipo DNS SRV, devi specificare una combinazione di `containerName` e `containerPort` o un valore `port`, ma non entrambi.

`containerPort`

Tipo: integer

Campo obbligatorio: no

Il valore della porta da utilizzare per l'individuazione del servizio. Questo valore è specificato nella definizione dell'attività. Se la definizione di attività specificata dall'attività del servizio utilizza le modalità di rete `bridge` o `host`, devi specificare una combinazione di `containerName` e `containerPort` nella definizione di attività. Se la definizione di attività specificata dall'attività del servizio utilizza la modalità di rete `awsvpc` e utilizzi un record tipo DNS SRV, devi specificare una combinazione di `containerName` e `containerPort` o un valore `port`, ma non entrambi.

Token client

`clientToken`

- Tipo: stringa

Campo obbligatorio: no

L'identificatore univoco (con distinzione tra maiuscole e minuscole) che fornisci per assicurare l'idempotenza della richiesta. Può contenere fino a 32 caratteri ASCII.

Configurazioni dei volumi

`volumeConfigurations`

Tipo: oggetto

Campo obbligatorio: no

La configurazione che verrà utilizzata per creare volumi per le attività gestite dal servizio. Viene creato un volume per ogni attività del servizio. Solo i volumi contrassegnati come `configuredAtLaunch` nella definizione dell'attività possono essere configurati utilizzando questo oggetto. Questo oggetto è necessario per collegare i volumi di dati di Amazon EBS alle attività gestite da un servizio. Per ulteriori informazioni, consulta Volumi Amazon EBS.

`name`

Tipo: stringa

Campo obbligatorio: sì

Il nome di un volume configurato durante la creazione o l'aggiornamento di un servizio. Sono consentite fino a 255 lettere (maiuscole e minuscole), numeri, caratteri di sottolineatura (_) e trattini (). - Questo valore deve corrispondere al nome del volume specificato nella definizione dell'attività.

`managedEBSVolume`

Tipo: oggetto

Campo obbligatorio: no

La configurazione del volume per i volumi Amazon EBS collegati alle attività gestite da un servizio quando un servizio viene creato o aggiornato.

`encrypted`

Tipo: Booleano

Campo obbligatorio: no

Valori validi: `true|false`

Speciifica se il volume Amazon EBS collegato alle attività gestite da un servizio verrà crittografato. Se hai attivato la crittografia Amazon EBS per impostazione predefinita per il tuo account, questa impostazione verrà sovrascritta e il volume verrà crittografato. Per ulteriori informazioni sulla crittografia EBS per impostazione predefinita, consulta Encryption by default nella Amazon EC2 User Guide.

`kmsKeyId`

-Tipo: stringa

Campo obbligatorio: no

L'identificatore della chiave AWS Key Management Service (AWS KMS) da utilizzare per la crittografia Amazon EBS. Se questo parametro non è specificato, viene utilizzato il tuo AWS KMS key per Amazon EBS. Se `KmsKeyId` viene specificata, lo stato crittografato deve essere `true`.

Puoi specificare la chiave KMS utilizzando uno dei seguenti metodi:

- ID chiave: ad esempio,`1234abcd-12ab-34cd-56ef-1234567890ab`.
- Alias chiave: ad esempio,`alias/ExampleAlias`.
- ARN chiave: ad esempio,. `arn:aws:kms:us-east-1:012345678910:key/1234abcd-12ab-34cd-56ef-1234567890ab`
- Alias ARN: ad esempio,. `arn:aws:kms:us-east-1:012345678910:alias/ExampleAlias`

> **Important**
>
> AWS autentica la chiave KMS in modo asincrono. Pertanto, se si specifica un ID, un alias o un ARN non valido, l'azione può sembrare riuscita, ma alla fine non

riesce. Per ulteriori informazioni, consulta Risoluzione dei problemi relativi ai volumi allegati di Amazon EBS.

`volumeType`

-Tipo: stringa

Campo obbligatorio: no

Valori validi: `gp2` | `gp3` | `io1` | | `io2` | `sc1` | `st1` `standard`

Il tipo di volume EBS. Per ulteriori informazioni sui tipi di volume, consulta i tipi di volume di Amazon EBS nella Guida per l'utente di Amazon EC2. Il tipo di volume di default è `gp3`.

> **Note**
>
> Il tipo di `standard` volume non è supportato per i volumi Amazon EBS configurati per il collegamento alle attività di Fargate.

`sizeInGiB`

Tipo: integer

Campo obbligatorio: no

Intervallo valido: numeri interi compresi tra 1 e 16.384

La dimensione del volume EBS in gibibyte (GiB). Se non fornisci un ID snapshot per configurare un volume per gli allegati, devi fornire un valore di dimensione. Se si configura un volume per l'allegato utilizzando un'istantanea, il valore predefinito è la dimensione dell'istantanea. È quindi possibile specificare una dimensione maggiore o uguale alla dimensione dell'istantanea.

Per `gp2` i tipi di `gp3` volume, l'intervallo valido è 1-16.384.

Per i tipi `io1` di `io2` volume, l'intervallo valido è 4-16.384.

Per i tipi `st1` di `sc1` volume, l'intervallo valido è 125-16.384.

Per il tipo di `standard` volume, l'intervallo valido è 1-1.024.

`snapshotId`

- Tipo: stringa

Campo obbligatorio: no

L'ID dell'istantanea di un volume EBS esistente utilizzato per creare un nuovo volume collegato all'attività ECS.

`iops`

Tipo: integer

Campo obbligatorio: no

Il numero di operazioni I/O al secondo (IOPS). Per volumi `gp3`, `io1` e `io2`, questo rappresenta il numero di IOPS che sono assegnate al volume. Per `gp2` i volumi, questo valore rappresenta le prestazioni di base del volume e la velocità con cui il volume accumula crediti di I/O per il bursting. Questo parametro è obbligatorio per i volumi `io1` e `io2`. Questo parametro non è supportato per`gp2`,`st1`, `sc1` o volumi. `standard`

Per `gp3` i volumi, l'intervallo di valori valido è compreso tra 3.000 e 16.000.

Per `io1` i volumi, l'intervallo di valori valido è compreso tra 100 e 64.000.

Per `io2` i volumi, l'intervallo di valori valido è compreso tra 100 e 64.000.

`throughput`

Tipo: integer

Campo obbligatorio: no

La velocità effettiva necessaria per il provisioning dei volumi collegati alle attività gestite da un servizio.

> **Important**
>
> Questo parametro è supportato solo per i `gp3` volumi.

`roleArn`

Tipo: stringa

Campo obbligatorio: sì

Il ruolo Amazon Resource ARN (ARN) dell'infrastruttura AWS Identity and Access Management (IAM) che fornisce le autorizzazioni Amazon ECS per gestire le risorse Amazon EBS per le tue attività. Per ulteriori informazioni, consulta Ruolo IAM dell'infrastruttura Amazon ECS.

`tagSpecifications`

Tipo: oggetto

Campo obbligatorio: no

La specifica per i tag da applicare ai volumi Amazon EBS gestiti dal servizio.

`resourceType`

Tipo: stringa

Campo obbligatorio: sì

Valori validi: `volume`

Il tipo di risorsa cui applicare tag al momento della creazione.

`tags`

Tipo: matrice di oggetti

Campo obbligatorio: no

I metadati che applichi ai volumi per aiutarti a classificarli e organizzarli. Ogni tag è composto da una chiave e da un valore opzionale, entrambi definiti dall'utente. `AmazonECSCreated` e `AmazonECSManaged` sono tag riservati aggiunti da Amazon ECS per tuo conto, quindi puoi specificare un massimo di 48 tag personalizzati. Quando un volume viene eliminato, vengono eliminati anche i tag. Per ulteriori informazioni, consulta Taggare le risorse Amazon ECS.

`key`

- Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 128 caratteri.

Campo obbligatorio: no

Una parte di una coppia chiave-valore che costituisce un tag. Una chiave è un'etichetta generale che funge da categoria per più valori di tag specifici.

`value`

- Tipo: stringa

Limitazioni di lunghezza: lunghezza minima di 0. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

La parte opzionale di una coppia chiave-valore che costituisce un tag. Un valore agisce come descrittore all'interno di una categoria di tag (chiave).

`propagateTags`

- Tipo: stringa

Valori validi: `TASK_DEFINITION` | `SERVICE` | `NONE`

Campo obbligatorio: no

Speciifica se copiare i tag dalla definizione dell'attività o dal servizio in un volume. Se `NONE` è specificato o non viene specificato alcun valore, i tag non vengono copiati.

`fileSystemType`

- Tipo: stringa

Campo obbligatorio: no

Valori validi: `xfs`|`ext3`|`ext4`

Il tipo di file system su un volume. Il tipo di file system del volume determina il modo in cui i dati vengono archiviati e recuperati nel volume. Per i volumi creati da un'istantanea, è necessario specificare lo stesso tipo di file system utilizzato dal volume al momento della creazione dell'istantanea. Se il tipo di file system non corrisponde, l'operazione non verrà avviata. L'impostazione predefinita per i volumi collegati alle attività di Linux è. `XFS`

Modello di definizione del servizio

Di seguito viene illustrata la rappresentazione JSON di una definizione di servizio Amazon ECS.

Tipo di lancio di Amazon 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,
    "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": "",
                "iops": 0,
                "throughput": 0,
                "tagSpecifications": [
                    {
                        "resourceType": "volume",
                        "tags": [
                            {
                                "key": "",
```

```
                                "value": ""
                            }
                        ],
                        "propagateTags": "NONE"
                    }
                ],
                "roleArn": "",
                "filesystemType": ""
            }
        }
    ]
}
```

Tipo di avvio di Fargate

```
{
    "cluster": "",
    "serviceName": "",
    "taskDefinition": "",
    "loadBalancers": [
        {
            "targetGroupArn": "",
            "loadBalancerName": "",
            "containerName": "",
            "containerPort": 0
        }
    ],
    "serviceRegistries": [
        {
            "registryArn": "",
            "port": 0,
            "containerName": "",
            "containerPort": 0
        }
    ],
    "desiredCount": 0,
    "clientToken": "",
    "launchType": "FARGATE",
    "capacityProviderStrategy": [
        {
            "capacityProvider": "",
            "weight": 0,
            "base": 0
```

```
        }
    ],
    "platformVersion": "",
    "platformFamily": "",
    "role": "",
    "deploymentConfiguration": {
        "deploymentCircuitBreaker": {
            "enable": true,
            "rollback": true
        },
        "maximumPercent": 0,
        "minimumHealthyPercent": 0,
        "alarms": {
            "alarmNames": [
                ""
            ],
            "enable": true,
            "rollback": true
        }
    },
    "placementStrategy": [
        {
            "type": "binpack",
            "field": ""
        }
    ],
    "networkConfiguration": {
        "awsvpcConfiguration": {
            "subnets": [
                ""
            ],
            "securityGroups": [
                ""
            ],
            "assignPublicIp": "DISABLED"
        }
    },
    "healthCheckGracePeriodSeconds": 0,
    "schedulingStrategy": "REPLICA",
    "deploymentController": {
        "type": "EXTERNAL"
    },
    "tags": [
        {
```

```
            "key": "",
            "value": ""
        }
    ],
    "enableECSManagedTags": true,
    "propagateTags": "TASK_DEFINITION",
    "enableExecuteCommand": true,
    "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": "",
                "iops": 0,
                "throughput": 0,
                "tagSpecifications": [
                    {
                        "resourceType": "volume",
                        "tags": [
                            {
                                "key": "",
                                "value": ""
                            }
                        ],
                        "propagateTags": "NONE"
                    }
                ],
                "roleArn": "",
                "filesystemType": ""
            }
        }
    ]
}
```

È possibile creare questo modello di definizione del servizio utilizzando il comando AWS CLI seguente.

```
aws ecs create-service --generate-cli-skeleton
```

Taggare le risorse Amazon ECS

Per semplificare la gestione delle risorse Amazon ECS, è possibile assegnare metadati personalizzati a ciascuna risorsa sotto forma di tag. Ciascun tag è formato da una chiave e da un valore facoltativo.

Puoi utilizzare i tag per categorizzare le risorse Amazon ECS in vari modi, ad esempio per scopo, proprietario o ambiente. Questa funzione è utile quando si dispone di numerose risorse dello stesso tipo. Puoi identificare velocemente una risorsa specifica in base ai tag a questa assegnati. Ad esempio, puoi definire un set di tag per le istanze del container Amazon ECS del tuo account. Questo ti aiuta a monitorare il proprietario dell'istanza e il livello dello stack.

Puoi utilizzare i tag per i report su costi e utilizzo. Puoi utilizzare questi report per analizzare il costo e l'utilizzo delle tue risorse di Amazon ECS. Per ulteriori informazioni, consulta the section called “Report di utilizzo di”.

> **Warning**
>
> Esistono molte API che restituiscono le chiavi dei tag e i relativi valori. Negando l'accesso a `DescribeTags` non viene automaticamente negato l'accesso ai tag restituiti da altre API. Come best practice, consigliamo di non includere dati sensibili nei tag.

Ti consigliamo di creare un set di chiavi di tag in grado di soddisfare i requisiti di ciascun tipo di risorsa. Tramite un set di chiavi di tag coerente la gestione delle risorse risulta notevolmente semplificata. Puoi cercare e filtrare le risorse in base ai tag aggiunti.

I tag non hanno alcun significato semantico per Amazon ECS e vengono interpretati rigorosamente come una stringa di caratteri. Puoi modificare chiavi e valori di tag e rimuovere tag da una risorsa in qualsiasi momento. È possibile impostare il valore di un tag su una stringa vuota, ma non su null. Se a una risorsa specifica aggiungi un tag con la stessa chiave di un tag esistente, il nuovo valore sovrascriverà quello precedente. Quando si elimina una risorsa, vengono eliminati anche tutti i tag relativi alla risorsa.

Se utilizzi AWS Identity and Access Management (IAM), puoi controllare quali utenti del tuo AWS account sono autorizzati a gestire i tag.

Assegnazione di tag alle risorse

Esistono diversi modi per assegnare tag ad attività, servizi, definizioni di attività e cluster Amazon ECS:

- Un utente tagga manualmente una risorsa utilizzando l' AWS Management Console API Amazon ECS AWS CLI, o un AWS SDK.
- Un utente crea un servizio o esegue un'attività autonoma e seleziona l'opzione dei tag gestiti da Amazon ECS.

  Amazon ECS assegna automaticamente i tag a tutte le attività appena avviate. Per ulteriori informazioni, consulta the section called “Tag gestiti da Amazon ECS”.
- Un utente crea una risorsa utilizzando la console. La console assegna automaticamente i tag alle risorse.

  Questi tag vengono restituiti nelle AWS CLI risposte e AWS SDK e vengono visualizzati nella console. Non è possibile modificare o eliminare questi tag.

  Per informazioni sui tag aggiunti, consulta la colonna Tag aggiunti automaticamente dalla console nella tabella Supporto del tagging per le risorse Amazon ECS.

Se specifichi dei tag quando crei una risorsa e i tag non possono essere applicati, Amazon ECS esegue il rollback del processo di creazione. Ciò fa sì che le risorse vengano create con i tag oppure che non vengano create affatto, nonché che nessuna risorsa sia mai sprovvista di tag. L'aggiunta di tag alle risorse in fase di creazione consente di evitare di eseguire script di tagging personalizzati dopo la creazione delle risorse.

La tabella seguente descrive le risorse Amazon ECS che supportano il tagging.

Supporto del tagging per le risorse Amazon ECS

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Attività di Amazon ECS	Sì	Sì, dalla definizione di attività.	Chiave: `aws:ecs:clusterName`

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
			Value (Valore): `cluster-name`
Servizi Amazon ECS	Sì	Sì, dalla definizione di attività o dal servizio alle attività del servizio.	Chiave: `ecs:service:stackId` Value `arn:aws:cloudformation: arn`
Set di processi di Amazon ECS	Sì	No	N/D
Definizione dei processi di Amazon ECS	Sì	No	Chiave: `ecs:taskDefinition:createdFrom` Value (Valore): `ecs-console-v2`

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Cluster Amazon ECS	Sì	No	Chiave: `aws:cloudformation:logical-id` Value (Valore): `ECSCluster` Chiave: `aws:cloudformation:stack-id` Value (Valore): `arn:aws:cloudformation: arn` Chiave: `aws:cloudformation:stack-name` Value (Valore): `ECS-Console-V2-Cluster- EXAMPLE`
Istanze di container di Amazon ECS	Sì	Sì, dall'istanza Amazon EC2. Per ulteriori informazioni, consulta Aggiungere tag a un'istanza di container Amazon ECS.	N/D
Istanze esterne di Amazon ECS	Sì	No	N/D

Risorsa	Supporta tag	Supporta la propagazione di tag	Tag aggiunti automaticamente dalla console
Provider di capacità Amazon ECS	Sì. Non è possibile applicare tag ai provider di capacità predefiniti `FARGATE` e `FARGATE_SPOT` .	No	N/D

Applicazione di tag alle risorse durante la creazione

Le seguenti risorse supportano l'etichettatura alla creazione utilizzando l'API Amazon ECS o AWS CLI AWS SDK:

- Attività di Amazon ECS
- Servizi Amazon ECS
- Definizione dell'attività di Amazon ECS
- Set di processi di Amazon ECS
- Cluster Amazon ECS
- Istanze di container di Amazon ECS
- Provider di capacità Amazon ECS

Amazon ECS offre la possibilità di utilizzare l'autorizzazione all'assegnazione di tag per la creazione di risorse. Quando Account AWS è configurato per l'autorizzazione all'etichettatura, gli utenti devono disporre delle autorizzazioni per le azioni che creano la risorsa, ad esempio. `ecsCreateCluster` Se specifichi i tag nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare se gli utenti o i ruoli dispongono delle autorizzazioni per creare tag. Pertanto, devi concedere le autorizzazioni esplicite per utilizzare l'operazione `ecs:TagResource`. Per ulteriori informazioni, consulta the section called “Assegnazione di tag alle risorse al momento della creazione”. Per informazioni su come configurare l'opzione, consulta. the section called “Autorizzazione all'assegnazione di tag”

Restrizioni

Ai tag si applicano le limitazioni seguenti:

- È possibile associare un massimo di 50 tag a una risorsa.
- Le chiavi dei tag non possono essere ripetute per una risorsa. Ogni chiave di tag deve essere univoca e può avere un solo valore.
- Le chiavi di tag possono contenere fino a 128 caratteri in UTF-8.
- Ogni valore può contenere fino a 256 caratteri UTF-8.
- Se più risorse Servizi AWS e utilizzano lo schema di etichettatura, limita i tipi di caratteri che usi. Alcuni servizi potrebbero avere restrizioni sui caratteri consentiti. I caratteri generalmente consentiti sono lettere, numeri, spazi e i simboli seguenti: **+ - = . _ : / @**.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Non è possibile utilizzare `aws:`, `AWS:` o qualsiasi combinazione di maiuscole o minuscole di un tale prefisso per chiavi o valori. Questi sono riservati solo all' AWS uso. Non è possibile modificare né eliminare le chiavi o i valori di tag con tale prefisso. I tag con questo prefisso non rientrano nel tuo tags-per-resource limite.

Tag gestiti da Amazon ECS

Quando utilizzi tag gestiti da Amazon ECS, Amazon ECS etichetta automaticamente tutte le attività appena avviate e tutti i volumi Amazon EBS collegati alle attività con le informazioni sul cluster e i tag di definizione delle attività aggiunti dall'utente o i tag di servizio. Di seguito vengono descritti i tag aggiunti:

- Attività autonome: un tag con una Chiave come `aws:ecs:clusterName` e un Valore impostato sul nome del cluster. Tutti i tag di definizione delle attività aggiunti dagli utenti. Un volume Amazon EBS collegato a un'attività autonoma riceverà il tag con una chiave come `aws:ecs:clusterName` e un valore impostato sul nome del cluster. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS.
- Attività che fanno parte di un servizio: un tag con una Chiave come `aws:ecs:clusterName` e un Valore impostato sul nome del cluster. Un tag con una Chiave come `aws:ecs:serviceName` e un Valore impostato sul nome del servizio. Tag da una delle risorse seguenti:
  - Definizioni delle attività: tutti i tag di definizione delle attività aggiunti dagli utenti.
  - Servizi: tutti i tag di servizio aggiunti dagli utenti.

Un volume Amazon EBS collegato a un'attività che fa parte di un servizio riceverà un tag con Key as `aws:ecs:clusterName` e Value impostati sul nome del cluster e un tag con Key as `aws:ecs:serviceName` e Value impostato sul nome del servizio. Per ulteriori informazioni sull'etichettatura dei volumi Amazon EBS, consulta Tagging dei volumi Amazon EBS.

Per questa funzionalità sono richieste le seguenti opzioni:

- Devi scegliere esplicitamente i nuovi formati per il nome della risorsa Amazon (ARN) e l'identificatore di risorsa (ID). Per ulteriori informazioni, consulta Amazon Resource Name (ARN) e ID.
- Quando utilizzi le API per creare un servizio o eseguire un'attività, devi impostare `enableECSManagedTags` su `true` per `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. Ti consigliamo di non modificare manualmente i tag in modo che Amazon ECS possa gestire efficacemente le funzionalità.

Utilizza i tag per la fatturazione

AWS fornisce uno strumento di reporting chiamato Cost Explorer che puoi utilizzare per analizzare il costo e l'utilizzo delle tue risorse Amazon ECS.

Utilizza Cost Explorer per visualizzare i grafici relativi all'utilizzo e ai costi. Puoi visualizzare i dati fino agli ultimi 13 mesi e prevedere le spese dei prossimi tre mesi. Puoi utilizzare Cost Explorer per visualizzare i modelli relativi a quanto spendi in risorse AWS nel tempo. Ad esempio, puoi utilizzarlo per identificare aree che richiedono ulteriore studio e visualizzare le tendenze che puoi utilizzare per comprendere i costi. Puoi anche specificare intervalli di tempo per i dati e visualizzare i dati temporali per mese o per giorno.

Puoi utilizzare i tag gestiti da Amazon ECS oppure tag aggiunti dall'utente per il tuo report su costi e utilizzo. Per ulteriori informazioni, consulta Report di utilizzo di Amazon ECS.

Per visualizzare il costo delle risorse combinate, puoi organizzare le informazioni di fatturazione in base alle risorse con gli stessi valori di chiave di tag. Puoi ad esempio applicare tag a numerose risorse con un nome di applicazione specifico, quindi organizzare le informazioni di fatturazione

per visualizzare il costo totale dell'applicazione in più servizi. Per ulteriori informazioni sulla configurazione di un report di allocazione dei costi mediante i tag, consulta Report di allocazione dei costi mensili nella Guida per l'utente di AWS Billing .

Inoltre, puoi attivare Dati di ripartizione dei costi per inserire i dati sull'utilizzo della CPU e della memoria a livello di attività in Report di costi utilizzo. Per ulteriori informazioni, consulta Report di costi e utilizzo a livello di attività.

> **Note**
>
> Se hai attivato il report, i dati relativi al mese corrente saranno disponibili per la visualizzazione entro 24 ore.

Aggiungere tag alle risorse Amazon ECS

Puoi taggare attività, servizi, definizioni di attività o cluster nuovi o esistenti. Per informazioni sull'etichettatura delle istanze del contenitore, consulta. Aggiungere tag a un'istanza di container Amazon ECS

> **Warning**
>
> Non aggiungere Informazioni personali di identificazione (PII) o altre informazioni riservate o sensibili nei tag. I tag sono accessibili a molti AWS servizi, inclusa la fatturazione. I tag non sono destinati ad essere utilizzati per dati privati o sensibili.

Le risorse seguenti consentono di specificare i tag al momento della creazione della risorsa.

Attività	Console	AWS CLI	Operazione API
Esegui una o più operazioni.	Esecuzione di un'applicazione come attività Amazon ECS	run-task	RunTask
Crea un servizio.	Creazione di un servizio Amazon	create-service	CreateService

Attività	Console	AWS CLI	Operazione API
	ECS utilizzando la console		
Crea un set di attività.	Implementa i servizi Amazon ECS utilizzando un controller di terze parti	create-task-set	CreateTaskSet
Registra una definizione di attività.	the section called “Creazione di una definizione di attività attraverso la nuova console”	register-task-definition	RegisterTaskDefinition
Crea un cluster.	Creazione di un cluster Amazon ECS per il tipo di lancio Fargate	create-cluster	CreateCluster
Esegui una o più istanze di container.	Avvio di un'istanza di container Linux di Amazon ECS	run-instances	RunInstances

Aggiungere tag alle risorse esistenti (console Amazon ECS)

Puoi aggiungere o eliminare tag associati ai cluster, ai servizi, alle attività e alle definizioni delle attività direttamente dalla pagina della risorsa.

Modifica di un tag per una singola risorsa

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
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. Selezionare Salva.

Aggiungere tag alle risorse esistenti (AWS CLI)

Puoi aggiungere o sovrascrivere uno o più tag utilizzando AWS CLI o un'API.

- AWS CLI - tag-risorsa
- Azione API - TagResource

Aggiungere tag a un'istanza di container Amazon ECS

Puoi associare i tag alle istanze di container utilizzando uno dei seguenti metodi:

- Metodo 1: durante la creazione dell'istanza di container tramite l'API Amazon EC2, la CLI o la console, specifica i tag passando i dati dell'utente all'istanza con il parametro di configurazione dell'agente del container `ECS_CONTAINER_INSTANCE_TAGS`. Ciò crea i tag associati solo all'istanza di container in Amazon ECS, non possono essere elencati utilizzando l'API Amazon EC2. Per ulteriori informazioni, consulta Avvio delle istanze di container Amazon ECS Linux per il trasferimento di dati.

  > **Important**
  >
  > Se avvii le istanze del container utilizzando un gruppo Amazon EC2 Auto Scaling, devi utilizzare 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 istanze Amazon EC2 avviate utilizzando i gruppi 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: durante la creazione dell'istanza di container tramite l'API Amazon EC2, la CLI o la console, specifica prima i tag con il parametro `TagSpecification.N`. 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'istanza Amazon EC2 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 fornire l'accesso che consenta ai tag dell'istanza di container di propagarsi da Amazon EC2 ad Amazon ECS, aggiungi manualmente le autorizzazioni seguenti sotto forma di policy in linea al ruolo IAM dell'istanza di container di Amazon ECS. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

  - `ec2:DescribeTags`

  Di seguito è riportata una policy di esempio utilizzata per aggiungere tali autorizzazioni.

  ```
  {
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
               "ec2:DescribeTags"
              ],
              "Resource": "*"
          }
      ]
  ```

```
}
```

Istanze di container esterni

Puoi associare i tag alle istanze di container esterne utilizzando uno dei seguenti metodi:

- Metodo 1: prima di eseguire lo script di installazione per registrare l'istanza esterna nel cluster, crea o modifica il file di configurazione dell'agente del container di Amazon ECS all'indirizzo `/etc/ecs/ecs.config` e aggiungi il parametro di configurazione dell'agente del container `ECS_CONTAINER_INSTANCE_TAGS`. Questo consente di creare i tag associati all'istanza esterna.

  Di seguito è riportato un esempio di sintassi.

  ```
  ECS_CONTAINER_INSTANCE_TAGS={"tag_key": "tag_value"}
  ```

- Metodo 2: dopo aver registrato l'istanza esterna nel cluster, puoi utilizzarla AWS Management Console per aggiungere tag. Per ulteriori informazioni, consulta Aggiungere tag alle risorse esistenti (console Amazon ECS).

Report di utilizzo di Amazon ECS

AWS fornisce uno strumento di reporting chiamato Cost Explorer che puoi utilizzare per analizzare il costo e l'utilizzo delle tue risorse Amazon ECS.

Utilizza Cost Explorer per visualizzare i grafici relativi all'utilizzo e ai costi. Puoi visualizzare i dati fino agli ultimi 13 mesi e prevedere le spese dei prossimi tre mesi. Puoi utilizzare Cost Explorer per visualizzare i modelli relativi a quanto spendi in risorse AWS nel tempo. Ad esempio, puoi utilizzarlo per identificare aree che richiedono ulteriore studio e visualizzare le tendenze che puoi utilizzare per comprendere i costi. Puoi anche specificare intervalli di tempo per i dati e visualizzare i dati temporali per mese o per giorno.

I dati di misurazione nel report su costi e utilizzo mostrano l'utilizzo in tutti i processi Amazon ECS. I dati di misurazione includono l'utilizzo della CPU come `vCPU-Hours` e l'utilizzo della memoria come `GB-Hours` per ogni attività eseguita. Come i dati presentati dipendono dal tipo di avvio del processo.

Per i processi che utilizzano il tipo di avvio Fargate, la colonna `lineItem/Operation` mostrerà `FargateTask` e verrà visualizzato il costo associato a ciascun processo.

Per i processi che utilizzano il tipo di avvio EC2, la colonna `lineItem/Operation` mostrerà `ECSTask-EC2` e i processi non avranno 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. Questi valori possono essere utilizzati per determinare il costo del cluster sottostante delle istanze Amazon EC2. I dati relativi ai costi e all'utilizzo delle istanze di Amazon EC2 verranno elencati separatamente nell'ambito del servizio Amazon EC2.

Puoi anche utilizzare i tag gestiti di Amazon ECS per identificare il servizio o il cluster a cui appartengono i singoli processi. Per ulteriori informazioni, consulta Utilizza i tag per la fatturazione.

> **Important**
>
> I dati di misurazione possono essere visualizzati solo per le attività avviate dopo il 16 novembre 2018. I processi avviati prima di questa data non mostrano i dati di misurazione.

Di seguito è illustrato un esempio dei campi in base a cui puoi ordinare i dati di allocazione dei costi utilizzando lo strumento Cost Explorer.

- Nome cluster
- Nome servizio
- Tag delle risorse
- Tipo di avvio
- Regione AWS
- Tipo di utilizzo

Per ulteriori informazioni sulla creazione di un report sui AWS costi e sull'utilizzo, consulta il report AWS sui costi e sull'utilizzo nella Guida per l'AWS Billing utente.

Report di costi e utilizzo a livello di attività

AWS Cost Management può fornire dati sull'utilizzo della CPU e della memoria AWS Cost and Usage Report per ogni attività su Amazon ECS, incluse le attività su Fargate e le attività su EC2. Questi dati sono denominati Dati di ripartizione dei costi. Puoi utilizzare questi dati per analizzare i costi e l'utilizzo delle applicazioni. Inoltre, puoi suddividere e allocare i costi a singole unità aziendali e team utilizzando tag di allocazione dei costi e categorie di costi. Per ulteriori informazioni sui dati di

allocazione dei costi suddivisi, consulta Comprendere i dati di allocazione dei costi suddivisi nella Guida per l'utente. AWS Cost and Usage Report

Puoi attivare Dati di ripartizione dei costi a livello di attività per l'account nella AWS Cost Management Console. Se disponi di un account di gestione (pagante), puoi scegliere di applicare questa configurazione a tutti gli account collegati.

Dopo aver impostato i dati di allocazione dei costi suddivisi, ci saranno colonne aggiuntive sotto l'splitLineItemintestazione 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 su EC2, questi dati suddividono il costo dell'istanza EC2 in base all'utilizzo delle risorse o alle prenotazioni e alle risorse rimanenti sull'istanza.

I seguenti sono prerequisiti:

- Imposta il parametro di configurazione dell'agente `ECS_DISABLE_METRICS` Amazon ECS 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 a monitorare Amazon ECS, crea un piano di monitoraggio che includa le risposte alle seguenti domande:

- Quali sono gli obiettivi del monitoraggio?
- Di quali risorse si intende eseguire il monitoraggio?
- Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno utilizzati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

I parametri resi disponibili dipendono dal tipo di avvio dei processi e dei servizi nei cluster. Se utilizzi il tipo di avvio Fargate per i tuoi servizi, vengono forniti i parametri di utilizzo della memoria e della CPU per semplificare il monitoraggio dei servizi. Per il tipo di avvio Amazon EC2, puoi e devi monitorare le istanze EC2 che costituiscono la tua infrastruttura sottostante. Ulteriori parametri di prenotazione e utilizzo della CPU e della memoria sono disponibili presso il cluster, il servizio e l'attività.

La fase successiva consiste nello stabilire una baseline per le prestazioni normali di Amazon ECS nell'ambiente, misurando le prestazioni in diversi momenti e con condizioni di carico differenti. Quando monitori Amazon ECS, archivia i dati storici di monitoraggio in modo da poterli confrontare con i dati sulle prestazioni correnti e per poter identificare i normali modelli di prestazioni e le anomalie e individuare i metodi per risolvere i problemi.

Per stabilire una baseline, è necessario monitorare almeno gli elementi seguenti:

- I parametri di utilizzo e prenotazione di CPU e memoria per i cluster Amazon ECS
- I parametri di utilizzo di CPU e memoria per i tuoi servizi Amazon ECS

  Per ulteriori informazioni, consulta Visualizzazione dei parametri Amazon ECS.

Best practice per il monitoraggio di Amazon ECS

Utilizza le seguenti best practice per monitorare Amazon ECS.

- Rendi il monitoraggio una priorità per risolvere piccoli problemi prima che diventino grandi
- Crea un piano di monitoraggio che includa le risposte alla seguente domanda
  - Quali sono gli obiettivi del monitoraggio?
  - Di quali risorse si intende eseguire il monitoraggio?
  - Con quale frequenza sarà eseguito il monitoraggio di queste risorse?
  - Quali strumenti di monitoraggio verranno utilizzati?
  - Chi eseguirà i processi di monitoraggio?
  - Chi deve ricevere una notifica quando si verifica un problema?
- Automatizza il monitoraggio il più possibile.
- Controlla i file di registro di Amazon ECS. Per ulteriori informazioni, consulta Visualizzazione dei log degli agenti container Amazon ECS.

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 policy di Amazon EC2 Auto Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta Monitora Amazon ECS utilizzando CloudWatch .

Per i servizi con attività che utilizzano il tipo di avvio Fargate, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le attività del servizio in base a CloudWatch metriche, come l'utilizzo della CPU e della memoria. Per ulteriori informazioni, consulta Ridimensiona automaticamente il tuo servizio Amazon ECS.

Per i cluster con attività o servizi che utilizzano il tipo di lancio EC2, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le istanze del contenitore in base CloudWatch a metriche, come la prenotazione della memoria del cluster.

Per le istanze di container che sono state lanciate con l'AMI Amazon Linux ottimizzata per Amazon ECS, puoi CloudWatch usare Logs per visualizzare diversi log delle tue istanze di container in un'unica comoda posizione. È necessario installare l'agente sulle istanze del contenitore. CloudWatch Per ulteriori informazioni, consulta Scaricare e configurare l' CloudWatch agente utilizzando la riga di comando nella Amazon CloudWatch User Guide. Devi inoltre aggiungere la policy `ECS-CloudWatchLogs` al ruolo `ecsInstanceRole`. Per ulteriori informazioni, consulta Monitoraggio delle autorizzazioni delle istanze dei container.

- Amazon CloudWatch Logs: monitora, archivia e accedi ai file di registro dai contenitori nelle tue attività Amazon ECS specificando il driver di `awslogs` registro nelle definizioni delle attività. Per ulteriori informazioni, consulta Invia i log di Amazon ECS a CloudWatch .

  Puoi anche monitorare, archiviare e accedere al sistema operativo e ai file di log dell'agente del container Amazon ECS dalle istanze di container Amazon ECS. Questo metodo per accedere ai log può essere utilizzato per i container che utilizzano il tipo di avvio EC2.
- Amazon CloudWatch Events: abbina gli eventi e li indirizza a una o più funzioni o stream di destinazione per apportare modifiche, acquisire informazioni sullo stato e intraprendere azioni correttive. Per ulteriori informazioni, consulta Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge questa guida e What Is Amazon CloudWatch Events? nella Guida per l'utente di Amazon CloudWatch Events.
- Container Insights: raccogli, aggrega e riepiloga metriche e log dalle tue applicazioni e microservizi containerizzati. Container Insights raccoglie dati come eventi di log delle prestazioni tramite Embedded Metric Format. Questi eventi del registro delle prestazioni sono voci che utilizzano uno schema JSON strutturato che consente di importare e archiviare dati ad alta cardinalità su larga scala. Da questi dati, CloudWatch crea metriche aggregate a livello di cluster, attività e servizio come metriche. CloudWatch Le metriche raccolte da Container Insights sono disponibili nei dashboard CloudWatch automatici e sono visualizzabili anche nella sezione Metriche della console. CloudWatch

- AWS CloudTrail monitoraggio dei log: condividi i file di registro tra account, monitora i file di CloudTrail registro in tempo reale inviandoli a CloudWatch Logs, scrivi applicazioni di elaborazione dei log in Java e verifica che i file di registro non siano cambiati dopo la consegna da parte di. CloudTrail Per ulteriori informazioni, Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail consultate questa guida e Lavorare con i file di CloudTrail registro nella Guida per l'AWS CloudTrail utente.
- Monitoraggio del runtime: rileva le minacce per cluster e contenitori all'interno del tuo AWS ambiente. Runtime Monitoring utilizza un agente di GuardDuty sicurezza che aggiunge visibilità di runtime ai singoli carichi di lavoro Amazon ECS, ad esempio accesso ai file, esecuzione dei processi e connessioni di rete.

Strumenti di monitoraggio manuali

Un'altra parte importante del monitoraggio di Amazon ECS consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Le dashboard della AWS console CloudWatch Trusted Advisor, e altre, forniscono una at-a-glance panoramica dello stato del tuo ambiente. AWS Ti consigliamo inoltre di controllare i file di log sulle tue istanze di container e i container nei tuoi processi.

- Console Amazon ECS:
  - Metriche del cluster per il tipo di lancio di EC2
  - Parametri del servizio
  - Stato di integrità dei servizi
  - Eventi di implementazione del servizio
- CloudWatch home page:
  - Stato e allarmi attuali
  - Grafici degli allarmi e delle risorse
  - Stato di integrità dei servizi

  Inoltre, è possibile utilizzare CloudWatch per effettuare le seguenti operazioni:
  - Creare pannelli di controllo personalizzati per monitorare i servizi di interesse.
  - Creare grafici dei dati dei parametri per la risoluzione di problemi e il rilevamento di tendenze.
  - Cerca e sfoglia tutte le metriche AWS delle tue risorse.
  - Crea e modifica gli allarmi per ricevere le notifiche dei problemi.

- Controllo dello stato del contenitore: si tratta di comandi che vengono eseguiti localmente su un contenitore e convalidano lo stato e la disponibilità delle applicazioni. Li configuri per contenitore nella definizione dell'attività.
- AWS Trusted Advisor può aiutarvi a monitorare AWS le vostre risorse per migliorare prestazioni, affidabilità, sicurezza ed economicità. Quattro Trusted Advisor controlli sono disponibili per tutti gli utenti; più di 50 controlli sono disponibili per gli utenti con un piano di supporto Business o Enterprise. Per ulteriori informazioni, consulta AWS Trusted Advisor.

  Trusted Advisor dispone dei seguenti controlli relativi ad Amazon ECS:

  - Una tolleranza agli errori che indica che un servizio è in esecuzione in un'unica zona di disponibilità.
  - Una tolleranza agli errori che indica che non è stata utilizzata la strategia di spread placement per più zone di disponibilità.
- AWS Compute Optimizer è un servizio che analizza le metriche di configurazione e utilizzo delle risorse. AWS Il servizio segnala se le risorse sono ideali e genera suggerimenti di ottimizzazione per ridurre i costi e migliorare le prestazioni dei carichi di lavoro.

  Per ulteriori informazioni, consulta AWS Compute Optimizer consigli per Amazon ECS.

Monitora Amazon ECS utilizzando CloudWatch

Puoi monitorare le tue risorse Amazon ECS utilizzando Amazon CloudWatch, che raccoglie ed elabora i dati grezzi da Amazon ECS in metriche leggibili quasi in tempo reale. Queste statistiche vengono registrate per un periodo di due settimane in modo da permettere l'accesso a informazioni storiche e per offrire una prospettiva migliore sulle prestazioni di cluster e servizi. I dati metrici di Amazon ECS vengono inviati automaticamente CloudWatch in periodi di 1 minuto. Per ulteriori informazioni CloudWatch, consulta la Amazon CloudWatch User Guide.

Amazon ECS fornisce parametri gratuiti per cluster e servizi. A un costo aggiuntivo, puoi attivare Amazon ECS CloudWatch Container Insights per il tuo cluster per parametri per attività, tra cui l'utilizzo di CPU, memoria e file system EBS. Per ulteriori informazioni su Container Insights, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.

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 istanze Amazon EC2, l'istanza Amazon EC2 richiede una `1.4.0` versione o successiva (Linux) o `1.0.0` o successiva (Windows) dell'agente CloudWatch contenitore per generare le 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 istanze Amazon EC2 richiedono anche l'`ecs:StartTelemetrySession`autorizzazione per il ruolo IAM con cui avvii le tue istanze Amazon EC2. Se hai creato il ruolo IAM dell'istanza del contenitore Amazon ECS prima che i CloudWatch parametri fossero disponibili per Amazon ECS, potresti dover aggiungere questa autorizzazione. Per informazioni sul ruolo IAM dell'istanza di contenitore e sull'associazione della politica IAM gestita per le istanze di container, consulta. Ruolo IAM delle istanze di container Amazon ECS
- Puoi disabilitare la raccolta di CloudWatch metriche sulle tue istanze Amazon EC2 impostandola `ECS_DISABLE_METRICS=true` nella configurazione dell'agente container Amazon ECS. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Parametri consigliati

Amazon ECS fornisce CloudWatch parametri gratuiti che puoi utilizzare per monitorare le tue risorse. La prenotazione di CPU e memoria e l'utilizzo di CPU, memoria e file system EBS nell'intero cluster e l'utilizzo di CPU, memoria e file system EBS sui servizi dei cluster possono essere misurati utilizzando questi parametri. Per i carichi di lavoro GPU, puoi misurare la prenotazione GPU nel cluster.

L'infrastruttura su cui sono ospitate le tue attività Amazon ECS nei tuoi cluster determina quali metriche sono disponibili. Per le attività ospitate sull'infrastruttura Fargate, Amazon ECS fornisce parametri di utilizzo di CPU, memoria e file system EBS per facilitare il monitoraggio dei tuoi servizi. Per le attività ospitate su istanze EC2, Amazon ECS fornisce parametri di prenotazione di CPU, memoria e GPU e parametri di utilizzo di CPU e memoria a livello di cluster e servizio. Devi monitorare le istanze Amazon EC2 che costituiscono la tua infrastruttura sottostante separatamente.

Per ulteriori informazioni sul monitoraggio delle istanze Amazon EC2, consulta Monitoring Amazon EC2 nella Amazon EC2 User Guide.

Per informazioni sugli allarmi consigliati da utilizzare con Amazon ECS, consulta uno dei seguenti articoli nella Amazon CloudWatch Logs User Guide:

- Amazon ECS
- Amazon ECS con Container Insights

Visualizzazione dei parametri Amazon ECS

Una volta che le risorse sono in esecuzione nel cluster, puoi visualizzare i parametri su Amazon ECS e CloudWatch sulle console. La console Amazon ECS fornisce una visualizzazione massima, minima e media di 24 ore dei parametri del cluster e del servizio. La CloudWatch console offre una visualizzazione dettagliata e personalizzabile delle risorse, nonché del numero di attività in esecuzione in un servizio.

Console Amazon ECS

I parametri di utilizzo di CPU e memoria del servizio Amazon ECS sono disponibili nella console Amazon ECS. La visualizzazione offerta per i parametri del servizio mostra i valori medio, minimo e massimo per il precedente periodo di 24 ore con punti di dati disponibili a intervalli di 5 minuti. Per ulteriori informazioni, consulta Metriche di utilizzo del servizio Amazon ECS.

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Seleziona il cluster di cui desideri visualizzare i parametri.
3. Determina le metriche da visualizzare.

Per visualizzare le metriche da	Fasi
Cluster	Nella pagina dei dettagli del cluster, scegli la scheda Metriche. È inoltre disponibile un collegamento alla CloudWatch console per visualizzare le metriche

Per visualizzare le metriche da	Fasi	
	di CloudWatch Container Insights, se le hai attivate.	
Servizi	Nella pagina dei dettagli del cluster, nella scheda Servizi, seleziona il servizio. Le metriche sono quindi disponibili nella scheda Health and metrics.	

CloudWatch console

Per il tipo di lancio di Fargate, i parametri del servizio Amazon ECS possono essere visualizzati anche sulla console. CloudWatch La console fornisce la vista più dettagliata dei parametri Amazon ECS e puoi personalizzare le visualizzazioni in base alle esigenze. È possibile visualizzare l'utilizzo del servizio e il conteggio delle attività del servizio IN ESECUZIONE.

Per il tipo di lancio EC2, i parametri del cluster e 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.

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.

`CPUReservation`

La percentuale di unità CPU riservate nel cluster o nel servizio.

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 istanze Amazon EC2 registrate nel cluster. Solo le istanze Amazon EC2 con `ACTIVE` o in `DRAINING` stato influiranno sui parametri di prenotazione della CPU. La metrica è supportata solo per le attività ospitate su un'istanza Amazon EC2.

Dimensioni valide: `ClusterName`.

Statistiche utili: media, minima, massima

Unità: percentuale.

`CPUUtilization`

La percentuale di unità CPU utilizzata dal cluster o dal servizio.

L'utilizzo della CPU a livello di cluster (filtrato per`ClusterName`) 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 istanze Amazon EC2 registrate nel cluster. Solo le istanze Amazon EC2 con `ACTIVE` o in `DRAINING` stato influiranno sui parametri di prenotazione della CPU. La metrica a livello di cluster è supportata solo per le attività ospitate su un'istanza Amazon EC2.

L'utilizzo della CPU a livello di servizio (filtrato per`ClusterName`,`ServiceName`) viene misurato come il totale delle unità CPU utilizzate dalle attività che appartengono al servizio, diviso per il numero totale di unità CPU riservate alle attività che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su istanze Amazon EC2 e Fargate.

Dimensioni valide: `ClusterName`, `ServiceName`.

Statistiche utili: media, minima, massima

Unità: percentuale.

`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 istanze Amazon EC2 registrate nel cluster. Questa metrica può essere filtrata solo per. `ClusterName` Solo le istanze Amazon EC2 con `ACTIVE` o in `DRAINING` stato influiranno sui parametri di prenotazione della memoria. La metrica di prenotazione della memoria a livello di cluster è supportata solo per le attività ospitate su un'istanza Amazon EC2.

> **Note**
>
> Quando si calcola l'utilizzo della memoria, se specificato, `MemoryReservation` viene utilizzato nel calcolo anziché la memoria totale.

Dimensioni valide: `ClusterName`.

Statistiche utili: media, minima, massima

Unità: percentuale.

`MemoryUtilization`

La percentuale di memoria utilizzata dal cluster o dal servizio.

L'utilizzo della memoria a livello di cluster (filtrato per`ClusterName`) viene misurato come la memoria totale utilizzata dalle attività di Amazon ECS sul cluster, divisa per la memoria totale di tutte le istanze Amazon EC2 registrate nel cluster. Solo le istanze Amazon EC2 con `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 su un'istanza Amazon EC2.

L'utilizzo della memoria a livello di servizio (filtrato per`ClusterName`,`ServiceName`) viene misurato come la memoria totale utilizzata dalle attività che appartengono al servizio, divisa per la memoria totale riservata alle attività che appartengono al servizio. La metrica del livello di servizio è supportata per le attività ospitate su istanze Amazon EC2 e Fargate.

Dimensioni valide: `ClusterName`, `ServiceName`.

Statistiche utili: media, minima, massima

Unità: percentuale.

`EBSFilesystemUtilization`

La percentuale del file system Amazon EBS utilizzata dalle attività di un servizio.

La metrica di utilizzo del file system EBS a livello di servizio (filtrata per`ClusterName`,`ServiceName`) viene misurata come la quantità totale del file system EBS utilizzata dalle attività che appartengono al servizio, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività che appartengono al servizio. La metrica di utilizzo del

file system EBS a livello di servizio è disponibile solo per le attività ospitate su istanze Amazon EC2 (utilizzando la versione container agent) `1.79.0` e Fargate (utilizzando la versione della piattaforma) a cui è collegato un volume EBS. `1.4.0`

> **Note**
>
> Per le attività ospitate su Fargate, c'è spazio sul disco che viene utilizzato solo da Fargate. Non ci sono costi associati allo spazio utilizzato da Fargate, ma vedrai questo spazio di archiviazione aggiuntivo utilizzando strumenti come. `df`

Dimensioni valide: `ClusterName, ServiceName`.

Statistiche utili: media, minima, massima

Unità: percentuale.

`GPUReservation`

La percentuale di unità GPU totali disponibili che viene prenotata dai processi in esecuzione nel cluster.

La metrica di prenotazione GPU a livello di cluster viene misurata come il numero di GPU riservate dalle attività di Amazon ECS sul cluster, diviso per il numero totale di GPU disponibili su tutte le istanze Amazon EC2 con GPU registrate nel cluster. Solo le istanze Amazon EC2 con `DRAINING` status `ACTIVE` o stato influiranno sui parametri di prenotazione della GPU.

Dimensioni valide: `ClusterName`.

Statistiche utili: media, minima, massima

Tutte le statistiche: media, minima, massima, somma, conteggio dei campioni.

Unità: percentuale.

`ActiveConnectionCount`

Il numero totale di connessioni simultanee 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`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`NewConnectionCount`

Il numero totale di nuove connessioni stabilite 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`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`ProcessedBytes`

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.

Dimensioni valide: `DiscoveryName` e `DiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: byte.

`RequestCount`

Il numero richieste di traffico in entrata elaborato dai proxy Service Connect.

Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.

È inoltre necessario configurare `appProtocol` la mappatura delle porte nella definizione dell'attività.

Dimensioni valide: `DiscoveryName` e `DiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`GrpcRequestCount`

Il numero 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.

Dimensioni valide: `DiscoveryName` e `DiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`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.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`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.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`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.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma

Unità: numero.

`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.

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`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.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media.

Unità: numero.

`TargetProcessedBytes`

Il numero totale di byte elaborati dai proxy Service Connect.

Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima, somma.

Unità: byte.

`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.

Dimensioni valide: `TargetDiscoveryName` e `TargetDiscoveryName, ServiceName, ClusterName`.

Statistiche utili: media, minima, massima.

Tutte le statistiche: media, minima, massima, somma, conteggio dei campioni.

Unità: millisecondi.

`ClientTLSNegotiationErrorCount`

Il numero totale di volte in cui la connessione TLS non è riuscita. Questa metrica viene utilizzata solo quando TLS è abilitato.

Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.

Dimensioni valide: `DiscoveryName` e`DiscoveryName`,,`ServiceName`. `ClusterName`

Statistiche utili: media, minima, massima, somma.

Unità: numero.

`TargetTLSNegotiationErrorCount`

Il numero totale di volte in cui la connessione TLS non è riuscita a causa di certificati client mancanti, AWS Private CA verifiche non riuscite o verifiche SAN non riuscite. Questa metrica viene utilizzata solo quando TLS è abilitato.

Questo parametro è disponibile solo se è stato configurato Amazon ECS Service Connect.

Dimensioni valide: `ServiceNameClusterName`, `TargetDiscoveryName` e. `TargetDiscoveryName`

Statistiche utili: media, minima, massima, somma.

Unità: numero.

Dimensioni per i parametri Amazon ECS

I parametri di Amazon ECS utilizzano lo spazio dei nomi `AWS/ECS` e forniscono i parametri per le seguenti dimensioni. Amazon ECS invia solo i parametri per le risorse contenenti attività nello stato `RUNNING`. Ad esempio, se si dispone di un cluster con un servizio ma tale servizio non ha attività in uno `RUNNING` stato, 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.

`ClusterName`

Questa dimensione filtra i dati richiesti per tutte le risorse in un cluster specificato. Tutti i parametri di Amazon ECS vengono filtrati per `ClusterName`.

`ServiceName`

Questa dimensione filtra i dati richiesti per tutte le risorse in un servizio specificato all'interno di un cluster specificato.

`DiscoveryName`

Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato per tutti i cluster Amazon ECS.

Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

`DiscoveryName, ServiceName, ClusterName`

Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato tra le attività che hanno questo nome di rilevamento e che vengono create dal servizio in questo cluster.

Se hai riutilizzato lo stesso nome di rilevamento in più servizi in spazi dei nomi diversi, utilizza questa dimensione per visualizzare i parametri del traffico in entrata per un servizio specifico.

Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

`TargetDiscoveryName`

Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato per tutti i cluster Amazon ECS.

A differenza di `DiscoveryName`, questi parametri del traffico misurano solo il traffico in entrata a questo `DiscoveryName` proveniente da altre attività di Amazon ECS che hanno una configurazione Service Connect in questo spazio dei nomi. Ciò include le attività eseguite dai servizi con una configurazione Service Connect solo client o client-server.

Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

`TargetDiscoveryName, ServiceName, ClusterName`

Questa dimensione filtra i dati richiesti per i parametri del traffico in base a un nome di rilevamento di Service Connect specificato, ma conta solo il traffico proveniente dalle attività create dal servizio in questo cluster.

Utilizza questa dimensione per visualizzare i parametri del traffico in entrata provenienti da un client specifico in un altro servizio.

A differenza di `DiscoveryName, ServiceName, ClusterName`, questi parametri del traffico misurano solo il traffico in entrata a questo `DiscoveryName` proveniente da altre attività di Amazon ECS che hanno una configurazione Service Connect in questo spazio dei nomi. Ciò include le attività eseguite dai servizi con una configurazione Service Connect solo client o client-server.

Tieni presente che una porta specifica in un container in esecuzione può avere più nomi di rilevamento.

AWS Fargate metriche di utilizzo

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

AWS Fargate le metriche di utilizzo corrispondono alle quote di servizio. AWS È possibile configurare gli allarmi che avvisano quando l'uso si avvicina a una quota di servizio. Per ulteriori informazioni sulle quote di servizio per Fargate, consulta AWS Fargate quote di servizio.

AWS Fargate pubblica le seguenti metriche nel namespace. `AWS/Usage`

Parametro	Descrizione
`ResourceCount`	Il numero totale delle risorse specificate in esecuzione nell'acco unt. La risorsa è definita dalle dimensioni associate attraverso il parametro.

Le seguenti dimensioni vengono utilizzate per perfezionare i parametri di utilizzo pubblicati da AWS Fargate.

Dimensione	Descrizione
`Service`	Il nome del AWS servizio che contiene la risorsa. Per i parametri di utilizzo di AWS Fargate , il valore di questa dimensione è `Fargate`.
`Type`	Il tipo di entità che viene segnalato. Attualmente, l'unico valore valido per le metriche di AWS Fargate utilizzo è`Resource`.
`Resource`	Il tipo di risorsa in esecuzione. Il tipo di risorsa in esecuzion e. 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

Dimensione	Descrizione
	come valore della dimensione Resource, i valori `Standard/OnDemand` validi sono e. `Standard/Spot`

Puoi utilizzare la console Service Quotas per visualizzare l'utilizzo su un grafico e configurare allarmi che ti avvisano quando l' AWS Fargate utilizzo si avvicina a una quota di servizio. Per informazioni su come creare un CloudWatch allarme per avvisarti quando ti avvicini a una soglia di quota, consulta Service Quotas and Amazon CloudWatch alarms nella Guida per l'utente Service Quotas

.

Metriche di prenotazione dei cluster Amazon ECS

I parametri di prenotazione del cluster vengono misurati come la percentuale di CPU, memoria e GPU che sono prenotati da tutti i processi di Amazon ECS in un cluster rispetto a CPU, memoria e GPU aggregati registrati per ogni istanza di container attiva nel cluster. Solo le istanze di container nello stato `ACTIVE` o `DRAINING` influenzano i parametri di prenotazione del cluster. Questa metrica viene utilizzata solo su cluster con attività o servizi ospitati su istanze EC2. Non è supportata nei cluster con attività ospitate su. AWS Fargate

```
                              (Total CPU units reserved by tasks in cluster) x 100
Cluster CPU reservation =
 ------------------------------------------------------------
                          (Total CPU units registered by container instances in
 cluster)
```

```
                                 (Total MiB of memory reserved by tasks in cluster x
 100)
Cluster memory reservation =
 ------------------------------------------------------------------
                             (Total MiB of memory registered by container instances in
 cluster)
```

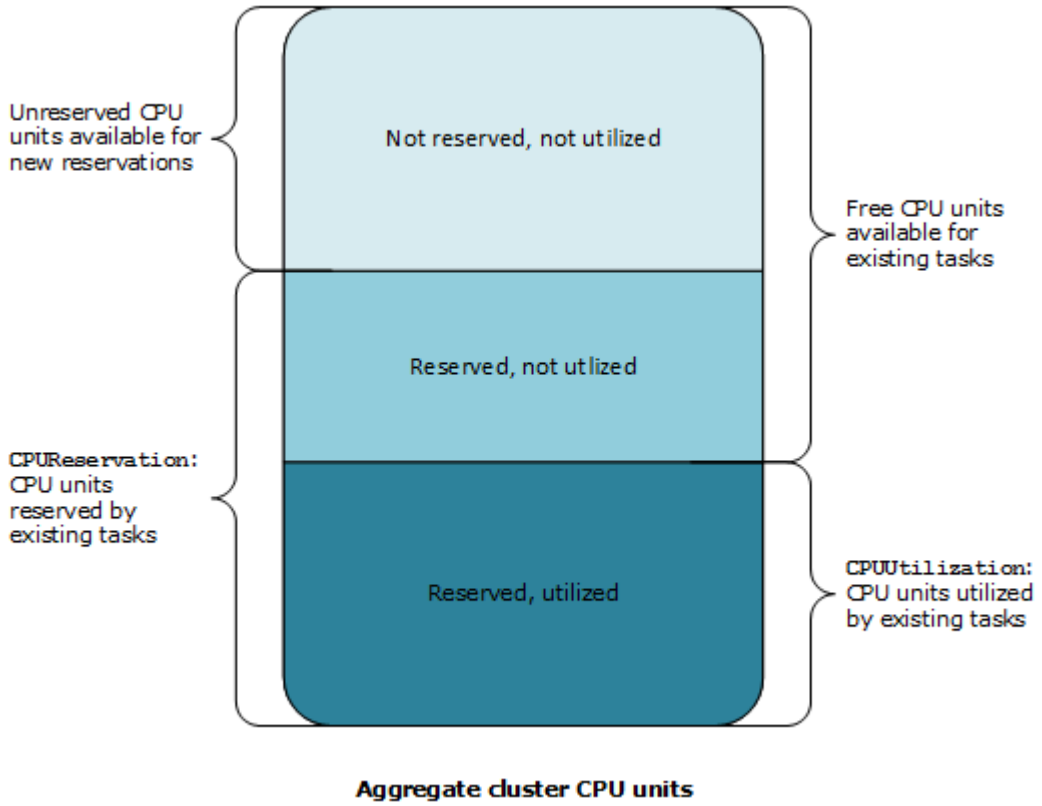
```
                                (Total GPUs reserved by tasks in cluster x 100)
Cluster GPU reservation =
 ------------------------------------------------------------------
                             (Total GPUs registered by container instances in cluster)
```

Quando esegui un processo in un cluster, Amazon ECS ne analizza la definizione di attività e prenota le unità di CPU, i MiB di memoria e le GPU aggregati che sono specificati nelle relative definizioni del container. Ogni minuto, Amazon ECS calcola il numero di unità di CPU, MiB di memoria e GPU attualmente prenotate per ogni processo in esecuzione nel cluster. Viene calcolata la quantità totale di CPU, memoria e GPU riservata per tutte le attività in esecuzione sul cluster e tali numeri vengono riportati CloudWatch come percentuale del totale delle risorse registrate per il cluster. Se nella definizione dell'attività specifichi un limite flessibile (`memoryReservation`), questo viene utilizzato per calcolare la quantità di memoria prenotata. In caso contrario, verrà utilizzato il limite rigido (`memory`). Il numero di MiB totali di memoria riservato dalle attività in un cluster include anche la dimensione del volume del file system (`tmpfs`) temporaneo e `sharedMemorySize` se definito nella definizione delle attività. Per ulteriori informazioni sui limiti rigidi e flessibili, dimensione della memoria condivisa e del volume tmpfs, consulta Parametri di definizione delle attività.

Ad esempio, un cluster ha due istanze di container attive registrate: un'istanza `c4.4xlarge` e un'istanza `c4.large`. L'istanza `c4.4xlarge` si registra nel cluster con 16.384 unità CPU e 30.158 MiB di memoria. L'istanza `c4.large` si registra con 2.048 unità CPU e 3.768 MiB di memoria. Le risorse aggregate di questo cluster sono 18.432 unità di CPU e 33.926 MiB di memoria.

Se una definizione di attività prenota 1.024 unità di CPU e 2.048 MiB di memoria e dieci attività vengono avviate con questa definizione attività su questo cluster (e nessun altra attività è attualmente in esecuzione), viene prenotato un totale di 10.240 unità di CPU e 20.480 MiB di memoria. Questo valore viene riportato CloudWatch come 55% di prenotazione della CPU e 60% di prenotazione di memoria per il cluster.

La figura seguente mostra il totale di unità di CPU registrate in un cluster e il significato della loro prenotazione e del loro utilizzo per le attività esistenti e il posizionamento di nuove attività. I blocchi inferiore (riservato, usato) e centrale (riservato, non utilizzato) rappresentano le unità CPU totali riservate alle attività esistenti in esecuzione sul cluster o alla `CPUReservation` CloudWatch metrica. Il blocco inferiore rappresenta le unità CPU riservate effettivamente utilizzate dalle attività in esecuzione sul cluster o la `CPUUtilization` CloudWatch metrica. Il blocco superiore rappresenta le unità di CPU che non sono prenotate dalle attività esistenti; queste unità di CPU sono disponibili per il nuovo posizionamento. Le attività esistenti possono usare queste unità di CPU non prenotate, se aumenta la loro necessità di risorse di CPU. Per ulteriori informazioni, consulta la documentazione del parametro della definizione di attività `cpu`.



Metriche di utilizzo dei cluster Amazon ECS

I parametri di utilizzo del cluster sono disponibili per CPU, memoria e, quando è presente un volume EBS collegato alle attività, per l'utilizzo del file system EBS. Questi parametri sono disponibili solo per i cluster con attività o servizi ospitati su istanze Amazon EC2. Non sono supportati su cluster con attività ospitate su. AWS Fargate

Parametri di utilizzo della CPU e della memoria a livello di cluster Amazon ECS

L'utilizzo della CPU e della memoria viene misurato come la percentuale di CPU e memoria utilizzata da tutte le attività su un cluster rispetto alla CPU e alla memoria aggregate registrate per ogni istanza Amazon EC2 attiva registrata nel cluster. Solo le istanze Amazon EC2 presenti `ACTIVE` o in `DRAINING` stato influiranno sui parametri di utilizzo del cluster.

```
                                (Total CPU units used by tasks in cluster) x 100
Cluster CPU utilization =
 ------------------------------------------------------------
                           (Total CPU units registered by container instances in
 cluster)
```

```
                                    (Total MiB of memory used by tasks in cluster x
 100)
Cluster memory utilization =
 ------------------------------------------------------------------
                              (Total MiB of memory registered by container instances in
 cluster)
```

Ogni minuto, l'agente container Amazon ECS su ogni istanza Amazon EC2 calcola il numero di unità CPU e MiB di memoria attualmente utilizzati per ogni attività in esecuzione su quell'istanza Amazon EC2 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 istanze Amazon EC2 attive registrate, un'`c4.4xlarge`istanza e un'istanza. `c4.large` L'`c4.4xlarge`istanza si registra nel cluster con unità `16,384` CPU e `30,158` MiB di memoria. L'`c4.large`istanza viene registrata con unità `2,048` CPU e `3,768` MiB di memoria. Le risorse aggregate di questo cluster sono unità `18,432` CPU e `33,926` MiB di memoria.

Se su questo cluster sono in esecuzione dieci attività e ogni attività consuma unità `1,024` CPU e `2,048` MiB di memoria, nel cluster viene utilizzato un totale `10,240` di unità CPU e `20,480` MiB di memoria. Ciò corrisponde al CloudWatch 55% di utilizzo della CPU e al 60% di utilizzo della memoria per il cluster.

Utilizzo del file system Amazon EBS a livello di cluster Amazon ECS

La metrica di utilizzo del file system EBS a livello di cluster viene misurata come la quantità totale del filesystem EBS utilizzata dalle attività in esecuzione sul cluster, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività del cluster.

```
                                      (Total GB of EBS filesystem used by tasks in
 cluster x 100)
Cluster EBS filesystem utilization =
 ------------------------------------------------------------------
                                      (Total GB of EBS filesystem allocated to tasks
 in cluster)
```

Metriche di utilizzo del servizio Amazon ECS

I parametri di utilizzo del servizio sono disponibili per la CPU, la memoria e, se alle tue attività è associato un volume EBS, l'utilizzo del file system EBS. Le metriche del livello di servizio sono supportate per i servizi con attività ospitate sia su istanze Amazon EC2 che su Fargate.

Livello di servizio, utilizzo della CPU e della memoria

L'utilizzo della CPU e della memoria viene misurato come la percentuale di CPU e memoria utilizzata dalle attività di Amazon ECS che appartengono a un servizio su un cluster rispetto alla CPU e alla memoria specificate nella definizione delle attività del servizio.

```
                                    (Total CPU units used by tasks in service) x 100
Service CPU utilization =
 ------------------------------------------------------------------------------
                          (Total CPU units specified in task definition) x (number of
 tasks in service)
```

```
                                     (Total MiB of memory used by tasks in service) x
 100
Service memory utilization =
 ------------------------------------------------------------------------------
                             (Total MiB of memory specified in task definition) x
 (number of tasks in service)
```

Ogni minuto, l'agente container Amazon ECS calcola il numero di unità CPU e MiB di memoria attualmente utilizzati per ogni attività di proprietà del servizio e queste informazioni vengono riportate ad Amazon ECS. Viene calcolata la quantità totale di CPU e memoria utilizzata per tutte le attività di proprietà del servizio in esecuzione sul cluster e tali numeri vengono riportati CloudWatch come percentuale delle risorse totali specificate per il servizio nella definizione dell'attività del servizio. 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 software, vedereDimensioni processo.

Ad esempio, la definizione di attività per un servizio specifica un totale di 512 unità di CPU e 1.024 MiB di memoria (con il parametro `memory` di limite rigido) per tutti i suoi container. Il servizio dispone di un conteggio desiderato di 1 attività in esecuzione, il servizio è in esecuzione in un cluster con 1 istanza di container `c4.large` (con 2.048 unità di CPU e 3.768 MiB di memoria totale) e non sono presenti altre attività in esecuzione nel cluster. Anche se l'attività specifica 512 unità di CPU, perché

è l'unica attività in esecuzione in un'istanza di container con 2.048 unità di CPU, può utilizzare fino a quattro volte la quantità specificata (2.048/512). Tuttavia, la memoria specificata di 1.024 MiB è un limite insuperabile, perciò in questo caso, l'utilizzo della memoria di servizio non può superare il 100%.

Se l'esempio precedente utilizzasse il parametro `memoryReservation` di limite flessibile anziché il parametro `memory` di limite rigido, le attività del servizio potrebbero utilizzare una quantità superiore a 1.024 MB di memoria, in base alle necessità. In questo caso, l'utilizzo della memoria di servizio di utilizzo potrebbe superare il 100%.

Se la tua applicazione presenta un picco improvviso nell'utilizzo della memoria per un breve periodo di tempo, non vedrai aumentare l'utilizzo della memoria di servizio perché Amazon ECS raccoglie più punti dati ogni minuto e poi li aggrega in un punto dati a cui viene inviato. CloudWatch

Se questa attività esegue un lavoro a uso intensivo di CPU per un periodo e utilizza tutte le 2.048 unità di CPU e tutti i 512 MiB di memoria disponibili, il servizio segnala il 400% di utilizzo di CPU e il 50% di utilizzo di memoria. Se l'attività è inattiva e l'utilizzo di 128 unità di CPU e 128 MiB di memoria, il servizio segnala il 25% di utilizzo di CPU e il 12,5% di utilizzo della memoria.

> **Note**
>
> In questo esempio, l'utilizzo della CPU supererà il 100% solo quando le unità CPU sono definite a livello di container. Se definisci le unità CPU a livello di processo, l'utilizzo non supererà il limite definito a livello di processo.

Utilizzo del file system EBS a livello di servizio

L'utilizzo del file system EBS a livello di servizio viene misurato come la quantità totale del file system EBS utilizzata dalle attività che appartengono al servizio, divisa per la quantità totale di storage del file system EBS allocata per tutte le attività che appartengono al servizio.

```
                                    (Total GB of EBS filesystem used by tasks in the
 service x 100)
Service EBS filesystem utilization =
 ------------------------------------------------------------------
                                    (Total GB of EBS filesystem allocated to tasks
 in the service)
```

Conteggio dei processi **RUNNING** del servizio

Puoi utilizzare le CloudWatch metriche per visualizzare il numero di attività dei tuoi servizi che si trovano nello stato. `RUNNING` Ad esempio, puoi impostare un CloudWatch allarme per questa metrica per avvisarti se il numero di attività in esecuzione nel tuo servizio scende al di sotto di un valore specificato.

Numero di **RUNNING** attività di servizio in Amazon ECS CloudWatch Container Insights

Una metrica «Number of Running Tasks» (`RunningTaskCount`) è disponibile per cluster e per servizio quando utilizzi Amazon ECS CloudWatch Container Insights. Puoi utilizzare Container Insights per tutti i nuovi cluster creati attivando l'impostazione dell'`containerInsights`account, 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 CloudWatch prezzi, consulta CloudWatch Prezzi.

Per visualizzare questo parametro, consulta i parametri di Amazon ECS Container Insights nella Amazon CloudWatch User Guide.

Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge

Con Amazon EventBridge, puoi automatizzare AWS i tuoi servizi e rispondere automaticamente a eventi di sistema come problemi di disponibilità delle applicazioni o modifiche delle risorse. Gli eventi AWS relativi ai servizi vengono forniti quasi EventBridge in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Le azioni che possono essere configurate automaticamente includono le seguenti:

- Aggiungere eventi ai gruppi di log in CloudWatch Logs
- Invocare una funzione AWS Lambda
- Richiamo del comando di esecuzione di Amazon EC2
- 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 Getting Started with 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 i tuoi processi utilizzano il tipo di avvio EC2, puoi visualizzare sia lo stato delle istanze di container che lo stato corrente di tutti i processi in esecuzione su tali istanze di container. Se le tue attività utilizzano il tipo di avvio Fargate, puoi vedere lo stato delle istanze del contenitore.

Utilizzando EventBridge, puoi creare pianificatori personalizzati su Amazon ECS responsabili dell'orchestrazione delle attività tra i cluster e del monitoraggio dello stato dei cluster quasi in tempo reale. Puoi eliminare il codice di pianificazione e monitoraggio che interroga continuamente il servizio Amazon ECS per le modifiche di stato e gestire invece le modifiche di stato di Amazon ECS in modo asincrono utilizzando qualsiasi destinazione. EventBridge Gli obiettivi potrebbero includere AWS Lambda Amazon Simple Queue Service, Amazon Simple Notification Service o Amazon Kinesis Data Streams.

Un flusso di eventi Amazon ECS assicura che ogni evento venga rilasciato almeno una volta. Se vengono inviati eventi duplicati, l'evento fornisce informazioni sufficienti a identificare i duplicati. Per ulteriori informazioni, consulta Gestione degli eventi Amazon ECS.

Gli eventi sono relativamente ordinati, in modo che tu possa facilmente stabilire quando un evento si verifica in relazione ad altri eventi.

Argomenti

Eventi Amazon ECS

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 ha generato e invia i seguenti tipi di eventi a EventBridge: eventi di modifica dello stato dell'istanza del contenitore, eventi di modifica dello stato delle attività, azioni del servizio ed eventi di modifica dello stato di distribuzione del servizio.

- Modifica dello stato dell'istanza del contenitore
- Modifica dello stato dell'attività
- Deployment state change (Cambio stato distribuzione)
- Azione di servizio

> **Note**
>
> Amazon ECS potrebbe aggiungere in futuro altri tipi di eventi, origini e dettagli. Se state deserializzando i dati JSON degli eventi nel codice, assicuratevi che l'applicazione sia pronta a gestire proprietà sconosciute per evitare problemi se e quando queste proprietà aggiuntive vengono 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 Events and Event Patterns 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 dalle API di Amazon ECS 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.

Per ulteriori informazioni su come integrare Amazon ECS e EventBridge, consulta Integrating Amazon and EventBridge Amazon ECS.

Eventi di modifica dello stato dell'istanza del contenitore Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle istanze di container.

Puoi chiamare le operazioni API `StartTask`, `RunTask` o `StopTask` direttamente o con la AWS Management Console o gli SDK.

> 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.

Annulla la registrazione dell'istanza del contenitore con l'operazione `DeregisterContainerInstance` API, direttamente o con gli AWS Management Console SDK o.

> 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 un'istanza EC2 è stata interrotta.

> Quando interrompi un'istanza di container, le attività in esecuzione vengono trasferite in stato `STOPPED`.

L'agente del container di Amazon ECS registra un'istanza di container per la prima volta.

> La prima volta che l'agente del container Amazon ECS registra un'istanza di container (all'avvio durante la prima esecuzione manuale), viene creato un evento di modifica dello stato per l'istanza.

L'agente del container Amazon ECS si connette o si disconnette da Amazon ECS.

Quando l'agente del container Amazon ECS si connette o si disconnette dal back-end di Amazon ECS modifica lo stato `agentConnected` dell'istanza di container.

> **Note**
>
> L'agente del container di Amazon ECS si disconnette e si ricollega più volte all'ora come parte del normale funzionamento, pertanto è necessario prevedere gli eventi di connessione dell'agente. Questi eventi non indicano l'esistenza di un problema con l'agente container o l'istanza di container.

Puoi aggiornare l'agente del container Amazon ECS su un'istanza.

Il dettaglio di un'istanza di container contiene un oggetto per la versione dell'agente del container. Se aggiorni l'agente, le informazioni sulla versione cambiano e generano un evento.

Example Evento di modifica dello stato delle istanze di container

Gli eventi di modifica dello stato delle istanza di container vengono forniti nel formato seguente. La `detail` sezione seguente è simile all'ContainerInstanceoggetto restituito da un'operazione API DescribeContainerInstances nel riferimento all'API di Amazon Elastic Container Service. Per ulteriori informazioni sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

```
{
  "version": "0",
  "id": "8952ba83-7be2-4ab5-9c32-6687532d15a2",
  "detail-type": "ECS Container Instance State Change",
  "source": "aws.ecs",
  "account": "111122223333",
  "time": "2016-12-06T16:41:06Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:ecs:us-east-1:111122223333:container-instance/
b54a2a04-046f-4331-9d74-3f6d7f6ca315"
  ],
  "detail": {
    "agentConnected": true,
    "attributes": [
```

```
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
      },
      {
        "name": "com.amazonaws.ecs.capability.task-iam-role-network-host"
      },
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.awslogs"
      },
      {
        "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
      },
      {
        "name": "com.amazonaws.ecs.capability.privileged-container"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
      },
      {
        "name": "com.amazonaws.ecs.capability.ecr-auth"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.20"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.21"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.22"
      },
      {
        "name": "com.amazonaws.ecs.capability.docker-remote-api.1.23"
      },
      {
        "name": "com.amazonaws.ecs.capability.task-iam-role"
      }
    ],
    "clusterArn": "arn:aws:ecs:us-east-1:111122223333:cluster/default",
```

```
    "containerInstanceArn": "arn:aws:ecs:us-east-1:111122223333:container-instance/
b54a2a04-046f-4331-9d74-3f6d7f6ca315",
    "ec2InstanceId": "i-f3a8506b",
    "registeredResources": [
      {
        "name": "CPU",
        "type": "INTEGER",
        "integerValue": 2048
      },
      {
        "name": "MEMORY",
        "type": "INTEGER",
        "integerValue": 3767
      },
      {
        "name": "PORTS",
        "type": "STRINGSET",
        "stringSetValue": [
          "22",
          "2376",
          "2375",
          "51678",
          "51679"
        ]
      },
      {
        "name": "PORTS_UDP",
        "type": "STRINGSET",
        "stringSetValue": []
      }
    ],
    "remainingResources": [
      {
        "name": "CPU",
        "type": "INTEGER",
        "integerValue": 1988
      },
      {
        "name": "MEMORY",
        "type": "INTEGER",
        "integerValue": 767
      },
      {
        "name": "PORTS",
```

```
        "type": "STRINGSET",
        "stringSetValue": [
          "22",
          "2376",
          "2375",
          "51678",
          "51679"
        ]
      },
      {
        "name": "PORTS_UDP",
        "type": "STRINGSET",
        "stringSetValue": []
      }
    ],
    "status": "ACTIVE",
    "version": 14801,
    "versionInfo": {
      "agentHash": "aebcbca",
      "agentVersion": "1.13.0",
      "dockerVersion": "DockerVersion: 1.11.2"
    },
    "updatedAt": "2016-12-06T16:41:06.991Z"
  }
}
```

Eventi di modifica dello stato delle attività di Amazon ECS

I seguenti scenari provocano eventi di modifica dello stato delle attività:

Puoi chiamare le operazioni API `StartTask`, `RunTask` o `StopTask` direttamente o con la AWS Management Console, l' AWS CLI o gli SDK.

L'avvio o l'arresto delle attività crea nuove risorse delle attività o modifica lo stato delle risorse delle attività esistenti.

Il pianificatore del servizio Amazon ECS inizia o interrompe un processo.

L'avvio o l'arresto delle attività crea nuove risorse delle attività o modifica lo stato delle risorse delle attività esistenti.

L'agente del container Amazon ECS chiama l'operazione API `SubmitTaskStateChange`

Per il tipo di avvio di Amazon EC2, l'agente container Amazon ECS monitora lo stato delle attività sulle istanze di container. L'agente container Amazon ECS segnala eventuali modifiche di stato. Le modifiche dello stato potrebbero includere modifiche da `PENDING` a `RUNNING` o da `RUNNING` a `STOPPED`.

Imponi la cancellazione dell'istanza del contenitore sottostante con l'operazione `DeregisterContainerInstance` API e il `force` flag, direttamente o con gli SDK o. AWS Management Console

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. La `detail` sezione seguente è simile all'oggetto Task restituito da un'operazione DescribeTasksAPI nell'Amazon Elastic Container Service API Reference. Se i contenitori utilizzano un'immagine ospitata con Amazon ECR, viene restituito il campo `imageDigest`.

> **Note**
>
> I valori per i campi `createdAt`, `connectivityAt`, `pullStartedAt`, `startedAt`, `pullStoppedAt` e `updatedAt` sono timestamp UNIX nella risposta di un'operazione `DescribeTasks`mentre nell'evento di modifica di stato dell'attività sono timestamp di stringhe ISO.

Per ulteriori informazioni sui parametri CloudWatch Events, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

Per informazioni su come configurare una regola di EventBridge eventi Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita perché uno dei suoi contenitori essenziali è terminato, consulta Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS

```
{
    "version": "0",
    "id": "3317b2af-7005-947d-b652-f55e762e571a",
    "detail-type": "ECS Task State Change",
    "source": "aws.ecs",
    "account": "111122223333",
    "time": "2020-01-23T17:57:58Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:task/FargateCluster/
c13b4cb40f1f4fe4a2971f76ae5a47ad"
    ],
    "detail": {
        "attachments": [
            {
                "id": "1789bcae-ddfb-4d10-8ebe-8ac87ddba5b8",
                "type": "eni",
                "status": "ATTACHED",
                "details": [
                    {
                        "name": "subnetId",
                        "value": "subnet-abcd1234"
                    },
                    {
                        "name": "networkInterfaceId",
```

```
                        "value": "eni-abcd1234"
                    },
                    {
                        "name": "macAddress",
                        "value": "0a:98:eb:a7:29:ba"
                    },
                    {
                        "name": "privateIPv4Address",
                        "value": "10.0.0.139"
                    }
                ]
            }
        ],
        "availabilityZone": "us-west-2c",
        "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/FargateCluster",
        "containers": [
            {
                "containerArn": "arn:aws:ecs:us-west-2:111122223333:container/
cf159fd6-3e3f-4a9e-84f9-66cbe726af01",
                "lastStatus": "RUNNING",
                "name": "FargateApp",
                "image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/hello-
repository:latest",
                "imageDigest":
 "sha256:74b2c688c700ec95a93e478cdb959737c148df3fbf5ea706abe0318726e885e6",
                "runtimeId":
 "ad64cbc71c7fb31c55507ec24c9f77947132b03d48d9961115cf24f3b7307e1e",
                "taskArn": "arn:aws:ecs:us-west-2:111122223333:task/FargateCluster/
c13b4cb40f1f4fe4a2971f76ae5a47ad",
                "networkInterfaces": [
                    {
                        "attachmentId": "1789bcae-ddfb-4d10-8ebe-8ac87ddba5b8",
                        "privateIpv4Address": "10.0.0.139"
                    }
                ],
                "cpu": "0"
            }
        ],
        "createdAt": "2020-01-23T17:57:34.402Z",
        "launchType": "FARGATE",
        "cpu": "256",
        "memory": "512",
        "desiredStatus": "RUNNING",
        "group": "family:sample-fargate",
```

```
        "lastStatus": "RUNNING",
        "overrides": {
            "containerOverrides": [
                {
                    "name": "FargateApp"
                }
            ]
        },
        "connectivity": "CONNECTED",
        "connectivityAt": "2020-01-23T17:57:38.453Z",
        "pullStartedAt": "2020-01-23T17:57:52.103Z",
        "startedAt": "2020-01-23T17:57:58.103Z",
        "pullStoppedAt": "2020-01-23T17:57:55.103Z",
        "updatedAt": "2020-01-23T17:57:58.103Z",
        "taskArn": "arn:aws:ecs:us-west-2:111122223333:task/FargateCluster/
c13b4cb40f1f4fe4a2971f76ae5a47ad",
        "taskDefinitionArn": "arn:aws:ecs:us-west-2:111122223333:task-definition/
sample-fargate:1",
        "version": 4,
        "platformVersion": "1.3.0"
    }
}
```

Eventi di azione del servizio Amazon ECS

Amazon ECS invia eventi di operazioni di servizio con il tipo di dettaglio Operazione di servizio ECS. A differenza degli eventi di modifica dello stato delle istanze di container e delle attività, gli eventi di operazioni di servizio non includono un numero di versione nel campo di risposta `details`. Di seguito è riportato uno schema di eventi utilizzato per creare una EventBridge regola per gli eventi di azione del servizio Amazon ECS. Per ulteriori informazioni, consulta Creating an EventBridge Rule nella 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.

Eventi di operazione di servizio con il tipo di eventi **`WARN`**

`SERVICE_TASK_START_IMPAIRED`

Il servizio non è in grado di avviare le attività in modo coerente.

`SERVICE_DISCOVERY_INSTANCE_UNHEALTHY`

Un servizio che utilizza l'individuazione dei servizi contiene un'attività non integra. Il pianificatore del servizio rileva che un'attività in un registro di servizio non è integra.

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 l'individuazione dei servizi.

`SERVICE_TASK_PLACEMENT_FAILURE`

Il pianificatore del servizio non è in grado di posizionare un'attività. La causa è descritta nel campo `reason`.

Una causa comune per la generazione di questo evento di servizio è dovuta alla mancanza di risorse nel cluster per collocare l'attività. Ad esempio, non vi è una sufficiente capacità di CPU o memoria nelle istanze di container disponibili o non è disponibile alcuna istanza di container. Un'altra causa comune si verifica quando l'agente di container di Amazon ECS viene disconnesso nell'istanza di container, impedendo all'utilità di pianificazione di collocare l'attività.

`SERVICE_TASK_CONFIGURATION_FAILURE`

Il pianificatore del servizio non è in grado di posizionare un'attività a causa di un errore di configurazione. La causa è descritta nel campo `reason`.

Una causa comune della generazione di questo evento di servizio è dovuta al fatto che i tag sono stati applicati al servizio ma l'utente o il ruolo non avevano scelto il nuovo formato del nome della risorsa Amazon (ARN) nella regione. Per ulteriori informazioni, consulta Amazon Resource Name (ARN) e ID. Un'altra causa comune è che Amazon ECS non è riuscito ad assumere il ruolo IAM del processo fornito.

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 Events and Event Patterns 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 sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

Nell'esempio seguente, l'attività tenta di utilizzare il provider di capacità `FARGATE_SPOT`, ma il pianificatore del servizio non è stato in grado di acquisire alcuna capacità Fargate Spot.

```
{
    "version": "0",
    "id": "ddca6449-b258-46c0-8653-e0e3a6d0468b",
    "detail-type": "ECS Service Action",
    "source": "aws.ecs",
    "account": "111122223333",
    "time": "2019-11-19T19:55:38Z",
    "region": "us-west-2",
    "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
    ],
    "detail": {
        "eventType": "ERROR",
        "eventName": "SERVICE_TASK_PLACEMENT_FAILURE",
```

```
        "clusterArn": "arn:aws:ecs:us-west-2:111122223333:cluster/default",
        "capacityProviderArns": [
            "arn:aws:ecs:us-west-2:111122223333:capacity-provider/FARGATE_SPOT"
        ],
        "reason": "RESOURCE:FARGATE",
        "createdAt": "2019-11-06T19:09:33.087Z"
    }
}
```

Nell'esempio seguente per il tipo di avvio EC2, è stato tentato di avviare il processo sull'istanza di container `2dd1b186f39845a584488d2ef155c131` ma il pianificatore del servizio non è stato in grado di posizionare il processo perché la CPU è 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, consulta Creating an EventBridge Rule nella Amazon EventBridge User Guide.

```
{
    "source": [
        "aws.ecs"
    ],
    "detail-type": [
        "ECS Deployment State Change"
    ]
}
```

Amazon ECS invia eventi di tipo `INFO` e `ERROR`. 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 Events and Event Patterns nella Amazon EventBridge User Guide.

Di seguito è illustrato un output di esempio per l'avvio di un'implementazione iniziale.

```
{
   "version": "0",
```

```
   "id": "ddca6449-b258-46c0-8653-e0e3a6EXAMPLE",
   "detail-type": "ECS Deployment State Change",
   "source": "aws.ecs",
   "account": "111122223333",
   "time": "2020-05-23T12:31:14Z",
   "region": "us-west-2",
   "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
   ],
   "detail": {
        "eventType": "INFO",
        "eventName": "SERVICE_DEPLOYMENT_IN_PROGRESS",
        "deploymentId": "ecs-svc/123",
        "updatedAt": "2020-05-23T11:11:11Z",
        "reason": "ECS deployment deploymentId in progress."
   }
}
```

Di seguito è illustrato un output di esempio per l'avvio di un'implementazione del ripristino dello stato precedente. Il campo `reason` fornisce l'ID dell'implementazione in cui il servizio sta eseguendo il ripristino dello stato precedente.

```
{
   "version": "0",
   "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
   "detail-type": "ECS Deployment State Change",
   "source": "aws.ecs",
   "account": "111122223333",
   "time": "2020-05-23T12:31:14Z",
   "region": "us-west-2",
   "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
   ],
   "detail": {
        "eventType": "INFO",
        "eventName": "SERVICE_DEPLOYMENT_IN_PROGRESS",
        "deploymentId": "ecs-svc/123",
        "updatedAt": "2020-05-23T11:11:11Z",
        "reason": "ECS deployment circuit breaker: rolling back to
 deploymentId deploymentID."
   }
}
```

Example Evento di implementazione del servizio completata

Gli eventi di implementazione del servizio completata vengono forniti nel seguente formato. Per ulteriori informazioni, consulta Implementa i servizi Amazon ECS sostituendo le attività.

```
{
   "version": "0",
   "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
   "detail-type": "ECS Deployment State Change",
   "source": "aws.ecs",
   "account": "111122223333",
   "time": "2020-05-23T12:31:14Z",
   "region": "us-west-2",
   "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
   ],
   "detail": {
        "eventType": "INFO",
        "eventName": "SERVICE_DEPLOYMENT_COMPLETED",
        "deploymentId": "ecs-svc/123",
        "updatedAt": "2020-05-23T11:11:11Z",
        "reason": "ECS deployment deploymentID completed."
   }
}
```

Example Evento di implementazione del servizio non riuscita

Gli eventi di implementazione del servizio non riuscita vengono forniti nel seguente formato. Un evento di implementazione del servizio con stato non riuscito verrà inviato solo per i servizi che dispongono della logica dell'interruttore automatico di implementazione abilitata. Per ulteriori informazioni, consulta Implementa i servizi Amazon ECS sostituendo le attività.

```
{
   "version": "0",
   "id": "ddca6449-b258-46c0-8653-e0e3aEXAMPLE",
   "detail-type": "ECS Deployment State Change",
   "source": "aws.ecs",
   "account": "111122223333",
   "time": "2020-05-23T12:31:14Z",
   "region": "us-west-2",
   "resources": [
        "arn:aws:ecs:us-west-2:111122223333:service/default/servicetest"
   ],
```

```
    "detail": {
         "eventType": "ERROR",
         "eventName": "SERVICE_DEPLOYMENT_FAILED",
         "deploymentId": "ecs-svc/123",
         "updatedAt": "2020-05-23T11:11:11Z",
         "reason": "ECS deployment circuit breaker: task failed to start."
    }
}
```

Gestione degli eventi Amazon ECS

Amazon ECS invia gli eventi secondo il criterio almeno una volta. Ciò significa che potresti ricevere più copie di un determinato evento. Inoltre, gli eventi potrebbero non essere inviati al listener di eventi nell'ordine in cui si sono verificati.

Per ordinare correttamente gli eventi, la sezione `detail` di ogni evento contiene una proprietà `version`. Ogni volta che una risorsa cambia stato, avviene un incremento di `version`. Gli eventi duplicati hanno la stessa `version` nell'oggetto `detail`. Se stai replicando l'istanza del container Amazon ECS e lo stato dell'attività con EventBridge, puoi confrontare la versione di una risorsa riportata dalle API di Amazon ECS 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:

- Stato ECS: memorizza CtrInstance lo stato più recente per un'istanza di contenitore. L'ID tabella è il valore `containerInstanceArn` dell'istanza di container;
- ECS TaskState: memorizza lo stato più recente di un'attività. L'ID tabella è il valore `taskArn` dell'attività.

```
import json
import boto3

def lambda_handler(event, context):
```

```
    id_name = ""
    new_record = {}

    # For debugging so you can see raw event format.
    print('Here is the event:')
    print((json.dumps(event)))

    if event["source"] != "aws.ecs":
        raise ValueError("Function only supports input from events with a source type
of: aws.ecs")

    # Switch on task/container events.
    table_name = ""
    if event["detail-type"] == "ECS Task State Change":
        table_name = "ECSTaskState"
        id_name = "taskArn"
        event_id = event["detail"]["taskArn"]
    elif event["detail-type"] == "ECS Container Instance State Change":
        table_name = "ECSCtrInstanceState"
        id_name =  "containerInstanceArn"
        event_id = event["detail"]["containerInstanceArn"]
    else:
        raise ValueError("detail-type for event is not a supported type. Exiting
without saving event.")

    new_record["cw_version"] = event["version"]
    new_record.update(event["detail"])

    # "status" is a reserved word in DDB, but it appears in containerPort
    # state change messages.
    if "status" in event:
        new_record["current_status"] = event["status"]
        new_record.pop("status")


    # Look first to see if you have received a newer version of an event ID.
    # If the version is OLDER than what you have on file, do not process it.
    # Otherwise, update the associated record with this latest information.
    print("Looking for recent event with same ID...")
    dynamodb = boto3.resource("dynamodb", region_name="us-east-1")
    table = dynamodb.Table(table_name)
    saved_event = table.get_item(
        Key={
            id_name : event_id
```

```
        }
    )
    if "Item" in saved_event:
        # Compare events and reconcile.
        print(("EXISTING EVENT DETECTED: Id " + event_id + " - reconciling"))
        if saved_event["Item"]["version"] < event["detail"]["version"]:
            print("Received event is a more recent version than the stored event -
 updating")
            table.put_item(
                Item=new_record
            )
        else:
            print("Received event is an older version than the stored event -
 ignoring")
    else:
        print(("Saving new event - ID " + event_id))

        table.put_item(
            Item=new_record
        )
```

Il seguente esempio di Fargate mostra una funzione Lambda scritta in Python 3.9 che acquisisce gli eventi di modifica dello stato delle attività e li salva nella seguente tabella Amazon DynamoDB:

```
import json
import boto3

def lambda_handler(event, context):
    id_name = ""
    new_record = {}

    # For debugging so you can see raw event format.
    print('Here is the event:')
    print((json.dumps(event)))

    if event["source"] != "aws.ecs":
       raise ValueError("Function only supports input from events with a source type
 of: aws.ecs")

    # Switch on task/container events.
    table_name = ""
    if event["detail-type"] == "ECS Task State Change":
        table_name = "ECSTaskState"
```

```
        id_name = "taskArn"
        event_id = event["detail"]["taskArn"]
    else:
        raise ValueError("detail-type for event is not a supported type. Exiting
without saving event.")

    new_record["cw_version"] = event["version"]
    new_record.update(event["detail"])

    # "status" is a reserved word in DDB, but it appears in containerPort
    # state change messages.
    if "status" in event:
        new_record["current_status"] = event["status"]
        new_record.pop("status")


    # Look first to see if you have received a newer version of an event ID.
    # If the version is OLDER than what you have on file, do not process it.
    # Otherwise, update the associated record with this latest information.
    print("Looking for recent event with same ID...")
    dynamodb = boto3.resource("dynamodb", region_name="us-east-1")
    table = dynamodb.Table(table_name)
    saved_event = table.get_item(
        Key={
            id_name : event_id
        }
    )
    if "Item" in saved_event:
        # Compare events and reconcile.
        print(("EXISTING EVENT DETECTED: Id " + event_id + " - reconciling"))
        if saved_event["Item"]["version"] < event["detail"]["version"]:
            print("Received event is a more recent version than the stored event -
updating")
            table.put_item(
                Item=new_record
            )
        else:
            print("Received event is an older version than the stored event -
ignoring")
    else:
        print(("Saving new event - ID " + event_id))

        table.put_item(
            Item=new_record
```

```
)
```

Monitora i contenitori Amazon ECS utilizzando Container Insights

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log delle applicazioni e dei microservizi containerizzati.

Container Insights scoprirà tutti i container in esecuzione in un cluster e raccoglierà dati sulle prestazioni a ogni livello dello stack di prestazioni. I dati operativi sono raccolti come eventi di log delle prestazioni. Questi sono elementi che usano uno schema JSON strutturato che consente ai dati ad alta cardinalità di essere acquisiti e archiviati su larga scala. Da questi dati, CloudWatch crea metriche aggregate di livello superiore a livello di cluster, servizio e attività come metriche. CloudWatch I parametri includono l'utilizzo di risorse come CPU, memoria, dischi e rete. Le metriche sono disponibili nei dashboard automatici. CloudWatch Per informazioni sui parametri disponibili, consulta i parametri di Amazon ECS Container Insights nella Amazon CloudWatch User Guide.

> **Important**
>
> Le metriche raccolte da CloudWatch Container Insights vengono addebitate come metriche personalizzate. Per ulteriori informazioni sui CloudWatch prezzi, consulta CloudWatch Prezzi. Amazon ECS offre anche parametri di monitoraggio che vengono forniti senza costi aggiuntivi. Per ulteriori informazioni, consulta Monitora Amazon ECS utilizzando CloudWatch .

Considerazioni

Quando si utilizza CloudWatch Container Insights, è necessario considerare quanto segue.

- CloudWatch Le metriche di Container Insights riflettono solo le risorse con attività in esecuzione nell'intervallo di tempo specificato. Ad esempio, se hai un cluster con un solo servizio ma quel servizio non ha attività in uno `RUNNING` stato, 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.
- I parametri di rete sono disponibili per tutti i processi eseguiti su Fargate e i processi vengono eseguiti su istanze Amazon EC2 che utilizzano le modalità di rete `bridge` o `awsvpc`.

Puoi visualizzare gli eventi del ciclo di vita delle attività e dei servizi di Amazon ECS all'interno della console CloudWatch Container Insights. Ciò ti consente di raggruppare i parametri, i log e gli eventi dei container in un'unica vista, in modo da offrirti una visibilità operativa più completa.

Gli eventi che puoi visualizzare sono quelli che Amazon ECS invia ad Amazon EventBridge. Per ulteriori informazioni, consulta Eventi di Amazon ECS.

Puoi scegliere di configurare i parametri delle prestazioni per cluster, attività o servizi. A seconda della risorsa scelta, vengono segnalati i seguenti eventi:

- Eventi di modifica dello stato delle istanze di container
- Eventi di operazioni di servizio
- Eventi di modifica dello stato dei processi

Configurazione di CloudWatch Container Insights per Amazon ECS

Puoi configurare Container Insights utilizzando la console Amazon ECS AWS CLI, l'API e gli SDK.

Utilizza la tabella seguente per determinare l'azione da intraprendere per aggiungere Container Insights.

Supporto del tagging per le risorse Amazon ECS

Attività	Console	AWS CLI	Azione API
Cambia l'impostazione predefinita per tutti gli utenti	Modifica delle impostazioni dell'account Amazon ECS	put-account-setting-default	PutAccountSettingDefault
Cambia l'impostazione predefinita per un utente specifico	Modifica delle impostazioni dell'account Amazon ECS	impostazione dell'account put	PutAccountImpostazione
Configura Container Insights per un cluster specifico	Creazione di un cluster Amazon ECS per il tipo di lancio Fargate Creazione di un cluster Amazon ECS	create-cluster UpdateCluster	CreateCluster UpdateCluster

Attività	Console	AWS CLI	Azione API
	per il tipo di lancio Amazon EC2 Aggiornamento di un cluster Amazon ECS		

> **Important**
>
> Per cluster contenenti processi o servizi che utilizzano il tipo di avvio EC2, le istanze di container devono eseguire la versione 1.29.0 o successiva dell'agente Amazon ECS. Per ulteriori informazioni, consulta Gestione delle istanze di container Amazon ECS Linux.

Autorizzazioni richieste per CloudWatch Container Insights per visualizzare gli eventi del ciclo di vita di Amazon ECS

È necessario configurare le autorizzazioni corrette, quindi è possibile configurare e visualizzare gli eventi nella console CloudWatch Container Insights. Per ulteriori informazioni, consulta gli eventi del ciclo di vita di Amazon ECS all'interno di Container Insights nella Amazon CloudWatch User Guide. Per ulteriori informazioni sulle politiche IAM per CloudWatch, consulta.AWS Identity and Access Management CloudWatch

Autorizzazioni necessarie per configurare Approfondimenti sui container per visualizzare gli eventi del ciclo di vita di Amazon ECS

Le seguenti autorizzazioni sono necessarie nel ruolo dell'attività per configurare gli eventi del ciclo di vita:

- eventi: PutRule
- eventi: PutTargets
- registri: Gruppo CreateLog

```
{
  "Version": "2012-10-17",
```

```
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:PutRule",
        "events:PutTargets",
        "logs:CreateLogGroup"
      ],
      "Resource": "*"
    }
  ]
}
```

Autorizzazioni necessarie per visualizzare gli eventi del ciclo di vita di Amazon ECS in Approfondimenti sui container

Per visualizzare gli eventi del ciclo di vita sono necessarie le seguenti autorizzazioni. Aggiungi le autorizzazioni seguenti come policy in linea al ruolo di esecuzione di attività. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

- eventi: DescribeRule
- eventi: ListTargets ByRule
- registri: gruppi DescribeLog

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "events:DescribeRule",
        "events:ListTargetsByRule",
        "logs:DescribeLogGroups"
      ],
      "Resource": "*"
    }
  ]
}
```

Determina lo stato delle attività di Amazon ECS utilizzando i controlli dello stato dei container

Quando crei una definizione di attività, puoi configurare un controllo dello stato dei tuoi contenitori. I controlli di integrità sono comandi eseguiti localmente su un contenitore e convalidano lo stato e la disponibilità delle applicazioni.

L'agente del container Amazon ECS monitora (e invia i report correlati) solo i controlli dell'integrità specificati nella definizione di attività. Amazon ECS non monitora i controlli dell'integrità di Docker incorporati in un'immagine del container e non specificati nella definizione del container. I parametri di controllo dello stato specificati in una definizione del container sostituiscono qualsiasi controllo dello stato Docker presente nell'immagine del container.

Quando un controllo di integrità è definito in una definizione di attività, il contenitore esegue il processo di controllo dello stato all'interno del contenitore, quindi valuta il codice di uscita per determinare lo stato dell'applicazione.

Il controllo dello stato di salute comprende i seguenti parametri:

- Comando: il comando che il contenitore esegue per determinare se è integro. La matrice di stringhe può iniziare con `CMD` per eseguire direttamente gli argomenti del comando oppure con `CMD-SHELL` per eseguire il comando con la shell predefinita del container.
- Intervallo: il periodo di tempo (in secondi) tra ogni controllo sanitario.
- Timeout: il periodo di tempo (in secondi) di attesa dell'esito positivo di un controllo sanitario prima che venga considerato un fallimento.
- Tentativi: il numero di volte in cui si ripete un controllo di integrità non riuscito prima che il contenitore venga considerato inintegro.
- Periodo di inizio: il periodo di prova opzionale per concedere ai container il tempo necessario per riavviarsi prima che i controlli di integrità non riusciti vengano conteggiati ai fini del numero massimo di nuovi tentativi.

Per informazioni su come specificare un controllo dello stato di salute nella definizione di un'attività, vedere. Controllo dello stato

Di seguito vengono descritti i possibili valori dello stato di salute di un contenitore:

- `HEALTHY`: il controllo dell'integrità del container è stato superato correttamente.

- `UNHEALTHY`: il controllo dell'integrità del container non è andato a buon fine.
- `UNKNOWN`: il controllo dell'integrità del container è in fase di valutazione, non è stato definito alcun controllo dell'integrità o Amazon ECS non dispone dello stato di integrità del container.

I comandi di controllo dello stato di salute vengono eseguiti sul contenitore. Pertanto è necessario includere i comandi nell'immagine del contenitore.

Il controllo dello stato si connette all'applicazione tramite l'interfaccia di loopback del contenitore all'indirizzo `localhost` o. `127.0.0.1` Un codice di uscita pari a `0` indica l'esito positivo, mentre un codice di uscita diverso da zero indica un errore.

Quando utilizzi i controlli dello stato dei contenitori, considera quanto segue:

- I controlli dello stato del container richiedono la versione 1.17.0 o successiva dell'agente del container Amazon ECS.
- I controlli dello stato dei container sono supportati per le attività Fargate se si utilizza una versione della piattaforma Linux `1.1.0` o superiore o una versione della piattaforma Windows o superiore `1.1.0`

In che modo Amazon ECS determina lo stato delle attività

I contenitori essenziali e con il comando Health Check nella definizione dell'attività sono gli unici considerati per determinare lo stato dell'attività.

Le seguenti regole vengono valutate nell'ordine:

1. Se lo stato di un contenitore essenziale è`UNHEALTHY`, lo stato dell'attività è`UNHEALTHY`.
2. Se lo stato di un contenitore essenziale è`UNKNOWN`, lo stato dell'attività è`UNKNOWN`.
3. Se lo stato di tutti i contenitori essenziali è`HEALTHY`, lo stato dell'attività è`HEALTHY`.

Considerate il seguente esempio di integrità di un'attività con 2 contenitori essenziali.

Contenitore 1: salute	Salute del contenitore 2	Stato dell'attività
`UNHEALTHY`	`UNKNOWN`	`UNHEALTHY`
`UNHEALTHY`	`HEALTHY`	`UNHEALTHY`

Contenitore 1: salute	Salute del contenitore 2	Stato dell'attività
HEALTHY	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY

Considerate il seguente esempio di integrità delle attività con 3 contenitori.

Salute del contenitore 1	Salute del contenitore 2	Salute del contenitore 3	Stato dell'attività
UNHEALTHY	UNKNOWN	UNKNOWN	UNHEALTHY
UNHEALTHY	UNKNOWN	HEALTHY	UNHEALTHY
UNHEALTHY	HEALTHY	HEALTHY	UNHEALTHY
HEALTHY	UNKNOWN	HEALTHY	UNKNOWN
HEALTHY	UNKNOWN	UNKNOWN	UNKNOWN
HEALTHY	HEALTHY	HEALTHY	HEALTHY

In che modo i controlli sanitari vengono influenzati dalle disconnessioni degli agenti

Se l'agente container Amazon ECS si disconnette dal servizio Amazon ECS, ciò non causerà la transizione del container a uno stato. UNHEALTHY Questo è stato progettato per garantire che i contenitori rimangano in funzione durante il riavvio degli agenti o la temporanea indisponibilità. Lo stato del controllo di integrità è l'ultima risposta ricevuta dall'agente Amazon ECS, quindi se il container è stato considerato HEALTHY prima della disconnessione, tale stato rimarrà invariato fino alla riconnessione dell'agente e all'esecuzione di un altro controllo dello stato. Non ci sono ipotesi sullo stato dei controlli dell'integrità del container.

Visualizzazione dello stato dei container Amazon ECS

Puoi visualizzare lo stato del contenitore nella console e utilizzare l'API nella `DescribeTasks` risposta. Per ulteriori informazioni, consulta il riferimento DescribeTasksall'API di Amazon Elastic Container Service.

Se utilizzi la registrazione per il tuo contenitore, ad esempio Amazon CloudWatch Logs, puoi configurare il comando health check per inoltrare l'output sullo stato del contenitore ai tuoi log. Assicurati di utilizzare `2&1` to catch sia `stdout` le `stderr` informazioni che.

```
"command": [
     "CMD-SHELL",
     "curl -f http://localhost/ >> /proc/1/fd/1 2>&1  || exit 1"
   ],
```

Monitora lo stato delle istanze dei container Amazon ECS

Amazon ECS offre il monitoraggio dell'integrità delle istanze di container. Puoi determinare rapidamente se Amazon ECS ha rilevato problemi che potrebbero impedire alle istanze di container di eseguire i container. Amazon ECS esegue controlli automatici su ogni istanza di container in esecuzione con la versione `1.57.0` o successiva dell'agente per identificare i problemi. Per ulteriori informazioni sulla verifica della versione dell'agente su un'istanza di container, consulta Aggiornamento dell'agente del container Amazon ECS.

È necessario utilizzare la AWS CLI versione `1.22.3` o successiva o la AWS CLI versione `2.3.6` o successiva. Per informazioni su come aggiornare il AWS CLI, vedere Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente versione 2.

Le verifiche dello stato vengono eseguite circa due volte al minuto e restituiscono un risultato positivo o negativo. Se vengono superate tutte le verifiche, lo stato complessivo dell'istanza sarà `OK`. Se invece una o più verifiche non vengono superate, lo stato complessivo sarà `IMPAIRED` (danneggiata). Le verifiche dello stato sono integrate nell'agente di container Amazon ECS, perciò non possono essere disattivate o eliminate. Puoi visualizzare i risultati delle verifiche dello stato per individuare problemi specifici e rilevabili. Per ulteriori informazioni, consulta the section called “Controllo dello stato”.

Esegui l'`DescribeContainerInstances`API con l'`CONTAINER_INSTANCE_HEALTH`opzione per recuperare lo stato dell'istanza del contenitore.

```
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"
 }]
}
```

Argomenti correlati

- Monitora Amazon ECS utilizzando CloudWatch

Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni

Amazon ECS si integra con AWS Distro per OpenTelemetry raccogliere dati di traccia dalla tua applicazione. Amazon ECS utilizza un contenitore AWS Distro for OpenTelemetry sidecar per raccogliere e indirizzare i dati di traccia. AWS X-Ray Per ulteriori informazioni, consulta Configurazione di AWS Distro for OpenTelemetry Collector in Amazon ECS. Puoi quindi utilizzarlo AWS X-Ray per identificare errori ed eccezioni, analizzare i rallentamenti delle prestazioni e i tempi di risposta.

Affinché AWS Distro for OpenTelemetry Collector possa inviare i dati di traccia a AWS X-Ray, l'applicazione deve essere configurata per creare i dati di traccia. Per ulteriori informazioni, consulta Strumentazione dell'applicazione per AWS X-Ray nella AWS X-Ray Guida per gli sviluppatori.

Autorizzazioni IAM richieste per AWS Distro per l'integrazione con OpenTelemetry AWS X-Ray

L'integrazione di Amazon ECS con AWS Distro for OpenTelemetry richiede la creazione di un ruolo di attività e la specificazione del ruolo nella definizione dell'attività. Ti consigliamo di configurare AWS Distro for OpenTelemetry sidecar per indirizzare i log dei container verso Logs. CloudWatch

> **Important**
>
> Se raccogli anche le metriche delle applicazioni utilizzando AWS Distro per l' OpenTelemetryintegrazione, assicurati che il ruolo Task IAM contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni.

Crea la seguente policy, quindi collegala al ruolo di esecuzione dell'attività.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
```

```
                "logs:DescribeLogGroups",
                "logs:PutRetentionPolicy",
                "xray:PutTraceSegments",
                "xray:PutTelemetryRecords",
                "xray:GetSamplingRules",
                "xray:GetSamplingTargets",
                "xray:GetSamplingStatisticSummaries",
                "ssm:GetParameters"
            ],
            "Resource": "*"
        }
    ]
}
```

6. Seleziona Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Specificare il AWS Distro for OpenTelemetry sidecar da AWS X-Ray integrare nella definizione dell'attività

La console Amazon ECS semplifica la creazione del contenitore AWS Distro for OpenTelemetry sidecar utilizzando l'opzione Use trace collection. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

Se non utilizzi la console Amazon ECS, puoi aggiungere il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente frammento di definizione dell'attività mostra la definizione del contenitore per l'aggiunta del AWS Distro for sidecar for integration. OpenTelemetry AWS X-Ray

```
{
 "family": "otel-using-xray",
 "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryXrayRole",
 "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
 "containerDefinitions": [{
   "name": "aws-otel-emitter",
   "image": "application-image",
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-create-group": "true",
     "awslogs-group": "/ecs/aws-otel-emitter",
     "awslogs-region": "us-east-1",
     "awslogs-stream-prefix": "ecs"
    }
   },
   "dependsOn": [{
    "containerName": "aws-otel-collector",
    "condition": "START"
   }]
  },
  {
   "name": "aws-otel-collector",
   "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
   "essential": true,
   "command": [
    "--config=/etc/ecs/otel-instance-metrics-config.yaml"
   ],
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-create-group": "True",
     "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
     "awslogs-region": "us-east-1",
     "awslogs-stream-prefix": "ecs"
    }
   }
  }
 ],
 "networkMode": "awsvpc",
 "requiresCompatibilities": [
  "FARGATE"
 ],
```

```
 "cpu": "1024",
 "memory": "3072"
}
```

Correla le prestazioni delle applicazioni Amazon ECS utilizzando i parametri delle applicazioni

Amazon ECS on Fargate supporta la raccolta di metriche dalle applicazioni in esecuzione su Fargate e l'esportazione su Amazon o CloudWatch Amazon Managed Service for Prometheus.

Puoi utilizzare i metadati raccolti per correlare i dati sulle prestazioni delle applicazioni con i dati dell'infrastruttura sottostante, riducendo il tempo medio necessario per risolvere il problema.

Amazon ECS utilizza un contenitore AWS Distro for OpenTelemetry sidecar per raccogliere e indirizzare i parametri dell'applicazione verso la destinazione. L'esperienza della console Amazon ECS semplifica il processo di aggiunta di questa integrazione durante la creazione delle definizioni delle attività.

Argomenti

- Esportazione dei parametri delle applicazioni su Amazon CloudWatch
- Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus

Esportazione dei parametri delle applicazioni su Amazon CloudWatch

Amazon ECS on Fargate supporta l'esportazione dei parametri delle applicazioni personalizzate su Amazon CloudWatch come parametri personalizzati. Questo viene fatto aggiungendo il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. La console Amazon ECS semplifica questo processo aggiungendo l'opzione Use metric collection durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

I parametri dell'applicazione vengono esportati in CloudWatch Logs con il nome del gruppo di log `/aws/ecs/application/metrics` e i parametri possono essere visualizzati nel namespace. `ECS/AWSOTel/Application` L'applicazione deve essere dotata dell'SDK. OpenTelemetry Per ulteriori informazioni, consulta Introduzione a AWS Distro for OpenTelemetry in the AWS Distro per la documentazione. OpenTelemetry

Considerazioni

Quando si utilizza l'integrazione di Amazon ECS on Fargate AWS con Distro OpenTelemetry per inviare i parametri delle applicazioni ad Amazon, è necessario considerare quanto segue. CloudWatch

- Questa integrazione invia solo i parametri dell'applicazione personalizzati a. CloudWatch Se desideri parametri a livello di attività, puoi attivare Container Insights nella configurazione del cluster Amazon ECS. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.
- La AWS distribuzione per l' OpenTelemetry integrazione è supportata per i carichi di lavoro Amazon ECS ospitati su Fargate e i carichi di lavoro Amazon ECS ospitati su istanze Amazon EC2. Al momento le istanze esterne non sono supportate.
- CloudWatch supporta un massimo di 30 dimensioni per metrica. Di default, Amazon ECS include le dimensioni `TaskARN`, `ClusterARN`, `LaunchType`, `TaskDefinitionFamily` e `TaskDefinitionRevision` nei parametri. Le restanti 25 dimensioni possono essere definite dalla tua applicazione. Se sono configurate più di 30 dimensioni, non è CloudWatch possibile visualizzarle. Quando ciò si verifica, le metriche dell'applicazione verranno visualizzate nello spazio dei nomi delle `ECS/AWSOTel/Application` CloudWatch metriche ma senza alcuna dimensione. Puoi strumentare la tua applicazione per aggiungere ulteriori dimensioni. Per ulteriori informazioni, consulta Utilizzo delle CloudWatch metriche con AWS Distro for in the Distro per la OpenTelemetry documentazione. AWS OpenTelemetry

Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon CloudWatch

L'integrazione di Amazon ECS con AWS Distro for OpenTelemetry richiede la creazione di un ruolo IAM dell'attività e la specificazione del ruolo nella definizione dell'attività. Consigliamo di configurare AWS Distro for OpenTelemetry sidecar anche per indirizzare i log dei container verso Logs, il che richiede la creazione e la specificazione di un ruolo IAM per CloudWatch l'esecuzione dell'attività e che venga specificato anche nella definizione dell'attività. La console Amazon ECS si occupa dell'esecuzione delle attività (ruolo IAM) per tuo conto, ma il ruolo IAM dell'attività deve essere creato manualmente e aggiunto alla definizione dell'attività. Per ulteriori informazioni sul ruolo IAM di esecuzione dei processi, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

> **Important**
>
> Se stai anche raccogliendo dati di tracciamento delle applicazioni utilizzando AWS Distro per OpenTelemetry l'integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni.
> Se l'applicazione richiede ulteriori autorizzazioni, è necessario aggiungerle a questa policy. Ciascuna definizione di attività può specificare un solo ruolo IAM del processo. Ad esempio, se utilizzi un file di configurazione personalizzato memorizzato in Systems Manager, devi aggiungere l'autorizzazione `ssm:GetParameters` a questa policy IAM.

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Seleziona Successivo.
6. Nella sezione Aggiungi autorizzazioni, cerca AWSDistroOpenTelemetryPolicyForXray, quindi seleziona la politica.
7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.

   a. Apri la sezione Imposta i limiti delle autorizzazioni, quindi scegli Usa un limite di autorizzazioni per controllare il numero massimo di autorizzazioni per i ruoli.

      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. Seleziona Successivo.
9. Inserisci il nome del ruolo o il suffisso del nome del ruolo per aiutarti a identificare lo scopo del ruolo.

> **Important**
>
> Quando assegnate un nome a un ruolo, tenete 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 entrambi **`PRODROLE`** e**`prodrole`**. Quando un nome di 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 dai 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 perché altre entità potrebbero 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, nelle sezioni Passo 1: Seleziona entità attendibili o Passaggio 2: Aggiungi autorizzazioni, scegli Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consulta la sezione Applicazione di tag alle risorse IAM nella Guida per l'utente di IAM.
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 Amazon ECS utilizzando la console.

Se non utilizzi la console Amazon ECS, puoi aggiungere manualmente il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente esempio di definizione delle attività mostra la definizione del contenitore per l'aggiunta di AWS Distro for OpenTelemetry sidecar for Amazon CloudWatch integration.

```
{
 "family": "otel-using-cloudwatch",
 "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryCloudWatchRole",
```

```
"executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
"containerDefinitions": [
   {
  "name": "aws-otel-emitter",
  "image": "application-image",
  "logConfiguration": {
   "logDriver": "awslogs",
   "options": {
    "awslogs-create-group": "true",
    "awslogs-group": "/ecs/aws-otel-emitter",
    "awslogs-region": "us-east-1",
    "awslogs-stream-prefix": "ecs"
   }
  },
  "dependsOn": [{
   "containerName": "aws-otel-collector",
   "condition": "START"
  }]
 },
 {
  "name": "aws-otel-collector",
  "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
  "essential": true,
  "command": [
   "--config=/etc/ecs/ecs-cloudwatch.yaml"
  ],
  "logConfiguration": {
   "logDriver": "awslogs",
   "options": {
    "awslogs-create-group": "True",
    "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
    "awslogs-region": "us-east-1",
    "awslogs-stream-prefix": "ecs"
   }
  }
 }
],
"networkMode": "awsvpc",
"requiresCompatibilities": [
 "FARGATE"
],
"cpu": "1024",
"memory": "3072"
```

```
}
```

Esportazione di parametri delle applicazioni in Amazon Managed Service for Prometheus

Amazon ECS supporta l'esportazione di parametri di CPU, memoria, rete e archiviazione a livello di processo e i parametri delle applicazioni personalizzate in Amazon Managed Service for Prometheus. Questo viene fatto aggiungendo il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. La console Amazon ECS semplifica questo processo aggiungendo l'opzione Use metric collection durante la creazione di una nuova definizione di attività. Per ulteriori informazioni, consulta Creazione di una definizione di attività Amazon ECS utilizzando la console.

I parametri vengono esportati in Amazon Managed Service for Prometheus e possono essere visualizzati utilizzando il pannello di controllo di Amazon Managed Grafana. L'applicazione deve essere dotata di strumenti con le librerie Prometheus o con l'SDK. OpenTelemetry Per ulteriori informazioni sulla strumentazione dell'applicazione con l' OpenTelemetry SDK, consulta Introduzione a Distro per AWS la documentazione nella versione Distro. OpenTelemetry AWS OpenTelemetry

Quando si utilizzano le librerie Prometheus, l'applicazione deve esporre un endpoint `/metrics` utilizzato per lo scraping dei dati dei parametri. Per ulteriori informazioni sulla strumentazione della tua applicazione con le librerie Prometheus, consulta Librerie client Prometheus nella documentazione di Prometheus.

Considerazioni

Quando si utilizza l'integrazione di Amazon ECS on Fargate AWS con Distro OpenTelemetry per inviare i parametri delle applicazioni ad Amazon Managed Service for Prometheus, è necessario considerare quanto segue.

- La AWS distribuzione per l' OpenTelemetry integrazione è supportata per i carichi di lavoro Amazon ECS ospitati su Fargate e i carichi di lavoro Amazon ECS ospitati su istanze Amazon EC2. Attualmente le istanze esterne non sono supportate.
- Per impostazione predefinita, AWS Distro for OpenTelemetry include tutte le dimensioni a livello di attività disponibili per le metriche dell'applicazione durante l'esportazione in Amazon Managed Service for Prometheus. Puoi anche strumentare la tua applicazione per aggiungere ulteriori dimensioni. Per ulteriori informazioni, consulta la sezione Guida introduttiva a Prometheus Remote Write Exporter for Amazon Managed Service for Prometheus in the Distro per la documentazione. AWS OpenTelemetry

Autorizzazioni IAM richieste per AWS Distro per OpenTelemetry l'integrazione con Amazon Managed Service for Prometheus

L'integrazione di Amazon ECS con Amazon Managed Service for Prometheus utilizzando Distro OpenTelemetry for sidecar richiede AWS la creazione di un ruolo IAM dell'attività e la specificazione del ruolo nella definizione dell'attività. Questo ruolo IAM del processo deve essere creato manualmente utilizzando la procedura riportata di seguito prima di registrare la definizione di attività.

Consigliamo di configurare AWS Distro for OpenTelemetry sidecar anche per indirizzare i log dei container verso Logs, il che richiede la creazione di un ruolo IAM per CloudWatch l'esecuzione delle attività e la specificazione anche nella definizione dell'attività. La console Amazon ECS si occupa dell'esecuzione delle attività (ruolo IAM) per tuo conto, ma il ruolo IAM dell'attività deve essere creato manualmente. Per ulteriori informazioni sulla creazione di un ruolo IAM di creazione dei processi, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

> **Important**
>
> Se stai anche raccogliendo dati di tracciamento delle applicazioni utilizzando AWS Distro per OpenTelemetry l'integrazione, assicurati che il ruolo IAM dell'attività contenga anche le autorizzazioni necessarie per tale integrazione. Per ulteriori informazioni, consulta Identifica le opportunità di ottimizzazione di Amazon ECS utilizzando i dati di tracciamento delle applicazioni.

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Seleziona Successivo.
6. Nella sezione Aggiungi autorizzazioni, cerca AmazonPrometheusRemoteWriteAccess, quindi seleziona la politica.

7. (Facoltativo) Impostare un limite delle autorizzazioni. Questa è una caratteristica avanzata disponibile per i ruoli di servizio, ma non per i ruoli collegati ai servizi.

   a. Apri la sezione Imposta i limiti delle autorizzazioni, quindi scegli Usa un limite di autorizzazioni per controllare il numero massimo di autorizzazioni per i ruoli.

      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. Seleziona Successivo.
9. Inserisci il nome del ruolo o il suffisso del nome del ruolo per aiutarti a identificare lo scopo del ruolo.

   > **Important**
   >
   > Quando assegnate un nome a un ruolo, tenete 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 entrambi **`PRODROLE`** e**`prodrole`**. Quando un nome di 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 dai 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 perché altre entità potrebbero 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, nelle sezioni Passo 1: Seleziona entità attendibili o Passaggio 2: Aggiungi autorizzazioni, scegli Modifica.
12. (Facoltativo) Per facilitare l'identificazione, l'organizzazione o la ricerca del ruolo, aggiungi tag come coppie chiave-valore. Per ulteriori informazioni sull'utilizzo di tag in IAM, consulta la sezione Applicazione di tag alle risorse IAM nella Guida per l'utente di IAM.
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 Amazon ECS utilizzando la console.

Se non utilizzi la console Amazon ECS, puoi aggiungere manualmente il contenitore AWS Distro for OpenTelemetry sidecar alla definizione dell'attività. Il seguente esempio di definizione di attività mostra la definizione del contenitore per l'aggiunta di AWS Distro for OpenTelemetry sidecar per l'integrazione con Amazon Managed Service for Prometheus.

```
{
 "family": "otel-using-cloudwatch",
 "taskRoleArn": "arn:aws:iam::111122223333:role/AmazonECS_OpenTelemetryCloudWatchRole",
 "executionRoleArn": "arn:aws:iam::111122223333:role/ecsTaskExecutionRole",
 "containerDefinitions": [{
   "name": "aws-otel-emitter",
   "image": "application-image",
   "logConfiguration": {
    "logDriver": "awslogs",
    "options": {
     "awslogs-create-group": "true",
     "awslogs-group": "/ecs/aws-otel-emitter",
     "awslogs-region": "aws-region",
     "awslogs-stream-prefix": "ecs"
    }
   },
   "dependsOn": [{
    "containerName": "aws-otel-collector",
    "condition": "START"
   }]
  },
  {
   "name": "aws-otel-collector",
   "image": "public.ecr.aws/aws-observability/aws-otel-collector:v0.30.0",
   "essential": true,
   "command": [
    "--config=/etc/ecs/ecs-amp.yaml"
   ],
   "environment": [{
    "name": "AWS_PROMETHEUS_ENDPOINT",
    "value": "https://aps-workspaces.aws-region.amazonaws.com/workspaces/
ws-a1b2c3d4-5678-90ab-cdef-EXAMPLE11111/api/v1/remote_write"
```

```
    }],
    "logConfiguration": {
     "logDriver": "awslogs",
     "options": {
      "awslogs-create-group": "True",
      "awslogs-group": "/ecs/ecs-aws-otel-sidecar-collector",
      "awslogs-region": "aws-region",
      "awslogs-stream-prefix": "ecs"
     }
    }
   }
  ],
  "networkMode": "awsvpc",
  "requiresCompatibilities": [
   "FARGATE"
  ],
  "cpu": "1024",
  "memory": "3072"
}
```

Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail

Amazon ECS è integrato con AWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon ECS. CloudTrail acquisisce tutte le chiamate API per Amazon ECS come eventi, incluse le chiamate dalla console Amazon ECS e le chiamate di codice alle operazioni dell'API Amazon ECS. Per proteggere il tuo VPC, le richieste che vengono rifiutate da una policy sugli endpoint VPC, ma che altrimenti sarebbero state consentite, non vengono registrate in. CloudTrail

Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per Amazon ECS. Se non configuri un percorso, puoi comunque visualizzare gli eventi più recenti nella CloudTrail console nella cronologia degli eventi. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon ECS, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

Per ulteriori informazioni, consulta la Guida per l'utente AWS CloudTrail.

Informazioni su Amazon ECS in CloudTrail

CloudTrail è attivata nel tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in Amazon ECS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon ECS, crea un percorso da CloudTrail utilizzare per distribuire i file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consultare:

- Panoramica della creazione di un trail
- CloudTrail Servizi e integrazioni supportati
- Configurazione delle notifiche Amazon SNS per CloudTrail
- Ricezione di file di CloudTrail registro da più regioni e ricezione di file di CloudTrail registro da più account

Tutte le azioni di Amazon ECS vengono registrate CloudTrail e documentate nell'Amazon Elastic Container Service API Reference. Ad esempio, le chiamate alle `CreateService` `DeleteCluster` sezioni `RunTask` e generano voci nei file di registro. CloudTrail

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 con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro AWS servizio.

Per ulteriori informazioni, vedete l'elemento CloudTrail userIdentity.

Informazioni sulle voci del file di log Amazon ECS

Un trail è una configurazione che consente la distribuzione di eventi come file di log in un bucket Amazon S3 specificato dall'utente. CloudTrail i file di registro contengono una o più voci di registro. Un evento rappresenta una singola richiesta proveniente da qualsiasi fonte e include informazioni sull'azione richiesta, la data e l'ora dell'azione, i parametri della richiesta e così via. CloudTrail i file di registro non sono una traccia ordinata dello stack delle chiamate API pubbliche, quindi non vengono visualizzati in un ordine specifico.

> **Note**
>
> Questi esempi sono stati formattati 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

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`CreateCluster`azione:

```
{
    "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"
}
```

Monitora i carichi di lavoro utilizzando i metadati Amazon ECS

È possibile utilizzare i metadati delle attività e dei contenitori per risolvere i problemi dei carichi di lavoro e apportare modifiche alla configurazione in base all'ambiente di runtime.

I metadati includono le seguenti categorie:

- Attributi a livello di attività che forniscono informazioni sulla posizione in cui l'attività è in esecuzione.
- Attributi a livello di contenitore che forniscono l'ID Docker, il nome e i dettagli dell'immagine.

  Ciò fornisce visibilità all'interno del contenitore.
- Impostazioni di rete come indirizzi IP, sottoreti e modalità di rete.

  Ciò facilita la configurazione e la risoluzione dei problemi della rete.

- Stato e integrità dell'attività

  Ciò consente di sapere se le attività sono in esecuzione.

È possibile visualizzare i metadati con uno dei seguenti metodi:

- File di metadati del container

  A partire dalla versione 1.15.0 dell'agente di container di Amazon ECS, vari metadati di container sono disponibili all'interno dei contenitori o dell'istanza di container host. Attivando questa caratteristica, è possibile eseguire una query delle informazioni su un'attività, un container e un'istanza di container all'interno del container stesso o dell'istanza di container host. Il file di metadati viene creato sull'istanza host e montato nel container come volume Docker e pertanto non è disponibile quando un processo è ospitato su AWS Fargate.
- Endpoint di metadati delle attività

  L'agente del container Amazon ECS inserisce una variabile di ambiente in ogni container denominata endpoint dei metadati dei processi che fornisce vari metadati di processi e statistiche Docker al container.
- Introspezione del contenitore

  L'agente del container di Amazon ECS fornisce un'operazione API per raccogliere i dettagli sull'istanza di container su cui è in esecuzione l'agente e sui processi associati in esecuzione su tale istanza.

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 di attività e immagini in Amazon ECS.

Argomenti

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 percorse dell'host di Linux presuppone che venga utilizzato il percorso di montaggio della directory dei dati predefinito (`/var/lib/ecs/data`) all'avvio dell'agente. Se non utilizzi l'AMI ottimizzata per Amazon ECS (o il pacchetto `ecs-init` per avviare e mantenere l'agente di container), assicurati di impostare la variabile di configurazione dell'agente `ECS_HOST_DATA_DIR` sul percorso dell'host in cui è ubicato il file di stato dell'agente del container. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.
  - Percorso container: `/opt/ecs/metadata/`*`random_ID`*`/ecs-container-metadata.json`
- Per le istanze Windows:
  - Percorso host: `C:\ProgramData\Amazon\ECS\data\metadata\`*`task_id`*`\`*`container_name`*`\ecs-container-metadata.json`
  - Percorso container: `C:\ProgramData\Amazon\ECS\metadata\`*`random_ID`*`\ecs-container-metadata.json`

Tuttavia, per semplificare l'accesso, l'ubicazione del file di metadati di container è impostata sulla variabile di ambiente `ECS_CONTAINER_METADATA_FILE` all'interno del container. Puoi leggere il file contenuti dall'interno del container attraverso il comando seguente:

- Per le istanze Linux:

```
cat $ECS_CONTAINER_METADATA_FILE
```

- Per le istanze Windows (): PowerShell

```
Get-Content -path $env:ECS_CONTAINER_METADATA_FILE
```

Attivazione dei metadati dei contenitori Amazon ECS

È possibile attivare i metadati di container a livello di istanza di container impostando la variabile dell'agente di container `ECS_ENABLE_CONTAINER_METADATA` su `true`. Puoi impostare questa variabile nel file di configurazione `/etc/ecs/ecs.config` e riavviare l'agente. Puoi inoltre impostarla come una variabile di ambiente Docker al runtime quando viene avviato l'agente di container. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Se l'opzione `ECS_ENABLE_CONTAINER_METADATA` è impostata su `true` quando viene avviato l'agente, i file di metadati vengono creati per tutti i container creati da quel punto in avanti. L'agente di container di Amazon ECS non è in grado di creare file di metadati per container creati prima dell'impostazione della variabile dell'agente di container `ECS_ENABLE_CONTAINER_METADATA` su `true`. Per assicurarsi che tutti i container ricevano i file di metadati, è necessario impostare questa variabile dell'agente all'avvio dell'istanza di container. Di seguito è riportato uno script di dati utente di esempio che imposterà questa variabile e registrerà l'istanza di container con il cluster.

```
#!/bin/bash
cat <<'EOF' >> /etc/ecs/ecs.config
ECS_CLUSTER=your_cluster_name
ECS_ENABLE_CONTAINER_METADATA=true
EOF
```

Formato di file di metadati del contenitore Amazon ECS

Le informazioni seguenti vengono memorizzate nel file JSON di metadati del container.

`Cluster`

Il nome del cluster su cui viene eseguita l'attività di container.

`ContainerInstanceARN`

L'Amazon Resource Name (ARN) completo dell'istanza di container dell'host.

`TaskARN`

L'Amazon Resource Name (ARN) completo dell'attività a cui appartiene il container.

`TaskDefinitionFamily`

Il nome della famiglia di definizioni di processo utilizzata dal container.

`TaskDefinitionRevision`

La revisione della definizione di attività utilizzata dal container.

`ContainerID`

L'ID del container Docker (non si tratta dell'ID del container di Amazon ECS) per il container.

`ContainerName`

Il nome del container dalla definizione di attività di Amazon ECS per il container.

`DockerContainerName`

Il nome del container che il daemon Docker utilizza per il container (ad esempio, il nome che viene visualizzato nell'output del comando docker ps).

`ImageID`

Il digest SHA dell'immagine Docker utilizzato per avviare il container.

`ImageName`

Il nome e il tag dell'immagine Docker utilizzati per avviare il container.

`PortMappings`

Qualsiasi mappatura delle porte associate al container.

`ContainerPort`

La porta del container esposto.

`HostPort`

La porta dell'istanza di container host esposto.

`BindIp`

L'indirizzo IP di associazione assegnato al container da Docker. Questo indirizzo IP viene applicato solo con la modalità di rete `bridge` ed è accessibile solo dall'istanza di container.

`Protocol`

Il protocollo di rete utilizzato per la mappatura delle porte.

`Networks`

Le modalità di rete e l'indirizzo IP del container.

`NetworkMode`

Le modalità di rete dell'attività a cui appartiene il container.

`IPv4Addresses`

Gli indirizzi IP associati con il container.

> **Important**
>
> Se l'attività utilizza la modalità di rete `awsvpc`, l'indirizzo IP del container non verrà restituito. In questo caso, puoi recuperare l'indirizzo IP leggendo il file /etc/hosts con il comando seguente:
>
> ```
> tail -1 /etc/hosts | awk '{print $1}'
> ```

`MetadataFileStatus`

Lo stato del file di metadati. Quando lo stato è `READY`, il file di metadati è corrente e completo. Se il file non è ancora pronto (ad esempio, nel momento in cui l'attività viene avviata), è disponibile una versione troncata del formato di file. Per evitare una probabile race condition in cui il container viene avviato, ma i metadati non sono ancora stati scritti, puoi analizzare il file di metadati e attendere che questo parametro sia impostato su `READY` prima di dipendere dai metadati. In genere, ciò è disponibile in meno di 1 secondo dall'avvio del container.

`AvailabilityZone`

Zona di disponibilità in cui risiede l'istanza di container host.

`HostPrivateIPv4Address`

L'indirizzo IP privato per l'attività a cui appartiene il container.

`HostPublicIPv4Address`

L'indirizzo IP pubblico per l'attività a cui appartiene il container.

Example File di metadati del container di Amazon ECS (**READY**)

L'esempio seguente mostra un file di metadati di container in stato READY.

```
{
    "Cluster": "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/2b88376d-
aba3-4950-9ddf-bcb0f388a40c",
    "TaskDefinitionFamily": "console-sample-app-static",
    "TaskDefinitionRevision": "1",
    "ContainerID": "aec2557997f4eed9b280c2efd7afccdcedfda4ac399f7480cae870cfc7e163fd",
    "ContainerName": "simple-app",
    "CreatedAt": "2023-10-08T20:09:11.44527186Z",
    "StartedAt": "2023-10-08T20:09:11.44527186Z",
    "DockerContainerName": "/ecs-console-sample-app-static-1-simple-app-
e4e8e495e8baa5de1a00",
    "ImageID":
 "sha256:2ae34abc2ed0a22e280d17e13f9c01aaf725688b09b7a1525d1a2750e2c0d1de",
    "ImageName": "httpd:2.4",
    "PortMappings": [
        {
            "ContainerPort": 80,
            "HostPort": 80,
            "BindIp": "0.0.0.0",
            "Protocol": "tcp"
        }
    ],
    "Networks": [
        {
            "NetworkMode": "bridge",
            "IPv4Addresses": ["192.0.2.0"]
        }
    ],
    "MetadataFileStatus": "READY",
    "AvailabilityZone": "us-east-1b",
    "HostPrivateIPv4Address": "192.0.2.0",
    "HostPublicIPv4Address": "203.0.113.0"
}
```

Example File di metadati di container Amazon ECS incompleto (non ancora **READY**)

L'esempio seguente mostra un file di metadati di container che non ha ancora raggiunto lo stato READY. Le informazioni incluse nel file sono limitate a pochi parametri noti dalla definizione di attività. In genere, il file di metadati di container è disponibile entro 1 secondo dopo l'avvio del container.

```
{
    "Cluster": "default",
    "ContainerInstanceARN": "arn:aws:ecs:us-west-2:012345678910:container-instance/
default/1f73d099-b914-411c-a9ff-81633b7741dd",
    "TaskARN": "arn:aws:ecs:us-west-2:012345678910:task/default/
d90675f8-1a98-444b-805b-3d9cabb6fcd4",
    "ContainerName": "metadata"
}
```

Metadati delle attività disponibili per le attività di Amazon ECS su EC2

L'agente del container di Amazon ECS fornisce un metodo per recuperare vari metadati dei processi e statistiche Docker. Questo metodo è noto come endpoint dei metadati dei processi. Sono disponibili le seguenti versioni:

- Endpoint dei metadati dei processi versione 4: fornisce una varietà di metadati e statistiche Docker ai container. Può anche fornire dati sulla velocità di rete. Disponibile per processi Amazon ECS lanciati su istanze di Amazon EC2 Linux che eseguono almeno la versione `1.39.0` dell'agente del container Amazon ECS. Per istanze Windows di Amazon EC2 che utilizzano la modalità di rete `awsvpc`, l'agente del container Amazon ECS deve essere almeno versione `1.54.0`. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 4.
- Endpoint dei metadati dei processi versione 3: fornisce una varietà di metadati e statistiche Docker ai container. Disponibile per processi Amazon ECS lanciati su istanze di Amazon EC2 Linux che eseguono almeno la versione `1.21.0` dell'agente del container Amazon ECS. Per istanze Windows di Amazon EC2 che utilizzano la modalità di rete `awsvpc`, l'agente del container Amazon ECS deve essere almeno versione `1.54.0`. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 3.
- Endpoint dei metadati dei processi versione 2: disponibile per processi Amazon ECS lanciati su istanze di Amazon EC2 Linux che eseguono almeno la versione `1.17.0` dell'agente del container Amazon ECS. Per istanze Windows di Amazon EC2 che utilizzano la modalità di rete `awsvpc`, l'agente del container Amazon ECS deve essere almeno versione `1.54.0`. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 2.

Se l'attività Amazon ECS è ospitata su Amazon EC2, puoi anche accedere ai metadati dell'host delle attività utilizzando l'endpoint del servizio di metadati di istanza (IMDS). Il comando seguente, se eseguito dall'istanza che ospita l'attività, elenca l'ID dell'istanza host.

```
curl http://169.254.169.254/latest/meta-data/instance-id
```

Le informazioni che è possibile ottenere dall'endpoint sono suddivise in categorie, ad esempio *instance-id*. Per ulteriori informazioni sulle diverse categorie di metadati delle istanze host che è possibile ottenere utilizzando l'endpoint, consulta Categorie di metadati dell'istanza.

Endpoint di metadati delle attività Amazon ECS versione 4

L'agente del container Amazon ECS inserisce una variabile di ambiente in ogni container denominata endpoint dei metadati dei processi che fornisce vari metadati di processi e statistiche Docker al container.

I metadati delle attività e le statistiche sulla velocità di rete vengono inviati a CloudWatch Container Insights e possono essere visualizzati in. AWS Management Console Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.

> **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 inserita di default nei container dei processi Amazon ECS lanciati su istanze Linux di Amazon EC2 che eseguono almeno la versione `1.39.0` dell'agente del container Amazon ECS. Per istanze Windows di Amazon EC2 che utilizzano la modalità di rete `awsvpc`, l'agente del container Amazon ECS deve essere almeno versione `1.54.0`. Per ulteriori informazioni, consulta Gestione delle istanze di container Amazon ECS Linux.

> **Note**
>
> Questa caratteristica può essere supportata sulle istanze Amazon EC2 utilizzando versioni precedenti dell'agente di container Amazon ECS aggiornando l'agente alla versione più

recente. Per ulteriori informazioni, consulta Aggiornamento dell'agente del container Amazon ECS.

Percorsi dell'endpoint dei metadati dei processi versione 4

Per i container sono disponibili i seguenti endpoint dei metadati dei processi:

`${ECS_CONTAINER_METADATA_URI_V4}`

Questo percorso restituisce il file JSON dei metadati per il container.

`${ECS_CONTAINER_METADATA_URI_V4}/task`

Questo percorso restituisce i metadati per il processo, compreso un elenco degli ID container e i nomi di tutti i container associati al processo. Per ulteriori informazioni sulle risposte per questo endpoint, consulta Risposta JSON V4 dei metadati delle attività di Amazon ECS.

`${ECS_CONTAINER_METADATA_URI_V4}/taskWithTags`

Questo percorso restituisce i metadati per il processo incluso nell'endpoint `/task` oltre ai tag dell'istanza di processo e container che possono essere recuperati utilizzando l'API `ListTagsForResource`. Eventuali errori ricevuti durante il recupero dei metadati del tag saranno inclusi nel campo `Errors` della risposta.

> **Note**
>
> Il campo `Errors` è presente solo nella risposta per i processi ospitati su istanze Linux di Amazon EC2 che eseguono almeno la versione `1.50.0` dell'agente del container. Per istanze Windows di Amazon EC2 che utilizzano la modalità di rete `awsvpc`, l'agente del container Amazon ECS deve essere almeno versione `1.54.0`.
> Questo endpoint richiede l'autorizzazione `ecs.ListTagsForResource`.

`${ECS_CONTAINER_METADATA_URI_V4}/stats`

Questo percorso restituisce le statistiche Docker per il container specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta ContainerStatsla documentazione dell'API Docker.

Per i processi Amazon ECS che utilizzano le modalità di rete `awsvpc` o `bridge` ospitate su istanze Linux Amazon EC2 che eseguono almeno la versione `1.43.0` dell'agente del container,

nella risposta saranno presenti anche altre statistiche 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 ContainerStatsla documentazione dell'API Docker.

Per i processi Amazon ECS che utilizzano le modalità di rete `awsvpc` o `bridge` ospitate su istanze Linux Amazon EC2 che eseguono almeno la versione `1.43.0` dell'agente del container, nella risposta saranno presenti anche altre statistiche sulla velocità di rete. Per tutti gli altri processi, la risposta includerà solo le statistiche cumulative di rete.

Risposta JSON V4 dei metadati delle attività di Amazon ECS

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`${ECS_CONTAINER_METADATA_URI_V4}/task`) restituisce le seguenti informazioni. Oltre ai metadati per ogni container all'interno del processo, sono inclusi anche i metadati associati al processo.

`Cluster`

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

`ServiceName`

Il nome del servizio a cui appartiene l'attività. ServiceName verrà visualizzato per le istanze di container Amazon EC2 e Amazon ECS Anywhere se l'attività è associata a un servizio.

> **Note**
>
> Questi metadati `ServiceName` sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione `1.63.1` o successiva.

`VPCID`

L'ID VPC dell'istanza di container di Amazon EC2. Questo campo viene visualizzato solo per le istanze Amazon EC2.

> **Note**
>
> Questi metadati `VPCID` sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione `1.63.1` o successiva.

`TaskARN`

L'Amazon Resource Name (ARN) completo 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 per le risorse specificati a livello di attività, ad esempio CPU (espressa in vCPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omesso.

`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 utilizzi i provider di capacità del cluster, questo indica se il processo utilizza l'infrastruttura Fargate o EC2.

> **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) p 1.0.0 o successiva (Windows).

`Containers`

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

Quando utilizzi Fargate, l'ID è un valore esadecimale a 32 cifre seguito da un numero di 10 cifre.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omesso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omesso.

`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 omesso.

`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 omesso.

`CreatedAt`

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omesso.

`StartedAt`

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omesso.

`FinishedAt`

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omesso.

`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`

L'Amazon Resource Name (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 omesso.

`ExecutionStoppedAt`

Il timestamp del momento in cui le attività `DesiredStatus` sono passate a essere `STOPPED`. Questo si verifica quando un container fondamentale passa allo stato `STOPPED`.

Esempi di metadati delle attività di Amazon ECS v4

Gli esempi seguenti mostrano gli output di esempio da ognuno degli endpoint dei metadati dei processi.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output.


```
{
    "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 i processi Amazon ECS che utilizzano le modalità di rete `awsvpc` o `bridge` ospitate su istanze Linux Amazon EC2 che eseguono almeno la versione `1.43.0` dell'agente del container, nella risposta saranno presenti anche altre statistiche 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 processo Amazon ECS su Amazon EC2 che utilizza la modalità di rete `bridge`.

```
{
    "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 dei processi

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-a6bcc3dbadfacfe85300",
        "id": "01999f2e5c6cf4df3873f28950e6278813408f281c54778efec860d0caad4854",
        "networks": {
            "eth0": {
                "rx_bytes": 84,
                "rx_packets": 2,
                "rx_errors": 0,
                "rx_dropped": 0,
                "tx_bytes": 84,
                "tx_packets": 2,
                "tx_errors": 0,
                "tx_dropped": 0
            }
        },
        "network_rate_stats": {
            "rx_bytes_per_sec": 0,
            "tx_bytes_per_sec": 0
        }
    },
```

```
    "5fc21e5b015f899d22618f8aede80b6d70d71b2a75465ea49d9462c8f3d2d3af": {
        "read": "2020-10-02T00:51:32.512771349Z",
        "preread": "2020-10-02T00:51:31.510597736Z",
        "pids_stats": {
            "current": 3
        },
        "blkio_stats": {
            "io_service_bytes_recursive": [

            ],
            "io_serviced_recursive": [

            ],
            "io_queue_recursive": [

            ],
            "io_service_time_recursive": [

            ],
            "io_wait_time_recursive": [

            ],
            "io_merged_recursive": [

            ],
            "io_time_recursive": [

            ],
            "sectors_recursive": [

            ]
        },
        "num_procs": 0,
        "storage_stats": {

        },
        "cpu_stats": {
            "cpu_usage": {
                "total_usage": 379075681,
                "percpu_usage": [
                    191355275,
                    187720406
                ],
                "usage_in_kernelmode": 60000000,
```

```
                "usage_in_usermode": 310000000
            },
            "system_cpu_usage": 13977800000000,
            "online_cpus": 2,
            "throttling_data": {
                "periods": 0,
                "throttled_periods": 0,
                "throttled_time": 0
            }
        },
        "precpu_stats": {
            "cpu_usage": {
                "total_usage": 378825197,
                "percpu_usage": [
                    191104791,
                    187720406
                ],
                "usage_in_kernelmode": 60000000,
                "usage_in_usermode": 310000000
            },
            "system_cpu_usage": 13975800000000,
            "online_cpus": 2,
            "throttling_data": {
                "periods": 0,
                "throttled_periods": 0,
                "throttled_time": 0
            }
        },
        "memory_stats": {
            "usage": 1814528,
            "max_usage": 6299648,
            "stats": {
                "active_anon": 606208,
                "active_file": 0,
                "cache": 0,
                "dirty": 0,
                "hierarchical_memory_limit": 134217728,
                "hierarchical_memsw_limit": 268435456,
                "inactive_anon": 0,
                "inactive_file": 0,
                "mapped_file": 0,
                "pgfault": 5377,
                "pgmajfault": 0,
                "pgpgin": 3613,
```

```
                "pgpgout": 3465,
                "rss": 606208,
                "rss_huge": 0,
                "total_active_anon": 606208,
                "total_active_file": 0,
                "total_cache": 0,
                "total_dirty": 0,
                "total_inactive_anon": 0,
                "total_inactive_file": 0,
                "total_mapped_file": 0,
                "total_pgfault": 5377,
                "total_pgmajfault": 0,
                "total_pgpgin": 3613,
                "total_pgpgout": 3465,
                "total_rss": 606208,
                "total_rss_huge": 0,
                "total_unevictable": 0,
                "total_writeback": 0,
                "unevictable": 0,
                "writeback": 0
            },
            "limit": 134217728
        },
        "name": "/ecs-curltest-26-curl-c2e5f6e0cf91b0bead01",
        "id": "5fc21e5b015f899d22618f8aede80b6d70d71b2a75465ea49d9462c8f3d2d3af",
        "networks": {
            "eth0": {
                "rx_bytes": 84,
                "rx_packets": 2,
                "rx_errors": 0,
                "rx_dropped": 0,
                "tx_bytes": 84,
                "tx_packets": 2,
                "tx_errors": 0,
                "tx_dropped": 0
            }
        },
        "network_rate_stats": {
            "rx_bytes_per_sec": 0,
            "tx_bytes_per_sec": 0
        }
    }
}
```

Endpoint di metadati delle attività Amazon ECS versione 3

> **Important**
>
> L'endpoint dei metadati delle attività versione 3 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta the section called “Endpoint metadati delle attività versione 4”.
> Se utilizzi attività di Amazon ECS ospitate su AWS Fargate, consulta la versione 3 dell'endpoint di metadati Task nella Guida per l'utente di Amazon Elastic Container Service per. AWS 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`.

L'endpoint dei metadati dei processi versione 3 è abilitato di default per i processi che utilizzano il tipo di avvio Fargate sulla versione della piattaforma v1.3.0 o successiva e per processi che utilizzano il tipo di avvio EC2 e che vengono avviate sull'infrastruttura Linux di Amazon EC2 che esegue almeno la versione 1.21.0 dell'agente del container Amazon ECS o sull'infrastruttura Windows di Amazon EC2. che esegue almeno la versione `1.54.0` dell'agente del container Amazon ECS e utilizzano la modalità di rete `awsvpc`. Per ulteriori informazioni, consulta Gestione delle istanze di container Amazon ECS Linux.

Questa caratteristica può essere supportata sulle istanze di container precedenti aggiornando l'agente alla versione più recente. Per ulteriori informazioni, consulta Aggiornamento dell'agente del container Amazon ECS.

> **Important**
>
> Per attività che utilizzano il tipo di avvio Fargate e versioni della piattaforma precedenti alla v1.3.0, è supportato l'endpoint dei metadati dei processi versione 2. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 2.

Percorsi della versione 3 dell'endpoint Task Metadata

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 un file JSON con i metadati per l'attività, compreso un elenco degli ID container e i nomi di tutti i container associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta Risposta JSON v3 dei metadati delle attività di Amazon ECS.

`${ECS_CONTAINER_METADATA_URI}/taskWithTags`

Questo percorso restituisce i metadati per il processo incluso nell'endpoint `/task` oltre ai tag dell'istanza di processo e container che possono essere recuperati utilizzando l'API `ListTagsForResource`.

`${ECS_CONTAINER_METADATA_URI}/stats`

Questo percorso restituisce un file JSON con le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta ContainerStatsla 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 ContainerStatsla documentazione dell'API Docker.

Risposta JSON v3 dei metadati delle attività di Amazon ECS

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`${ECS_CONTAINER_METADATA_URI}/task`) restituisce le seguenti informazioni.

`Cluster`

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

`TaskARN`

L'Amazon Resource Name (ARN) completo 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 per le risorse specificati a livello di attività, ad esempio CPU (espressa in vCPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omesso.

`PullStartedAt`

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

`PullStoppedAt`

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

`AvailabilityZone`

La zona di disponibilità in cui si trova l'attività.

> **Note**
>
> I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

`Containers`

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omesso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omesso.

`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 omesso.

`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 omesso.

`CreatedAt`

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omesso.

`StartedAt`

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omesso.

`FinishedAt`

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omesso.

`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 omesso.

`ClockDrift`

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta Impostare l'ora per l'istanza Linux nella Guida per l'utente di Amazon EC2 per le istanze Linux.

`ReferenceTime`

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio `2021-09-07T16:57:44Z`.

`ClockErrorBound`

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

`ClockSynchronizationStatus`

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono `SYNCHRONIZED` e `NOT_SYNCHRONIZED`.

`ExecutionStoppedAt`

Il timestamp del momento in cui le attività `DesiredStatus` sono passate a essere `STOPPED`. Questo si verifica quando un container fondamentale passa allo stato `STOPPED`.

Esempi di metadati delle attività di Amazon ECS v3

Gli esempi seguenti mostrano gli output di esempio degli endpoint dei metadati delle attività.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output.


```
{
    "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
    "Name": "nginx-curl",
    "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
    "Image": "nrdlngr/nginx-curl",
    "ImageID":
 "sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
    "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "nginx-curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
        "CPU": 512,
        "Memory": 512
    },
    "CreatedAt": "2018-02-01T20:55:10.554941919Z",
    "StartedAt": "2018-02-01T20:55:11.064236631Z",
    "Type": "NORMAL",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "10.0.2.106"
```

```
            ]
        }
    ]
}
```

Esempio di risposta dei metadati del processo

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI}/task` vengono restituiti i metadati relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

La seguente risposta JSON è relativa a un'attività con un unico container.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-
f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RESOURCES_PROVISIONED",
      "KnownStatus": "RESOURCES_PROVISIONED",
      "Limits": {
        "CPU": 0,
        "Memory": 0
      },
      "CreatedAt": "2018-02-01T20:55:08.366329616Z",
```

```
      "StartedAt": "2018-02-01T20:55:09.058354915Z",
      "Type": "CNI_PAUSE",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    },
    {
      "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
      "Name": "nginx-curl",
      "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
      "Image": "nrdlngr/nginx-curl",
      "ImageID":
 "sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "nginx-curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
        "CPU": 512,
        "Memory": 512
      },
      "CreatedAt": "2018-02-01T20:55:10.554941919Z",
      "StartedAt": "2018-02-01T20:55:11.064236631Z",
      "Type": "NORMAL",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    }
```

```
  ],
  "PullStartedAt": "2018-02-01T20:55:09.372495529Z",
  "PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
  "AvailabilityZone": "us-east-2b"
}
```

Endpoint di metadati delle attività Amazon ECS versione 2

> **Important**
>
> L'endpoint dei metadati delle attività versione 2 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta the section called “Endpoint metadati delle attività versione 4”.

A partire dalla versione 1.17.0 dell'agente del container Amazon ECS, sono disponibili diversi metadati dei processi e statistiche Docker per i processi che utilizzano la modalità di rete `awsvpc` in un endpoint HTTP fornito dall'agente del container Amazon ECS.

Tutti i container appartenenti ad attività avviate con la modalità di rete `awsvpc` ricevono un indirizzo IPv4 locale all'interno di un intervallo predefinito di indirizzi locali per il collegamento. Quando un container invia una query all'endpoint dei metadati, l'agente del container di Amazon ECS è in grado di determinare a quale processo appartiene il container in base al suo indirizzo IP univoco, per restituire quindi metadati e statistiche per tale processo.

Abilitazione dei metadati dei processi

La caratteristica dei metadati delle attività versione 2 è abilitata di default per:

- Processi che utilizzano il tipo di avvio Fargate e la versione della piattaforma v1.1.0 o versioni successive. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.
- Processi che utilizzano il tipo di avvio EC2 che utilizzano anche la modalità di rete `awsvpc` e che vengono avviati su un'infrastruttura Linux di Amazon EC2 con almeno la versione 1.17.0 dell'agente del container Amazon ECS o su un'infrastruttura Windows di Amazon EC2 con almeno la versione `1.54.0` dell'agente del 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 un file JSON con i metadati per l'attività, compreso un elenco degli ID container e i nomi di tutti i container 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 ContainerStatsla documentazione dell'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 ContainerStatsla documentazione dell'API Docker.

Risposta JSON per i metadati dei processi

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`169.254.170.2/v2/metadata`) restituisce le seguenti informazioni.

`Cluster`

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

`TaskARN`

L'Amazon Resource Name (ARN) completo 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 per le risorse specificati a livello di attività, ad esempio CPU (espressa in vCPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omesso.

`PullStartedAt`

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

`PullStoppedAt`

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

`AvailabilityZone`

La zona di disponibilità in cui si trova l'attività.

> **Note**
>
> I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

`Containers`

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omesso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omesso.

`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 omesso.

`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 omesso.

`CreatedAt`

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omesso.

`StartedAt`

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omesso.

`FinishedAt`

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omesso.

`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 omesso.

`ClockDrift`

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta Impostare l'ora per l'istanza Linux nella Guida per l'utente di Amazon EC2 per le istanze Linux.

`ReferenceTime`

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio `2021-09-07T16:57:44Z`.

`ClockErrorBound`

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

`ClockSynchronizationStatus`

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono `SYNCHRONIZED` e `NOT_SYNCHRONIZED`.

`ExecutionStoppedAt`

Il timestamp del momento in cui le attività `DesiredStatus` sono passate a essere `STOPPED`. Questo si verifica quando un container fondamentale passa allo stato `STOPPED`.

Esempio di risposta dei metadati del processo

La seguente risposta JSON è relativa a un'attività con un unico container.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-
f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
```

```
      "DesiredStatus": "RESOURCES_PROVISIONED",
      "KnownStatus": "RESOURCES_PROVISIONED",
      "Limits": {
        "CPU": 0,
        "Memory": 0
      },
      "CreatedAt": "2018-02-01T20:55:08.366329616Z",
      "StartedAt": "2018-02-01T20:55:09.058354915Z",
      "Type": "CNI_PAUSE",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    },
    {
      "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
      "Name": "nginx-curl",
      "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
      "Image": "nrdlngr/nginx-curl",
      "ImageID":
 "sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "nginx-curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
        "CPU": 512,
        "Memory": 512
      },
      "CreatedAt": "2018-02-01T20:55:10.554941919Z",
      "StartedAt": "2018-02-01T20:55:11.064236631Z",
      "Type": "NORMAL",
      "Networks": [
        {
```

```
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    }
  ],
  "PullStartedAt": "2018-02-01T20:55:09.372495529Z",
  "PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
  "AvailabilityZone": "us-east-2b"
}
```

Metadati delle attività di Amazon ECS disponibili per le attività su Fargate

Amazon ECS su Fargate fornisce un metodo per recuperare vari metadati, parametri di rete e statistiche Docker relativi ai container e i processi in essi contenuti. Questo metodo è noto come endpoint dei metadati dei processi. Le seguenti versioni degli endpoint dei metadati dei processi sono disponibili per i processi di Amazon ECS su Fargate:

- Endpoint dei metadati dei processi versione 4: disponibile per i processi che utilizzano la versione della piattaforma 1.4.0 o successiva.
- Endpoint dei metadati dei processi versione 3: disponibile per i processi che utilizzano la versione della piattaforma 1.1.0 o successiva.

Tutti i container appartenenti ad attività avviate con la modalità di rete `awsvpc` ricevono un indirizzo IPv4 locale all'interno di un intervallo predefinito di indirizzi locali per il collegamento. Quando un container invia una query all'endpoint dei metadati, l'agente del container è in grado di determinare a quale attività appartiene il container in base al suo indirizzo IP univoco, per restituire quindi metadati e statistiche per tale attività.

Argomenti

- Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate
- Endpoint di metadati delle attività Amazon ECS versione 3 per attività su Fargate

Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate

> **Important**
>
> Se utilizzi attività Amazon ECS ospitate su istanze Amazon EC2, 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 il processo, compreso un elenco degli ID container e i nomi di tutti i container associati al processo. Per ulteriori informazioni sulle risposte per questo

endpoint, consulta Risposta JSON dei metadati delle attività di Amazon ECS v4 per le attività su Fargate.

`${ECS_CONTAINER_METADATA_URI_V4}/stats`

Questo percorso le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta la documentazione dell'ContainerStatsAPI 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 ContainerStatsla documentazione dell'API Docker.

> **Note**
>
> Le attività di Amazon ECS AWS Fargate richiedono che il contenitore funzioni per circa 1 secondo prima di restituire le statistiche del contenitore.

Risposta JSON dei metadati delle attività di Amazon ECS v4 per le attività su Fargate

I seguenti metadati vengono restituiti nella risposta in formato JSON dell'endpoint dei metadati dei processi (`${ECS_CONTAINER_METADATA_URI_V4}/task`).

`Cluster`

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

`VPCID`

L'ID VPC dell'istanza di container di Amazon EC2. Questo campo viene visualizzato solo per le istanze Amazon EC2.

> **Note**
>
> Questi metadati `VPCID` sono inclusi solo quando si utilizza la versione dell'agente del container di Amazon ECS versione `1.63.1` o successiva.

`TaskARN`

L'Amazon Resource Name (ARN) completo 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 per le risorse specificati a livelli di attività, ad esempio CPU (espressa in vCPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omesso.

`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 utilizzi i provider di capacità del cluster, questo indica se il processo utilizza l'infrastruttura Fargate o EC2.

> **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) p 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 `ephermalStorage` nella definizione di attività. Lo `ephermalStorage` è specificato in GiB, non in MiB. Lo `ephermalStorage` e i parametri `EphemeralStorageMetrics` sono disponibili solo per i processi che utilizzano la piattaforma Fargate Linux versione 1.4.0 o successiva.

`Containers`

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

Quando utilizzi Fargate, l'ID è un valore esadecimale a 32 cifre seguito da un numero di 10 cifre.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omesso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omesso.

`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 omesso.

`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 omesso.

`CreatedAt`

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omesso.

`StartedAt`

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omesso.

`FinishedAt`

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omesso.

`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`

L'Amazon Resource Name (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 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. Questo parametro è disponibile solo per le attività in esecuzione sulla versione della piattaforma `1.4.0` di Linux.

`ClockDrift`

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questa funzionalità utilizza Amazon Time Sync Service per misurare la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta Impostare l'ora per l'istanza Linux nella Guida per l'utente di Amazon EC2 per le istanze Linux.

`ReferenceTime`

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio `2021-09-07T16:57:44Z`.

`ClockErrorBound`

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

`ClockSynchronizationStatus`

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono `SYNCHRONIZED` e `NOT_SYNCHRONIZED`.

`ExecutionStoppedAt`

Il timestamp del momento in cui le attività `DesiredStatus` sono passate a essere `STOPPED`. Questo si verifica quando un container fondamentale passa allo stato `STOPPED`.

Esempi di metadati delle attività di Amazon ECS v4 per attività su Fargate

Gli esempi seguenti mostrano gli output di esempio degli endpoint dei metadati dei processi per i processi di Amazon ECS in esecuzione su AWS Fargate.

Dal container è possibile utilizzare il comando curl seguito dall'endpoint dei metadati dell'attività per eseguire una query sull'endpoint, ad esempio `curl ${ECS_CONTAINER_METADATA_URI_V4}/task`.

Esempio di risposta dei metadati del container

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}` vengono restituiti solo i metadati relativi al container stesso. Di seguito è riportato un esempio di output.

```
{
    "DockerId": "cd189a933e5849daa93386466019ab50-2495160603",
    "Name": "curl",
    "DockerName": "curl",
    "Image": "111122223333.dkr.ecr.us-west-2.amazonaws.com/curltest:latest",
    "ImageID":
 "sha256:25f3695bedfb454a50f12d127839a68ad3caf91e451c1da073db34c542c4d2cb",
    "Labels": {
        "com.amazonaws.ecs.cluster": "arn:aws:ecs:us-west-2:111122223333:cluster/
default",
        "com.amazonaws.ecs.container-name": "curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-west-2:111122223333:task/default/
cd189a933e5849daa93386466019ab50",
        "com.amazonaws.ecs.task-definition-family": "curltest",
        "com.amazonaws.ecs.task-definition-version": "2"
    },
    "DesiredStatus": "RUNNING",
    "KnownStatus": "RUNNING",
    "Limits": {
        "CPU": 10,
        "Memory": 128
    },
    "CreatedAt": "2020-10-08T20:09:11.44527186Z",
    "StartedAt": "2020-10-08T20:09:11.44527186Z",
```

```
    "Type": "NORMAL",
    "Networks": [
        {
            "NetworkMode": "awsvpc",
            "IPv4Addresses": [
                "192.0.2.3"
            ],
            "AttachmentIndex": 0,
            "MACAddress": "0a:de:f6:10:51:e5",
            "IPv4SubnetCIDRBlock": "192.0.2.0/24",
            "DomainNameServers": [
                "192.0.2.2"
            ],
            "DomainNameSearchList": [
                "us-west-2.compute.internal"
            ],
            "PrivateDNSName": "ip-10-0-0-222.us-west-2.compute.internal",
            "SubnetGatewayIpv4Address": "192.0.2.0/24"
        }
    ],
    "ContainerARN": "arn:aws:ecs:us-west-2:111122223333:container/05966557-
f16c-49cb-9352-24b3a0dcd0e1",
    "LogOptions": {
        "awslogs-create-group": "true",
        "awslogs-group": "/ecs/containerlogs",
        "awslogs-region": "us-west-2",
        "awslogs-stream": "ecs/curl/cd189a933e5849daa93386466019ab50"
    },
    "LogDriver": "awslogs",
    "Snapshotter": "overlayfs"
}
```

Esempi di metadati delle attività di Amazon ECS v4 per attività su Fargate

Quando si esegue una query sull'endpoint `${ECS_CONTAINER_METADATA_URI_V4}/task` vengono restituiti i metadati relativi all'attività di cui fa parte il container. Di seguito è riportato un esempio di output.

```
{
  "Cluster": "arn:aws:ecs:us-east-1:123456789012:cluster/clusterName",
  "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
        ],
        "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
        ],
        "usage_in_kernelmode": 80000000,
        "usage_in_usermode": 810000000
      },
      "system_cpu_usage": 9373330000000,
      "online_cpus": 2,
      "throttling_data": {
        "periods": 0,
        "throttled_periods": 0,
        "throttled_time": 0
      }
    },
    "memory_stats": {
      "usage": 6504448,
      "max_usage": 8458240,
      "stats": {
        "active_anon": 1675264,
        "active_file": 557056,
        "cache": 651264,
        "dirty": 0,
        "hierarchical_memory_limit": 536870912,
        "hierarchical_memsw_limit": 9223372036854772000,
        "inactive_anon": 0,
        "inactive_file": 3088384,
        "mapped_file": 430080,
        "pgfault": 11034,
```

```
        "pgmajfault": 5,
        "pgpgin": 8436,
        "pgpgout": 7137,
        "rss": 4669440,
        "rss_huge": 0,
        "total_active_anon": 1675264,
        "total_active_file": 557056,
        "total_cache": 651264,
        "total_dirty": 0,
        "total_inactive_anon": 0,
        "total_inactive_file": 3088384,
        "total_mapped_file": 430080,
        "total_pgfault": 11034,
        "total_pgmajfault": 5,
        "total_pgpgin": 8436,
        "total_pgpgout": 7137,
        "total_rss": 4669440,
        "total_rss_huge": 0,
        "total_unevictable": 0,
        "total_writeback": 0,
        "unevictable": 0,
        "writeback": 0
      },
      "limit": 9223372036854772000
    },
    "name": "curltest",
    "id": "3d1f891cded94dc795608466cce8ddcf-464223573",
    "networks": {
      "eth1": {
        "rx_bytes": 2398415937,
        "rx_packets": 1898631,
        "rx_errors": 0,
        "rx_dropped": 0,
        "tx_bytes": 1259037719,
        "tx_packets": 428002,
        "tx_errors": 0,
        "tx_dropped": 0
      }
    },
    "network_rate_stats": {
      "rx_bytes_per_sec": 43.298687872232854,
      "tx_bytes_per_sec": 215.39347269466413
    }
  }
```

```
}
```

Endpoint di metadati delle attività Amazon ECS versione 3 per attività su Fargate

> **Important**
>
> L'endpoint dei metadati delle attività versione 3 non viene più gestito attivamente. Ti consigliamo di aggiornare l'endpoint dei metadati delle attività versione 4 per ottenere le informazioni più recenti sull'endpoint dei metadati. Per ulteriori informazioni, consulta the section called “Endpoint metadati delle attività versione 4 per le attività su Fargate”.

A partire dalla piattaforma Fargate versione `1.1.0`, una variabile di ambiente denominata `ECS_CONTAINER_METADATA_URI` viene inserita in ogni container di un processo. Quando esegui una query sull'endpoint dei metadati delle attività versione 3, per le attività vengono resi disponibili diversi metadati delle attività e statistiche Docker.

L'endpoint dei metadati dei processi è abilitato di default per tutti i processi di Amazon ECS ospitati su Fargate che utilizzano la versione della piattaforma `1.1.0` o versione successiva. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux 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 un file JSON con i metadati per l'attività, compreso un elenco degli ID container e i nomi di tutti i container associati all'attività. Per ulteriori informazioni sulle risposte per questo endpoint, consulta Risposta JSON dei metadati delle attività di Amazon ECS v3 per le attività su Fargate.

`${ECS_CONTAINER_METADATA_URI}/stats`

Questo percorso restituisce un file JSON con le statistiche Docker per il container Docker specificato. Per ulteriori informazioni su ciascuna delle statistiche restituite, consulta la documentazione dell'ContainerStatsAPI 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 ContainerStatsla documentazione dell'API Docker.

Risposta JSON dei metadati delle attività di Amazon ECS v3 per le attività su Fargate

La risposta in formato JSON dell'endpoint dei metadati per l'attività (`${ECS_CONTAINER_METADATA_URI}/task`) restituisce le seguenti informazioni.

`Cluster`

L'Amazon Resource Name (ARN) o nome breve del cluster Amazon ECS a cui appartiene il processo.

`TaskARN`

L'Amazon Resource Name (ARN) completo 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 per le risorse specificati a livello di attività, ad esempio CPU (espressa in vCPU) e memoria. Se non ci sono limiti di risorse definiti, questo parametro viene omesso.

`PullStartedAt`

Il timestamp dell'inizio della prima estrazione per l'immagine del container.

`PullStoppedAt`

Il timestamp del termine dell'ultima estrazione per l'immagine del container.

`AvailabilityZone`

La zona di disponibilità in cui si trova l'attività.

> **Note**
>
> I metadati della zona di disponibilità sono disponibili solo per i processi Fargate che utilizzano la versione 1.4 o successiva della piattaforma (Linux) o 1.0.0 o successiva (Windows).

`Containers`

Un elenco di metadati dei container per ogni container associato all'attività.

`DockerId`

L'ID Docker per il container.

`Name`

Il nome del container come specificato nella definizione di attività.

`DockerName`

Il nome del container fornito a Docker. L'agente del container di Amazon ECS genera un nome univoco per il container al fine di evitare conflitti quando, su una singola istanza, vengono eseguite più copie della stessa definizione di attività.

`Image`

L'immagine per il container.

`ImageID`

Il digest SHA-256 per l'immagine.

`Ports`

Eventuali porte esposte per il container. Se non ci sono porte esposte, questo parametro viene omesso.

`Labels`

Eventuali etichette applicate al container. Se non ci sono etichette applicate, questo parametro viene omesso.

`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 omesso.

`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 omesso.

`CreatedAt`

Il timestamp della creazione del container. Se il container non è ancora stato creato, questo parametro viene omesso.

`StartedAt`

Il timestamp dell'avvio del container. Se il container non è ancora stato avviato, questo parametro viene omesso.

`FinishedAt`

Il timestamp dell'arresto del container. Se il container non è ancora stato arrestato, questo parametro viene omesso.

`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 omesso.

`ClockDrift`

Le informazioni sulla differenza tra l'ora di riferimento e l'ora del sistema. Questo vale per il sistema operativo Linux. Questa funzionalità utilizza Amazon Time Sync Service per misurare

la precisione dell'orologio e fornire il limite di errore dell'orologio per i contenitori. Per ulteriori informazioni, consulta Impostare l'ora per l'istanza Linux nella Guida per l'utente di Amazon EC2 per le istanze Linux.

`ReferenceTime`

La base della precisione dell'orologio. Amazon ECS utilizza lo standard globale Coordinated Universal Time (UTC) tramite NTP, ad esempio `2021-09-07T16:57:44Z`.

`ClockErrorBound`

La misura dell'errore di clock, definita come offset rispetto a UTC. Questo errore è la differenza in millisecondi tra l'ora di riferimento e l'ora del sistema.

`ClockSynchronizationStatus`

Indica se il tentativo di sincronizzazione più recente tra l'ora del sistema e l'ora di riferimento ha avuto esito positivo.

I valori validi sono `SYNCHRONIZED` e `NOT_SYNCHRONIZED`.

`ExecutionStoppedAt`

Il timestamp del momento in cui le attività `DesiredStatus` sono passate a essere `STOPPED`. Questo si verifica quando un container fondamentale passa allo stato `STOPPED`.

Esempi di metadati delle attività di Amazon ECS v3 per attività su Fargate

La seguente risposta JSON è relativa a un'attività con un unico container.

```
{
  "Cluster": "default",
  "TaskARN": "arn:aws:ecs:us-east-2:012345678910:task/9781c248-0edd-4cdb-9a93-
f63cb662a5d3",
  "Family": "nginx",
  "Revision": "5",
  "DesiredStatus": "RUNNING",
  "KnownStatus": "RUNNING",
  "Containers": [
    {
      "DockerId": "731a0d6a3b4210e2448339bc7015aaa79bfe4fa256384f4102db86ef94cbbc4c",
      "Name": "~internal~ecs~pause",
      "DockerName": "ecs-nginx-5-internalecspause-acc699c0cbf2d6d11700",
      "Image": "amazon/amazon-ecs-pause:0.1.0",
      "ImageID": "",
```

```
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "~internal~ecs~pause",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RESOURCES_PROVISIONED",
      "KnownStatus": "RESOURCES_PROVISIONED",
      "Limits": {
        "CPU": 0,
        "Memory": 0
      },
      "CreatedAt": "2018-02-01T20:55:08.366329616Z",
      "StartedAt": "2018-02-01T20:55:09.058354915Z",
      "Type": "CNI_PAUSE",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    },
    {
      "DockerId": "43481a6ce4842eec8fe72fc28500c6b52edcc0917f105b83379f88cac1ff3946",
      "Name": "nginx-curl",
      "DockerName": "ecs-nginx-5-nginx-curl-ccccb9f49db0dfe0d901",
      "Image": "nrdlngr/nginx-curl",
      "ImageID":
 "sha256:2e00ae64383cfc865ba0a2ba37f61b50a120d2d9378559dcd458dc0de47bc165",
      "Labels": {
        "com.amazonaws.ecs.cluster": "default",
        "com.amazonaws.ecs.container-name": "nginx-curl",
        "com.amazonaws.ecs.task-arn": "arn:aws:ecs:us-
east-2:012345678910:task/9781c248-0edd-4cdb-9a93-f63cb662a5d3",
        "com.amazonaws.ecs.task-definition-family": "nginx",
        "com.amazonaws.ecs.task-definition-version": "5"
      },
      "DesiredStatus": "RUNNING",
      "KnownStatus": "RUNNING",
      "Limits": {
```

```
        "CPU": 512,
        "Memory": 512
      },
      "CreatedAt": "2018-02-01T20:55:10.554941919Z",
      "StartedAt": "2018-02-01T20:55:11.064236631Z",
      "Type": "NORMAL",
      "Networks": [
        {
          "NetworkMode": "awsvpc",
          "IPv4Addresses": [
            "10.0.2.106"
          ]
        }
      ]
    }
  ],
  "PullStartedAt": "2018-02-01T20:55:09.372495529Z",
  "PullStoppedAt": "2018-02-01T20:55:10.552018345Z",
  "AvailabilityZone": "us-east-2b"
}
```

Introspezione dei container Amazon ECS

L'agente del container di Amazon ECS fornisce un'operazione API per raccogliere i dettagli sull'istanza di container su cui è in esecuzione l'agente e sui processi associati in esecuzione su tale istanza. Puoi utilizzare il comando curl dall'interno dell'istanza di container per interrogare l'agente del container di Amazon ECS (porta 51678) e restituire metadati dell'istanza di container o informazioni sulle attività.

> **Important**
>
> L'istanza di container deve disporre di un ruolo IAM che consenta l'accesso ad Amazon ECS per recuperare i metadati. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

Per visualizzare i metadati dell'istanza di container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente. I metadati includono l'ID dell'istanza di container, il cluster Amazon ECS in cui è registrata l'istanza di container e le informazioni sulla versione dell'agente del container di Amazon ECS.

```
curl -s http://localhost:51678/v1/metadata | python3 -mjson.tool
```

Output:

```
{
    "Cluster": "cluster_name",
    "ContainerInstanceArn": "arn:aws:ecs:region:aws_account_id:container-
instance/cluster_name/container_instance_id",
    "Version": "Amazon ECS Agent - v1.30.0 (02ff320c)"
}
```

Per visualizzare le informazioni su tutte le attività in esecuzione in un'istanza di container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente:

```
curl http://localhost:51678/v1/tasks
```

Output:

```
{
    "Tasks": [
        {
            "Arn": "arn:aws:ecs:us-west-2:012345678910:task/default/example5-58ff-46c9-
ae05-543f8example",
            "DesiredStatus": "RUNNING",
            "KnownStatus": "RUNNING",
            "Family": "hello_world",
            "Version": "8",
            "Containers": [
                {
          "DockerId":
 "9581a69a761a557fbfce1d0f6745e4af5b9dbfb86b6b2c5c4df156f1a5932ff1",
          "DockerName": "ecs-hello_world-8-mysql-fcae8ac8f9f1d89d8301",
          "Name": "mysql",
          "CreatedAt": "2023-10-08T20:09:11.44527186Z",
          "StartedAt": "2023-10-08T20:09:11.44527186Z",
          "ImageID":
 "sha256:2ae34abc2ed0a22e280d17e13f9c01aaf725688b09b7a1525d1a2750e2c0d1de"
        },
        {
          "DockerId":
 "bf25c5c5b2d4dba68846c7236e75b6915e1e778d31611e3c6a06831e39814a15",
```

```
            "DockerName": "ecs-hello_world-8-wordpress-e8bfddf9b488dff36c00",
            "Name": "wordpress"
        }
    ]
}
    ]
}
```

Puoi visualizzare le informazioni per una determinata attività in esecuzione in un'istanza di container. Per specificare una determinata attività o uno specifico container, aggiungi uno dei seguenti elementi alla richiesta:

- L'ARN dell'attività (`?taskarn=`*`task_arn`*)
- L'ID Docker per un container (`?dockerid=`*`docker_id`*)

Per ottenere informazioni sulle attività con un ID Docker del container, accedi alla tua istanza di container tramite SSH ed esegui il comando seguente.

> **Note**
>
> Gli agenti del container di Amazon ECS precedenti alla versione 1.14.2 richiedono ID completi del container Docker per l'introspezione API, non la versione breve che viene mostrata con il comando docker ps. Puoi ottenere l'ID Docker completo per un container eseguendo il comando docker ps --no-trunc nell'istanza di container.

```
curl http://localhost:51678/v1/tasks?dockerid=79c796ed2a7f
```

Output:

```
{
    "Arn": "arn:aws:ecs:us-west-2:012345678910:task/default/e01d58a8-151b-40e8-
bc01-22647b9ecfec",
    "Containers": [
        {
            "DockerId":
 "79c796ed2a7f864f485c76f83f3165488097279d296a7c05bd5201a1c69b2920",
            "DockerName": "ecs-nginx-efs-2-nginx-9ac0808dd0afa495f001",
            "Name": "nginx",
```

```
            "CreatedAt": "2023-10-08T20:09:11.44527186Z",
            "StartedAt": "2023-10-08T20:09:11.44527186Z",
            "ImageID":
 "sha256:2ae34abc2ed0a22e280d17e13f9c01aaf725688b09b7a1525d1a2750e2c0d1de"
        }
    ],
    "DesiredStatus": "RUNNING",
    "Family": "nginx-efs",
    "KnownStatus": "RUNNING",
    "Version": "2"
}
```

Identifica i comportamenti non autorizzati utilizzando Runtime Monitoring

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

Runtime Monitoring in GuardDuty protegge i carichi di lavoro in esecuzione su istanze di container Fargate ed EC2 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 Runtime Monitoring nella Guida per l'utente. GuardDuty GuardDuty

L'amministratore della sicurezza abilita il Runtime Monitoring per uno o più account in AWS Organizations for GuardDuty. Inoltre, selezionano se distribuire GuardDuty automaticamente il GuardDuty Security Agent quando si utilizza Fargate. Tutti i tuoi cluster sono protetti automaticamente e GuardDuty gestiscono il security agent per tuo conto.

Puoi anche configurare manualmente il GuardDuty security agent nei seguenti casi:

- Utilizzi istanze di container EC2
- 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 contenitore EC2.

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 container EC2, il GuardDuty security agent viene eseguito come processo sull'istanza.

Il GuardDuty security agent raccoglie i dati dalle seguenti risorse e quindi li invia all' GuardDuty elaborazione. È possibile visualizzare i risultati nella GuardDuty console. Puoi anche inviarli ad altri, ad Servizi AWS esempio AWS Security Hub, o a un fornitore di sicurezza di terze parti per l'aggregazione e la correzione. Per informazioni su come visualizzare e gestire i risultati, consulta Managing Amazon GuardDuty findings nella Amazon GuardDuty User Guide.

- Risposte dalle seguenti chiamate API Amazon ECS:
  - DescribeClusters

    I parametri di risposta includono il tag Runtime Monitoring (quando il tag è impostato) quando si utilizza l'`--include TAGS`opzione.
  - DescribeTasks

    Per il tipo di lancio Fargate, i parametri di risposta includono il contenitore GuardDuty sidecar.
  - ListAccountSettings

    I parametri di risposta includono l'impostazione dell'account Runtime Monitoring, impostata dall'amministratore della sicurezza.
- I dati di introspezione dell'agente contenitore. Per ulteriori informazioni, consulta Introspezione dei container Amazon ECS.
- L'endpoint dei metadati dell'attività per il tipo di avvio:
  - Endpoint di metadati delle attività Amazon ECS versione 4
  - Endpoint di metadati delle attività Amazon ECS versione 4 per attività su Fargate

Considerazioni

Considerate quanto segue quando utilizzate Runtime Monitoring:

- Il Runtime Monitoring ha un costo associato. Per ulteriori informazioni, consulta la pagina GuardDuty dei prezzi di Amazon.
- Il monitoraggio del runtime non è supportato su Amazon ECS Anywhere.
- Il monitoraggio del runtime non è supportato per il sistema operativo Windows.
- Quando usi Amazon ECS Exec su Fargate, devi specificare il nome del contenitore perché il GuardDuty security agent funziona come contenitore secondario.
- Non puoi usare Amazon ECS Exec nel contenitore sidecar del GuardDuty Security Agent.
- L'utente IAM che controlla il Runtime Monitoring a livello di cluster deve disporre delle autorizzazioni IAM appropriate per l'etichettatura. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire l'agente di GuardDuty sicurezza, che è archiviato in un repository privato Amazon ECR, per tuo conto.

Utilizzo delle risorse

Il tag che aggiungi al cluster viene conteggiato ai fini della quota di tag del cluster.

Il contenitore GuardDuty Agent Sidecar non viene conteggiato ai fini della quota di definizione dei contenitori per attività.

Come con la maggior parte dei software di sicurezza, c'è un leggero sovraccarico per. GuardDuty Per informazioni sui limiti di memoria di Fargate, consultate Limiti di CPU e memoria nella Guida per l'GuardDuty utente. Per informazioni sui limiti di memoria di Amazon EC2, consulta Limite di CPU e memoria per l' GuardDuty agente.

Monitoraggio del runtime per carichi di lavoro Amazon ECS Fargate

Se utilizzi istanze di container EC2, devi configurare manualmente il Runtime Monitoring. Per ulteriori informazioni, consulta Monitoraggio del runtime per carichi di lavoro EC2 su Amazon ECS.

Puoi GuardDuty gestire il security agent sulle istanze del contenitore. Questa opzione è disponibile solo per Fargate. Questa opzione (gestione degli GuardDuty agenti) è disponibile in GuardDuty

Quando si utilizza la gestione degli GuardDuty agenti, GuardDuty esegue le seguenti operazioni:

- Crea endpoint VPC per GuardDuty ogni VPC che ospita un cluster.
- Recupera e installa il GuardDuty security agent più recente come contenitore secondario per tutte le nuove attività autonome di Fargate e le nuove implementazioni di servizi.

  La distribuzione di un nuovo servizio avviene la prima volta che si avvia un servizio o quando si aggiorna un servizio esistente con l'opzione Force New Deployment.

Attivazione del monitoraggio del runtime per Amazon ECS

È possibile GuardDuty configurare la gestione automatica del security agent per tutti i cluster Fargate.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
  - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.
  - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Connessione al repository Amazon ECR:

  Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:

  - Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

    Per ulteriori informazioni, consulta Usare una sottorete privata e un gateway NAT.

- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per. GuardDuty Per ulteriori informazioni, consulta la pagina relativa alle autorizzazioni dei ruoli collegati ai servizi GuardDuty nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo `755` su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze di container EC2:

- È necessario utilizzare una versione `20230929` o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione `5.10` del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring.
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, consulta Configurazione delle istanze di Systems Manager for EC2 nella Guida per l'AWS Systems Manager Session Manager utente.

È possibile abilitare il monitoraggio del runtime in. GuardDuty Per informazioni su come abilitare la funzionalità, consulta Enabling Runtime Monitoring nella Amazon GuardDuty User Guide.

Aggiungere il monitoraggio del runtime alle attività esistenti di Amazon ECS Fargate

Quando attivi il Runtime Monitoring, tutte le nuove attività autonome e le nuove implementazioni di servizi nel cluster vengono protette automaticamente. Al fine di preservare il vincolo di immutabilità, le attività esistenti non vengono influenzate.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
  - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

  - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Connessione al repository Amazon ECR:

  Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:

  - Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

    Per ulteriori informazioni, consulta Usare una sottorete privata e un gateway NAT.
- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per. GuardDuty Per ulteriori informazioni, consulta la pagina relativa alle autorizzazioni dei ruoli collegati ai servizi GuardDuty nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo `755` su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze di container EC2:

- È necessario utilizzare una versione `20230929` o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione `5.10` del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring.
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, consulta Configurazione delle istanze di Systems Manager for EC2 nella Guida per l'AWS Systems Manager Session Manager utente.

Per proteggere immediatamente un'attività, è necessario eseguire una delle seguenti azioni:

- Per le attività autonome, interrompi le attività e quindi avviale. Per ulteriori informazioni, consulta Interruzione di un'attività Amazon ECS e Esecuzione di un'applicazione come attività Amazon ECS.
- Per le attività che fanno parte di un servizio, aggiorna il servizio con l'opzione «forza la nuova distribuzione». Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console.

Rimuovere il monitoraggio del runtime da un cluster Amazon ECS

Potresti voler escludere determinati cluster dalla protezione, ad esempio i cluster che usi per i test. Ciò comporta GuardDuty l'esecuzione delle seguenti operazioni sulle risorse del cluster:

- Non utilizzate più il GuardDuty Security Agent per nuove attività autonome di Fargate o per nuove implementazioni di servizi.

  Al fine di preservare il vincolo di immutabilità, le attività e le distribuzioni esistenti con Runtime Monitoring abilitato non ne risentono.
- Interrompi la fatturazione e non accetta più eventi di runtime per le attività.

Esegui i passaggi seguenti per rimuovere Runtime Monitoring da un cluster.

1. Usa la console Amazon ECS o imposta AWS CLI la chiave del `GuardDutyManaged` tag sul cluster su. `false` Per ulteriori informazioni, consulta Aggiornamento di un cluster o Utilizzo dei tag utilizzando la CLI o l'API. Usa i seguenti valori per il tag.

   > **Note**
   >
   > La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

   Chiave =`GuardDutyManaged`, Valore = `false`
2. Elimina l'endpoint GuardDuty VPC per il cluster. Per ulteriori informazioni su come eliminare gli endpoint VPC, consulta Eliminare un endpoint di interfaccia nella Guida per l'utente.AWS PrivateLink

Rimuovere Runtime Monitoring for Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disattiva la funzionalità in GuardDuty. Per informazioni su come disabilitare la funzionalità, consulta Enabling Runtime Monitoring nella Amazon GuardDuty User Guide.

GuardDuty esegue le seguenti operazioni:

- Elimina gli endpoint VPC GuardDuty per ogni VPC che ospita un cluster.
- Non distribuisce più il GuardDuty Security Agent a nuove attività autonome di Fargate o a nuove implementazioni di servizi.

  Al fine di preservare il vincolo di immutabilità, le attività e le implementazioni esistenti non vengono influenzate finché non vengono interrotte, replicate o ridimensionate.
- Interrompe la fatturazione e non accetta più eventi di esecuzione per le attività.

Monitoraggio del runtime per carichi di lavoro EC2 su Amazon ECS

Utilizza questa opzione quando utilizzi istanze EC2 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 Runtime Monitoring aggiungendo un tag predefinito.

Per le istanze di container EC2, devi scaricare, installare e gestire 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 istanze EC2 o quando hai bisogno di un controllo granulare del Runtime Monitoring a livello di cluster su Fargate.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
  - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

  - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Connessione al repository Amazon ECR:

  Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:

  - Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

    Per ulteriori informazioni, consulta Usare una sottorete privata e un gateway NAT.
- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per. GuardDuty Per ulteriori informazioni, consulta la pagina relativa alle autorizzazioni dei ruoli collegati ai servizi GuardDuty nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo `755` su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze di container EC2:

- È necessario utilizzare una versione `20230929` o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione `5.10` del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring.
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, consulta Configurazione delle istanze di Systems Manager for EC2 nella Guida per l'AWS Systems Manager Session Manager utente.

Si attiva il Runtime Monitoring in. GuardDuty Per informazioni su come abilitare la funzionalità, consulta Enabling Runtime Monitoring nella Amazon GuardDuty User Guide.

Aggiungere il monitoraggio del runtime di un cluster Amazon ECS

Configura il monitoraggio del runtime per il cluster, quindi installa il GuardDuty security agent sulle istanze del contenitore EC2.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
  - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.
  - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Connessione al repository Amazon ECR:

  Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:
  - Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

    Per ulteriori informazioni, consulta Usare una sottorete privata e un gateway NAT.
- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per. GuardDuty Per ulteriori informazioni, consulta la pagina relativa alle autorizzazioni dei ruoli collegati ai servizi GuardDuty nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo `755` su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze di container EC2:

- È necessario utilizzare una versione `20230929` o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione `5.10` del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring.
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, consulta Configurazione delle istanze di Systems Manager for EC2 nella Guida per l'AWS Systems Manager Session Manager utente.

Esegui le seguenti operazioni per aggiungere Runtime Monitoring a un cluster.

1. Crea un endpoint VPC GuardDuty per ogni VPC del cluster. Per ulteriori informazioni, consulta Creazione manuale di un endpoint Amazon VPC nella Guida per l'GuardDuty utente.
2. Configura le istanze del contenitore EC2.

   a. Aggiorna l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze di container EC2 nel cluster. Per ulteriori informazioni, consulta Aggiornamento dell'agente del container Amazon ECS.

   b. Installa l'agente GuardDuty di sicurezza sulle istanze del contenitore EC2 nel cluster. Per ulteriori informazioni, consulta Gestire manualmente il security agent su un'istanza Amazon EC2 nella Guida per l'GuardDuty utente.

      Tutte le attività e le implementazioni nuove ed esistenti sono immediatamente protette perché il GuardDuty security agent viene eseguito come processo sull'istanza del contenitore EC2.
3. Usa la console Amazon ECS o imposta AWS CLI la chiave del `GuardDutyManaged` tag sul cluster su. `true` Per ulteriori informazioni, consulta Aggiornamento di un cluster o Utilizzo dei tag utilizzando la CLI o l'API. Usa i seguenti valori per il tag.

> **Note**
>
> La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

Chiave =`GuardDutyManaged`, Valore = `true`

Aggiungere il monitoraggio del runtime alle attività esistenti di Amazon ECS

Quando attivi il Runtime Monitoring, tutte le nuove attività autonome e le nuove implementazioni di servizi nel cluster vengono protette automaticamente. Al fine di preservare il vincolo di immutabilità, le attività esistenti non vengono influenzate.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring:

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva per Linux.
- Ruoli e autorizzazioni IAM per Amazon ECS:
  - Le attività Fargate devono utilizzare un ruolo di esecuzione delle attività. Questo ruolo concede alle attività l'autorizzazione a recuperare, aggiornare e gestire il GuardDuty security agent per conto dell'utente. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.
  - Puoi controllare il Runtime Monitoring per un cluster con un tag predefinito. Se le tue policy di accesso limitano l'accesso in base ai tag, devi concedere autorizzazioni esplicite agli utenti IAM per etichettare i cluster. Per ulteriori informazioni, consulta il tutorial IAM: Definisci le autorizzazioni per accedere alle AWS risorse in base ai tag nella IAM User Guide.
- Connessione al repository Amazon ECR:

  Il GuardDuty security agent è archiviato in un repository Amazon ECR. Ogni attività autonoma e di servizio deve avere accesso al repository. Puoi utilizzare una delle seguenti opzioni:
  - Per le attività nelle sottoreti pubbliche, puoi utilizzare un indirizzo IP pubblico per l'attività o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività. Per ulteriori informazioni, consulta Endpoint VPC dell'interfaccia Amazon ECR (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.
  - Per le attività in sottoreti private, puoi utilizzare un gateway NAT (Network Address Translation) o creare un endpoint VPC per Amazon ECR nella sottorete in cui viene eseguita l'attività.

Per ulteriori informazioni, consulta Usare una sottorete privata e un gateway NAT.

- Devi avere il `AWSServiceRoleForAmazonGuardDuty` ruolo per. GuardDuty Per ulteriori informazioni, consulta la pagina relativa alle autorizzazioni dei ruoli collegati ai servizi GuardDuty nella Amazon GuardDuty User Guide.
- Tutti i file che desideri proteggere con Runtime Monitoring devono essere accessibili dall'utente root. Se hai modificato manualmente le autorizzazioni di un file, devi impostarlo `755` su.

Di seguito sono riportati i prerequisiti per l'utilizzo di Runtime Monitoring sulle istanze di container EC2:

- È necessario utilizzare una versione `20230929` o successiva di Amazon ECS-AMI.
- È necessario eseguire l'agente Amazon ECS alla versione `1.77` o successiva sulle istanze del contenitore.
- È necessario utilizzare la versione `5.10` del kernel o successiva.
- Per informazioni sui sistemi operativi e sulle architetture Linux supportati, consulta Quali modelli operativi e carichi di lavoro supporta GuardDuty il Runtime Monitoring.
- È possibile utilizzare Systems Manager per gestire le istanze dei container. Per ulteriori informazioni, consulta Configurazione delle istanze di Systems Manager for EC2 nella Guida per l'AWS Systems Manager Session Manager utente.

Per proteggere immediatamente un'attività, è necessario eseguire una delle seguenti azioni:

- Per le attività autonome, interrompi le attività e quindi avviale. Per ulteriori informazioni, consulta Interruzione di un'attività Amazon ECS e Esecuzione di un'applicazione come attività Amazon ECS.
- Per le attività che fanno parte di un servizio, aggiorna il servizio con l'opzione «forza la nuova distribuzione». Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console.

Rimuovere il monitoraggio del runtime da un cluster Amazon ECS

Puoi rimuovere Runtime Monitoring da un cluster. Ciò GuardDuty causa l'interruzione del monitoraggio di tutte le risorse nel cluster.

Per rimuovere Runtime Monitoring da un cluster.

1. Usa la console Amazon ECS o imposta AWS CLI la chiave del `GuardDutyManaged` tag sul cluster su. `false` Per ulteriori informazioni, consulta Aggiornamento di un cluster o Utilizzo dei tag utilizzando la CLI o l'API.

   > **Note**
   >
   > La chiave e il valore fanno distinzione tra maiuscole e minuscole e devono corrispondere esattamente alle stringhe.

   Chiave =`GuardDutyManaged`, Valore = `false`
2. Disinstalla il GuardDuty security agent sulle istanze del contenitore EC2 nel cluster.

   Per ulteriori informazioni, consulta Disinstallazione manuale del security agent nella Guida per l'GuardDuty utente.
3. Elimina l'endpoint GuardDuty VPC per ogni VPC del cluster. Per ulteriori informazioni su come eliminare gli endpoint VPC, consulta Eliminare un endpoint di interfaccia nella Guida per l'utente.AWS PrivateLink

Aggiornamento del GuardDuty security agent sulle istanze di container Amazon ECS

Per informazioni su come aggiornare il GuardDuty security agent sulle istanze di container EC2, consulta Updating GuardDuty security agent nella Amazon GuardDuty User Guide.

Rimuovere Runtime Monitoring for Amazon ECS da un account

Quando non desideri più utilizzare Runtime Monitoring, disattiva la funzionalità in GuardDuty. Per informazioni su come disabilitare la funzionalità, consulta Enabling Runtime Monitoring nella Amazon GuardDuty User Guide.

Rimuovi Runtime Monitoring da tutti i cluster. Per ulteriori informazioni, consulta Rimuovere il monitoraggio del runtime da un cluster Amazon ECS.

Domande frequenti sulla risoluzione dei problemi relativi al monitoraggio del runtime

Potrebbe essere necessario risolvere i problemi o verificare che il Runtime Monitoring sia abilitato e in esecuzione sulle attività e sui contenitori.

Argomenti

Come posso sapere se Runtime Monitoring è attivo sul mio account?

Nella console Amazon ECS, le informazioni si trovano nella pagina Impostazioni account.

Puoi anche correre `list-account-settings` con l'`effective-settings`opzione.

```
aws ecs list-account-settings --effective-settings
```

Output

L'impostazione con nome impostato `guardDutyActivate` e valore impostato su on indica che l'account è configurato. È necessario verificare con GuardDuty l'amministratore se la gestione è automatica o manuale.

```
{
```

```
    "setting": {
        "name": "guardDutyActivate",
        "value": "enabled",
        "principalArn": "arn:aws:iam::123456789012:root",
        "type": "aws-managed"
    }
}
```

Come posso sapere se il Runtime Monitoring è attivo su un cluster?

Nella console Amazon ECS, le informazioni si trovano nella scheda Tag della pagina dei dettagli del cluster.

Puoi anche correre `describe-clusters` con l'`TAGS`opzione.

L'esempio seguente mostra l'output per il cluster predefinito

```
aws ecs describe-clusters --cluster default --include TAGS
```

Output

Il tag con Key impostato su `GuardDutyManaged` e Valore impostato su `true` indica che il cluster è configurato per il Runtime Monitoring.

```
{
    "clusters": [
        {
            "clusterArn": "arn:aws:ecs:us-east-1:1234567890:cluster/default",
            "clusterName": "default",
            "status": "ACTIVE",
            "registeredContainerInstancesCount": 0,
            "runningTasksCount": 1,
            "pendingTasksCount": 0,
            "activeServicesCount": 0,
            "statistics": [],
            "tags": [
                {
                    "key": "GuardDutyManaged",
                    "value": "true"
                }
            ],
            "settings": [],
```

```
            "capacityProviders": [],
            "defaultCapacityProviderStrategy": []
        }
    ],
    "failures": []
}
```

Come posso sapere se il GuardDuty security agent sta eseguendo un'attività di Fargate?

L'agente GuardDuty di sicurezza funge da contenitore secondario per le attività di Fargate.

Nella console Amazon ECS, il sidecar viene visualizzato in Contenitori nella pagina dei dettagli dell'attività.

Puoi eseguire `describe-tasks` e cercare il contenitore con un nome impostato su `aws-gd-agent` e lastStatus impostato su. `RUNNING`

L'esempio seguente mostra l'output per il cluster predefinito per l'attività`aws:ecs:us-east-1:123456789012:task/0b69d5c0-d655-4695-98cd-5d2d5EXAMPLE`.

```
aws ecs describe-tasks --cluster default --tasks aws:ecs:us-
east-1:123456789012:task/0b69d5c0-d655-4695-98cd-5d2d5EXAMPLE
```

Output

Il contenitore denominato `gd-agent` si trova nello `RUNNING` stato.

```
"containers": [
      {
        "containerArn": "arn:aws:ecs:us-east-1:123456789012:container/4df26bb4-
f057-467b-a079-96167EXAMPLE",
        "taskArn": "arn:aws:ecs:us-east-1:123456789012:task/0b69d5c0-
d655-4695-98cd-5d2d5EXAMPLE",
        "lastStatus": "RUNNING",
        "healthStatus": "UNKNOWN",
        "memory": "string",
        "name": "aws-gd-agent"
      }
    ]
```

Come posso sapere se il GuardDuty security agent è in esecuzione su un'istanza di contenitore EC2?

Esegui il seguente comando per visualizzare lo stato:

```
sudo systemctl status amazon-guardduty-agent
```

Il file di registro si trova nella seguente posizione:

```
/var/log/amzn-guardduty-agent
```

Cosa succede quando non esiste un ruolo di esecuzione dell'attività per un'attività in esecuzione nel cluster?

Per le attività di Fargate, l'attività inizia senza il contenitore sidecar del GuardDuty Security Agent. La GuardDuty dashboard mostrerà che all'attività manca la protezione nella dashboard delle statistiche di copertura.

Come posso sapere se dispongo delle autorizzazioni corrette per etichettare i cluster per il monitoraggio del runtime?

Per etichettare un cluster, è necessario disporre dell'`ecs:TagResource`azione per entrambi `CreateCluster` e. `UpdateCluster`

Di seguito è riportato un frammento di una policy di esempio.

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ecs:TagResource"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
             "ecs:CreateAction" : "CreateCluster",
             "ecs:CreateAction" : "UpdateCluster",
          }
       }
```

```
    }
  ]
}
```

Cosa succede in assenza di connessione Amazon ECR?

Per le attività di Fargate, l'attività inizia senza il contenitore sidecar del GuardDuty Security Agent. La GuardDuty dashboard mostrerà che all'attività manca la protezione nella dashboard delle statistiche di copertura.

Come posso risolvere gli errori di esaurimento della memoria nelle mie attività di Fargate dopo aver abilitato il Runtime Monitoring?

Il GuardDuty security agent è un processo leggero. Tuttavia, il processo consuma ancora risorse in base alle dimensioni del carico di lavoro. Ti consigliamo di utilizzare strumenti di tracciamento delle risorse dei container, come Amazon CloudWatch Container Insights, per organizzare le GuardDuty distribuzioni nel cluster. Questi strumenti ti aiutano a scoprire il profilo di consumo del GuardDuty security agent per le tue applicazioni. È quindi possibile modificare le dimensioni dell'attività di Fargate, se necessario, per evitare potenziali condizioni di esaurimento della memoria.

Monitora i contenitori Amazon ECS con ECS Exec

Con Amazon ECS Exec, puoi interagire direttamente con i container senza dover prima interagire con il sistema operativo del container host, aprire le porte in entrata o gestire le chiavi SSH. Puoi usare ECS Exec per eseguire comandi o ottenere una shell in un container 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. E negli scenari di produzione, puoi utilizzarlo per ottenere un accesso ininterrotto ai contenitori per risolvere i problemi.

Puoi eseguire comandi in un contenitore Linux o Windows in esecuzione utilizzando ECS Exec dall'API Amazon ECS, AWS Command Line Interface (AWS CLI), dagli AWS SDK o dalla CLI di Copilot. AWS 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

Puoi anche utilizzare ECS Exec per mantenere politiche di controllo degli accessi più rigorose. Attivando selettivamente questa funzionalità, è possibile controllare chi può eseguire comandi e quali

processi possono eseguire tali comandi. Con un registro di ogni comando e del relativo output, puoi usare ECS Exec per visualizzare quali attività sono state eseguite e puoi usarlo per controllare chi ha avuto accesso CloudTrail a un contenitore.

Considerazioni

Per questo argomento, è necessario acquisire familiarità con i seguenti aspetti relativi all'utilizzo di ECS Exec:

- ECS Exec non è attualmente supportato utilizzando. AWS Management Console
- ECS Exec è supportato per le attività eseguite sulla seguente infrastruttura:
  - Container Linux su Amazon EC2 su qualsiasi AMI ottimizzata per Amazon ECS, tra cui Bottlerocket
  - Contenitori Linux e Windows su istanze esterne (Amazon ECS Anywhere)
  - Contenitori Linux e Windows su AWS Fargate
  - Container Windows su Amazon EC2 per le seguenti AMI Windows ottimizzate per Amazon ECS (con la versione dell'agente del container `1.56` o successive):
    - 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
- ECS Exec e Amazon VPC
  - Se utilizzi gli endpoint dell'interfaccia Amazon VPC con Amazon ECS, devi creare tali endpoint per Session Manager di Systems Manager (`ssmmessages`). Per ulteriori informazioni sugli endpoint VPC di Systems Manager, vedere Utilizzare AWS PrivateLink per configurare un endpoint VPC per Session Manager nella Guida per l'utente.AWS Systems Manager
  - Se utilizzi gli endpoint dell'interfaccia Amazon VPC con Amazon ECS e AWS KMS key per la crittografia, devi creare tali endpoint per AWS KMS key. Per ulteriori informazioni, consulta Connessione a AWS KMS key mediante un endpoint VPC nella Guida per gli sviluppatori di AWS Key Management Service .
  - Se hai attività eseguite su istanze Amazon EC2, utilizza `awsvpc` la modalità di rete. Se non disponi di accesso a Internet (ad esempio se non sei configurato per utilizzare un gateway NAT), devi creare l'interfaccia Amazon VPC (endpoint) per Systems Manager Session Manager (). `ssmmessages` Per ulteriori informazioni sulle considerazioni relative alla modalità di rete

`awsvpc`, consulta Considerazioni. Per ulteriori informazioni sugli endpoint VPC di Systems Manager, vedere Utilizzare AWS PrivateLink per configurare un endpoint VPC per Session Manager nella Guida per l'utente.AWS Systems Manager

- ECS Exec e SSM
  - Quando un utente esegue comandi su un container utilizzando ECS Exec, questi comandi vengono eseguiti come utente `root`. SSM Agent e i relativi processi figlio vengono eseguiti come root anche quando si specifica un ID utente per il container.
  - L'agente SSM richiede la possibilità di scrivere sul file system del contenitore per creare le directory e i file richiesti. Pertanto, la specifica del file system root di sola lettura utilizzando il parametro di definizione di attività `readonlyRootFilesystem`, o qualsiasi altro metodo, non è supportata.
  - Sebbene l'avvio di sessioni di SSM al di fuori dell'operazione `execute-command` sia possibile, le sessioni non saranno registrate e saranno conteggiate rispetto al limite di sessione. Si consiglia di limitare questo accesso negando l'operazione `ssm:start-session` con una policy IAM. Per ulteriori informazioni, consulta Limitazione dell'accesso all'operazione Avvia sessione.
- Le seguenti funzionalità funzionano come contenitore sidecar. Pertanto, è necessario specificare il nome del contenitore su cui eseguire il comando.
  - Monitoraggio del runtime
  - Service Connect
- Gli utenti possono eseguire tutti i comandi disponibili nel contesto del container. Le seguenti operazioni potrebbero causare processi orfani e zombie: terminare il processo principale del container, terminare l'agente dei comandi ed eliminare le dipendenze. Per ripulire i processi zombie, ti consigliamo di aggiungere il flag `initProcessEnabled` alla definizione di attività.
- ECS Exec utilizza parte della CPU e della memoria. Sarà necessario adattare questi valori quando si specificano le allocazioni di risorse CPU e memoria nella definizione di attività.
- È necessario utilizzare la AWS CLI versione `1.22.3` o successiva o la AWS CLI versione `2.3.6` o successiva. Per informazioni su come aggiornare il AWS CLI, vedere Installazione o aggiornamento della versione più recente di AWS CLI nella Guida per l'AWS Command Line Interface utente versione 2.
- Puoi avere solo una sessione ECS Exec per ogni spazio dei nomi dell'ID processo (PID). Se condividi uno spazio dei nomi PID in un'attività, puoi avviare le sessioni ECS Exec solo in un container.
- La sessione di ECS Exec ha un valore di timeout di inattività pari a 20 minuti. Questo valore non può essere modificato.

- Non è possibile attivare ECS Exec per le attività esistenti. Il servizio può essere attivato solo per le nuove attività.
- Non è possibile utilizzare ECS Exec quando si avvia un'attività su un cluster che utilizza la scalabilità gestita con posizionamento asincrono (avvia un'attività senza istanza). `run-task`
- Non è possibile eseguire ECS Exec su contenitori Microsoft Nano Server.

Prerequisiti

Prima di iniziare a utilizzare ECS Exec, assicurati di aver completato queste azioni:

- Istalla e configura la AWS CLI. Per ulteriori informazioni, consulta AWS CLI.
- Installa il plug-in Session Manager per. AWS CLI Per ulteriori informazioni, consulta Installazione del plug-in Session Manager per AWS CLI.
- Devi utilizzare un ruolo attività con le autorizzazioni appropriate per ECS Exec. Per ulteriori informazioni, consulta Ruolo IAM dell'attività.
- ECS Exec ha requisiti di versione a seconda che i tuoi processi siano ospitati 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 dell'agente 1.50.2 o successiva. Per ulteriori informazioni, consulta AMI ottimizzate per Amazon ECS.
  - Se si utilizza AWS Fargate, è necessario utilizzare una versione della piattaforma `1.4.0` o superiore (Linux) o `1.0.0` (Windows). Per ulteriori informazioni, consulta Versioni della piattaforma AWS Fargate.

Architettura

ECS Exec utilizza AWS Systems Manager (SSM) Session Manager per stabilire una connessione con il contenitore in esecuzione e utilizza le policy AWS Identity and Access Management (IAM) per controllare l'accesso ai comandi in esecuzione in un contenitore in esecuzione. Ciò è reso possibile collegando i file binari di SSM Agent necessari nel container. L'Amazon ECS o AWS Fargate l'agente è responsabile dell'avvio dell'agente principale SSM all'interno del contenitore insieme al codice dell'applicazione. Per ulteriori informazioni, consulta Systems Manager Session Manager.

Puoi controllare quale utente ha effettuato l'accesso al contenitore utilizzando l'`ExecuteCommand`evento AWS CloudTrail e registrare ogni comando (e il relativo output) su

Amazon S3 o Amazon CloudWatch Logs. Per crittografare i dati tra il client locale e il contenitore con la tua chiave di crittografia, devi fornire la chiave AWS Key Management Service ()AWS KMS.

Utilizzo di ECS Exec

Modifiche facoltative alla definizione di attività

Se si imposta il parametro di definizione dell'attività `initProcessEnabled` su`true`, viene avviato il processo di inizializzazione all'interno del contenitore. Ciò rimuove tutti i processi secondari rilevati dall'agente SSM zombie. Il seguente comando fornisce un esempio.

```
{
    "taskRoleArn": "ecsTaskRole",
    "networkMode": "awsvpc",
    "requiresCompatibilities": [
        "EC2",
        "FARGATE"
    ],
    "executionRoleArn": "ecsTaskExecutionRole",
    "memory": ".5 gb",
    "cpu": ".25 vcpu",
    "containerDefinitions": [
        {
            "name": "amazon-linux",
            "image": "amazonlinux:latest",
            "essential": true,
            "command": ["sleep","3600"],
            "linuxParameters": {
                "initProcessEnabled": true
            }
        }
    ],
    "family": "ecs-exec-task"
}
```

Attivazione di ECS Exec per attività e servizi

Puoi attivare la funzionalità ECS Exec per i tuoi servizi e le tue attività autonome specificando il `--enable-execute-command` flag quando usi uno dei seguenti AWS CLI comandi:,, o. [create-service](#)[update-service](#)[start-task](#)[run-task](#)

Ad esempio, se si esegue il comando seguente, la funzionalità ECS Exec viene attivata per un servizio appena creato che viene eseguito su Fargate. Per ulteriori informazioni sulla creazione dei servizi, consulta create-service.

```
aws ecs create-service \
    --cluster cluster-name \
    --task-definition task-definition-name \
    --enable-execute-command \
    --service-name service-name \
    --launch-type FARGATE \
     --network-configuration
 "awsvpcConfiguration={subnets=[subnet-12344321],securityGroups=[sg-12344321],assignPublicIp=EN
 \
    --desired-count 1
```

Dopo aver abilitato ECS Exec per un'attività, è possibile emettere il comando seguente per confermare che l'attività è pronta per l'uso. Se la proprietà `lastStatus` di `ExecuteCommandAgent` è riportata come `RUNNING` e la proprietà `enableExecuteCommand` è impostata su `true`, allora il processo è pronto.

```
aws ecs describe-tasks \
    --cluster cluster-name \
    --tasks task-id
```

Il seguente frammento di output è un esempio di cosa potresti vedere.

```
{
    "tasks": [
        {
            ...
            "containers": [
                {
                    ...
                    "managedAgents": [
                        {
                            "lastStartedAt": "2021-03-01T14:49:44.574000-06:00",
                            "name": "ExecuteCommandAgent",
                            "lastStatus": "RUNNING"
                        }
                    ]
                }
```

```
            ],
            ...
            "enableExecuteCommand": true,
            ...
        }
    ]
}
```

Esecuzione di comandi tramite ECS Exec

Dopo aver confermato che `ExecuteCommandAgent` è in esecuzione, è possibile aprire una shell interattiva sul container utilizzando il seguente comando. Se i 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 task-id.

Il *task-id* è l'Amazon Resource Name (ARN) dell'attività.

```
aws ecs execute-command --cluster cluster-name \
    --task task-id \
    --container container-name \
    --interactive \
    --command "/bin/sh"
```

Registrazione tramite ECS Exec

Attivazione della registrazione delle attività e dei servizi

> **Important**
>
> Per ulteriori informazioni sui CloudWatch prezzi, consulta la sezione Prezzi. CloudWatch Amazon ECS offre anche parametri di monitoraggio che vengono forniti senza costi aggiuntivi. Per ulteriori informazioni, consulta Monitora Amazon ECS utilizzando CloudWatch .

Amazon ECS fornisce una configurazione predefinita per i comandi di registrazione eseguiti con ECS Exec inviando i log ai CloudWatch log utilizzando il driver di `awslogs` log configurato nella

definizione dell'attività. Se desideri fornire una configurazione personalizzata, la AWS CLI supporta `--configuration` per entrambi i comandi `create-cluster` e `update-cluster`. È anche importante sapere che l'immagine del contenitore richiede `script` e deve essere installata `cat` 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 su. `CloudWatchEncryptionEnabled` `true` Per informazioni su come crittografare il gruppo di log, consulta Crittografare i dati di registro in CloudWatch Logs using AWS Key Management Service, nella Guida per l'utente.Amazon CloudWatch Logs

```
aws ecs create-cluster \
    --cluster-name cluster-name \
    --configuration executeCommandConfiguration="{ \
        kmsKeyId=string, \
        logging=OVERRIDE, \
        logConfiguration={ \
            cloudWatchLogGroupName=cloudwatch-log-group-name, \
            cloudWatchEncryptionEnabled=true, \
            s3BucketName=s3-bucket-name, \
            s3EncryptionEnabled=true, \
            s3KeyPrefix=demo \
        } \
    }"
```

La proprietà `logging` determina il comportamento della funzionalità di registrazione di ECS Exec:

- `NONE`: la registrazione è disattivata.
- `DEFAULT`: i log vengono inviati al driver `awslogs` configurato. Se il driver non è configurato, non viene salvato alcun registro.

- `OVERRIDE`: i log vengono inviati al bucket Amazon CloudWatch Logs fornito LogGroup, Amazon S3 o a entrambi.

Autorizzazioni IAM richieste per Amazon CloudWatch Logs o Amazon S3 Logging

Per abilitare la registrazione, il ruolo del processo Amazon ECS a cui si fa riferimento nella definizione di attività deve disporre di autorizzazioni aggiuntive. Queste autorizzazioni aggiuntive possono essere aggiunte al ruolo del processo sotto forma di policy. Sono diversi a seconda che tu indirizzi i log ad Amazon CloudWatch Logs o Amazon S3.

Amazon CloudWatch Logs

La seguente policy di esempio aggiunge le autorizzazioni Amazon CloudWatch Logs richieste.

```
{
   "Version": "2012-10-17",
   "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:DescribeLogGroups"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
               "logs:CreateLogStream",
               "logs:DescribeLogStreams",
               "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:region:account-id:log-group:/aws/
ecs/cloudwatch-log-group-name:*"
        }
   ]
}
```

Amazon S3

La policy dell'esempio seguente aggiunge le autorizzazioni Amazon S3 necessarie.

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
         {
             "Effect": "Allow",
             "Action": [
                "s3:GetBucketLocation"
             ],
             "Resource": "*"
         },
         {
            "Effect": "Allow",
            "Action": [
                "s3:GetEncryptionConfiguration"
            ],
            "Resource": "arn:aws:s3:::s3-bucket-name"
         },
         {
             "Effect": "Allow",
             "Action": [
                 "s3:PutObject"
             ],
             "Resource": "arn:aws:s3:::s3-bucket-name/*"
         }
     ]
    }
```

Autorizzazioni IAM necessarie per la crittografia utilizzando le proprie AWS KMS key (chiave KMS)

Per impostazione predefinita, i dati trasferiti tra il client locale e il contenitore utilizzano la crittografia TLS 1.2 che fornisce. AWS Per crittografare ulteriormente i dati utilizzando la propria chiave KMS, è necessario creare una chiave KMS e aggiungere l'autorizzazione `kms:Decrypt` al ruolo IAM del processo. Questa autorizzazione sarà utilizzata dal tuo container per decrittare i dati. Per ulteriori informazioni sulla creazione di una chiave KMS, consulta Creazione di chiavi.

Aggiungete la seguente policy in linea al vostro ruolo Task IAM che richiede le AWS KMS autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni ECS Exec.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt"
            ],
            "Resource": "kms-key-arn"
        }
    ]
}
```

Per crittografare i dati utilizzando la propria chiave KMS, all'utente o al gruppo che utilizza l'operazione `execute-command` deve essere concessa l'autorizzazione `kms:GenerateDataKey`.

La seguente policy di esempio per l'utente o il gruppo contiene l'autorizzazione necessaria per utilizzare la propria chiave KMS. È necessario specificare l'Amazon Resource Name (ARN) della chiave KMS.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:GenerateDataKey"
            ],
            "Resource": "kms-key-arn"
        }
    ]
}
```

Utilizzo delle policy IAM per limitare l'accesso a ECS Exec

Limiti l'accesso degli utenti all'azione dell'API execute-command utilizzando una o più delle seguenti chiavi di condizione della policy IAM:

- `aws:ResourceTag/clusterTagKey`
- `ecs:ResourceTag/clusterTagKey`
- `aws:ResourceTag/taskTagKey`
- `ecs:ResourceTag/taskTagKey`
- `ecs:container-name`

- `ecs:cluster`
- `ecs:task`
- `ecs:enable-execute-command`

Con la seguente policy IAM di esempio, gli utenti possono eseguire comandi in container in esecuzione all'interno di processi con un tag che dispone di una chiave `environment` e un valore `development` in un cluster denominato `cluster-name`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:ExecuteCommand",
                "ecs:DescribeTasks"
            ],
         "Resource": [
                "arn:aws:ecs:region:aws-account-id:task/cluster-name/*",
                "arn:aws:ecs:region:aws-account-id:cluster/*"
            ],
            "Condition": {
                "StringEquals": {
                    "ecs:ResourceTag/environment": "development"
                }
            }
        }
    ]
}
```

Con l'esempio di policy IAM riportato di seguito, gli utenti non possono utilizzare l'API `execute-command` quando il nome del container è `production-app`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ecs:ExecuteCommand"
            ],
```

```
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ecs:container-name": "production-app"
                }
            }
        }
    ]
}
```

Con le seguenti policy IAM, gli utenti possono avviare i processi solo quando ECS Exec è disattivato.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:CreateService",
                "ecs:UpdateService"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ecs:enable-execute-command": "false"
                }
            }
        }
    ]
}
```

> **Note**
>
> Perché l'operazione API `execute-command` contiene solo risorse di processo e cluster in una richiesta, vengono valutati solo i tag di cluster e processo.

Per ulteriori informazioni sulle chiavi di condizione della policy IAM, consulta Operazioni, risorse e chiavi di condizione per Amazon Elastic Container Service in Service Authorization Reference.

Limitazione dell'accesso all'operazione Avvia sessione

Mentre è possibile avviare sessioni SSM sul container al di fuori di ECS Exec, ciò potrebbe causare potenzialmente la mancata registrazione delle sessioni. Anche le sessioni avviate al di fuori di ECS Exec vengono conteggiate per la quota di sessione. Si consiglia di limitare questo accesso negando l'operazione `ssm:start-session` direttamente per i processi Amazon ECS con una policy IAM. Puoi negare l'accesso a tutti i processi Amazon ECS o a processi specifici in base ai tag utilizzati.

Di seguito è riportato un esempio di policy IAM che nega l'accesso all'operazione `ssm:start-session` per i processi in tutte le regioni con un nome cluster specificato. Facoltativamente puoi includere un carattere jolly con *`cluster-name`*.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:StartSession",
            "Resource": [
                "arn:aws:ecs:region:aws-account-id:task/cluster-name/*",
                "arn:aws:ecs:region:aws-account-id:cluster/*"
            ]
        }
    ]
}
```

Di seguito è riportato un esempio di policy IAM che nega l'accesso all'operazione `ssm:start-session` sulle risorse in tutte le regioni con chiave di tag `Task-Tag-Key` e valore di tag `Exec-Task`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": "ssm:StartSession",
            "Resource": "arn:aws:ecs:*:*:task/*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/Task-Tag-Key": "Exec-Task"
                }
```

```
            }
        }
    ]
}
```

Per assistenza su eventuali problemi che potresti riscontrare durante l'utilizzo di Amazon ECS Exec, consulta Risoluzione dei problemi con Exec.

AWS Compute Optimizer consigli per Amazon ECS

AWS Compute Optimizer genera consigli per le dimensioni delle attività e dei container di Amazon ECS. Per ulteriori informazioni, consulta Che cos'è AWS Compute Optimizer? nella Guida per l'utente di AWS Compute Optimizer .

Suggerimenti sulle dimensioni delle attività e dei contenitori per i servizi Amazon ECS su AWS Fargate

AWS Compute Optimizer genera consigli per i servizi Amazon ECS su AWS Fargate. AWS Compute Optimizer consiglia le dimensioni della CPU e della memoria dell'attività e le dimensioni della CPU del contenitore, della memoria del contenitore e della memoria del contenitore di prenotazione. Questi suggerimenti vengono visualizzati nelle pagine seguenti della console Compute Optimizer.

- Pagina Suggerimenti per i servizi Amazon ECS su Fargate
- Pagina Dettagli dei servizi Amazon ECS su Fargate

Per ulteriori informazioni, consulta Visualizzazione dei suggerimenti per i servizi Amazon ECS su Fargate nella Guida per l'utente di AWS Compute Optimizer .

Risoluzione dei problemi legati ad Amazon ECS

Potrebbe essere necessario risolvere i problemi relativi ai sistemi di bilanciamento del carico, alle attività, ai servizi o alle istanze dei container. Questo capitolo ti aiuta a trovare informazioni di diagnostica nell'agent del container di Amazon ECS, nel daemon Docker sull'istanza di container e nel log di eventi del servizio nella console Amazon ECS.

Per informazioni sulle attività interrotte, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni
Risolvi gli errori delle attività interrotte.	Visualizzazione degli errori delle attività interrotte da Amazon ECS
Visualizza gli errori delle attività interrotte.	Risolvi gli errori relativi alle attività interrotte da Amazon ECS
Rivedi i codici di errore delle attività interrotte.	Messaggi di errore delle attività interrotte da Amazon ECS
Rivedi gli errori delle CannotPullContainer attività.	CannotPullContainer errori di attività in Amazon ECS
Visualizza le richieste di ruoli IAM per le attività.	Visualizzazione delle richieste di ruolo IAM per le attività di Amazon ECS

Per informazioni sugli errori di servizio, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni
Visualizza i messaggi relativi agli eventi di servizio.	Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS

Azione	Ulteriori informazioni
Rivedi i messaggi relativi agli eventi di servizio.	Messaggi relativi agli eventi del servizio Amazon ECS
Esamina i problemi relativi al bilanciamento del carico.	Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS
Esamina i problemi di scalabilità automatica del servizio.	Risoluzione dei problemi relativi alla scalabilità automatica del servizio in Amazon ECS

Per informazioni sugli errori di definizione delle attività, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni
Risolve l'errore di memoria di definizione dell'attività.	Risolvi gli errori di CPU o memoria non validi relativi alla definizione delle attività di Amazon ECS

Per informazioni sugli errori dell'agente Amazon ECS, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni
Visualizza i log degli agenti container Amazon ECS.	Visualizzazione dei log degli agenti container Amazon ECS
Scopri come raccogliere i log di Amazon ECS.	Raccolta dei log dei container con Amazon ECS logs collector

Azione	Ulteriori informazioni	
Recupera i dettagli diagnostici con l'agente Amazon ECS.	Recupera i dettagli diagnostici di Amazon ECS con l'introspezione degli agenti	

Per informazioni sugli errori Docker, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Usa la diagnostica Docker.	Diagnostica Docker in Amazon ECS	
Attiva la modalità di debug Docker.	Configurazione dell'output verboso dal demone Docker in Amazon ECS	
Risolvi l'errore 500 dell'API Docker.	Risolvi i problemi relativi al Docker in Amazon `API error (500): devmapper` ECS	

Per informazioni sugli errori ECS Exec e Amazon ECS Anywhere, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolvi i problemi relativi a ECS Exec.	Risolvi i problemi di Amazon ECS Exec	
Risolvi i problemi relativi ad Amazon ECS Anywhere.	Risolvi i problemi relativi ad Amazon ECS Anywhere	

Per informazioni sui problemi di limitazione, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Scopri di più sulle quote di limitazione di Fargate.	AWS Fargate limitazione delle quote	
Scopri le best practice per la limitazione di Amazon ECS.	Gestisci i problemi di limitazione di Amazon ECS	

Per informazioni sugli errori delle API, consulta una delle seguenti pagine.

Azione	Ulteriori informazioni	
Risolvi gli errori delle API.	Motivi di errore dell'API Amazon ECS	

Risolvi gli errori relativi alle attività interrotte da Amazon ECS

Quando l'attività non viene avviata, viene visualizzato un messaggio di errore nella console e nei parametri di `describe-tasks` output (StoppedReason e). StoppedCode Le sezioni seguenti forniscono informazioni aggiuntive su come risolvere i problemi relativi alle attività interrotte.

Le pagine seguenti forniscono informazioni sulle attività interrotte.

- Scopri le modifiche ai messaggi di errore relativi alle attività interrotte.

  Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore delle attività
- Visualizza le attività interrotte in modo da ottenere informazioni sulla causa.

  Visualizzazione degli errori delle attività interrotte da Amazon ECS
- Scopri i messaggi di errore relativi alle attività interrotte e i possibili motivi degli errori.

  Messaggi di errore delle attività interrotte da Amazon ECS
- Scopri come verificare la connettività delle attività interrotte e correggere gli errori.

  Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS

Amazon ECS ha interrotto gli aggiornamenti dei messaggi di errore delle attività

A partire dal 14 giugno 2024, il team di Amazon ECS modificherà i messaggi di errore dell'attività interrotta come descritto nelle tabelle seguenti. Non `stopCode` cambieranno. Se le applicazioni dipendono da stringhe di messaggi di errore esatte, è necessario aggiornare le applicazioni con le nuove stringhe. Per assistenza in caso di domande o problemi, contatta. AWS Support

> **Note**
>
> Ti consigliamo di non fare affidamento sui messaggi di errore per l'automazione, poiché i messaggi di errore sono soggetti a modifiche.

CannotPullContainerError

Vecchio messaggio di errore	Nuovo messaggio di errore	
CannotPullContainerError*: Risposta di errore dal demone: accesso pull negato per il repository, il repository non esiste o potrebbe richiedere un 'accesso docker': negato: Utente: ROLearn*	CannotPullContainerError: L'attività non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni necessarie per estrarre immagini dal registro. Risposta di errore dal demone: pull access negato per il *repository* , il repository non esiste o potrebbe richiedere 'docker login': denied: User: *ROLearn* is not authorized to perform: ecr: on resource: image perché nessuna politica basata sull'identità consente l'azione ecr BatchGetImage *:* . BatchGetImage	

Vecchio messaggio di errore	Nuovo messaggio di errore	
	CannotPullContainerError: L'attività non può estrarre l'immagine. Controlla se l'immagine esiste. Risposta di errore dal demone: accesso pull negato per il repository, il *repository* non esiste o potrebbe richiedere 'docker login': negato: l'accesso richiesto alla risorsa è negato.	
CannotPullContainerError*: Risposta di errore dal demone: Get imageURI: net/http: richiesta annullata in attesa della connessione*	CannotPullContainerError: L'operazione non può estrarre l'immagine. Controlla la configurazione della tua rete. Risposta di errore dal demone: Get *image*: net/http: richiesta annullata in attesa della connessione (Client.T imeout superato in attesa delle intestazioni)	

ResourceNotFoundException

Vecchio messaggio di errore	Nuovo messaggio di errore	
Recupero di dati segreti dalla AWS Secrets Manager regione us-west-2: *secret SercRetarn:: Secrets* Manager non ResourceN otFoundException riesce a trovare il segreto specificato.	ResourceNotFoundEx ception: l'operazione non può recuperare il segreto con l'ARN *'SercRetarn' from.* AWS Secrets Manager Controlla se il segreto esiste nella regione specifica ta. ResourceNotFoundEx	

Vecchio messaggio di errore	Nuovo messaggio di errore	
	ception: Recupero di dati segreti da una AWS Secrets Manager *regione*: *secret SercRetarn* :: Secrets ResourceNotFoundException Manager non riesce a trovare il segreto specificato.	

Visualizzazione degli errori delle attività interrotte da Amazon ECS

Se si verificano dei problemi con l'avvio di un'attività, questa potrebbe essere interrotta a causa di errori dell'applicazione o della configurazione. Ad esempio, quando esegui l'attività, questa visualizza lo stato `PENDING` e poi scompare.

Se l'attività è stata creata da un servizio Amazon ECS, le azioni intraprese da Amazon ECS per gestire il servizio sono pubblicate negli eventi del servizio. Puoi visualizzare gli eventi negli AWS Management Console,, AWS SDK AWS CLI, nell'API Amazon ECS o negli strumenti che utilizzano gli SDK e l'API. 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 computer esterni, puoi anche esaminare i log del runtime del contenitore e dell'agente Amazon ECS. Questi log si trovano sull'istanza Amazon EC2 host o sul computer esterno. Per ulteriori informazioni, consulta Visualizzazione dei log degli agenti container Amazon ECS.

Procedura

Console

AWS Management Console

I seguenti passaggi possono essere utilizzati per verificare la presenza di errori nelle attività interrotte utilizzando il nuovo. AWS Management Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).

3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Alla pagina Cluster : ***name*** (Cluster: nome), scegli la scheda Tasks (Processi).
5. Configura il filtro per visualizzare le attività interrotte. In Filtra per stato desiderato, scegli Interrotto o Qualsiasi stato desiderato.

   L'opzione Interrotto mostra le attività interrotte, mente Qualsiasi stato desiderato mostra tutte le attività.
6. Scegli il processo interrotto da ispezionare.
7. Nella riga relativa all'attività interrotta, nella colonna Stato più recente, scegli Interrotto.

   Una finestra pop-up mostra il motivo dell'interruzione.

AWS CLI

1. Elenca i processi arrestati in un cluster. L'output contiene l'Amazon Resource Name (ARN) del processo, necessario per descrivere il processo.

   ```
   aws ecs list-tasks \
        --cluster cluster_name \
        --desired-status STOPPED \
        --region region
   ```
2. Descrivi l'attività interrotta per recuperare le informazioni. Per ulteriori informazioni, consulta describe-tasks nel riferimento. AWS Command Line Interface

   ```
   aws ecs describe-tasks \
        --cluster cluster_name \
        --tasks arn:aws:ecs:region:account_id:task/cluster_name/task_ID \
        --region region
   ```

Utilizzate i seguenti parametri di output.

- `stopCode`- Il codice di arresto indica, ad esempio, il motivo per cui un'attività è stata interrotta ResourceInitializationError
- `StoppedReason`- Il motivo per cui l'attività è stata interrotta.
- `reason`(nella `containers` struttura) - Il motivo fornisce ulteriori dettagli sul contenitore fermo.

Passaggi successivi

Visualizza le attività interrotte in modo da ottenere informazioni sulla causa. Per ulteriori informazioni, consulta Messaggi di errore delle attività interrotte da Amazon ECS.

Messaggi di errore delle attività interrotte da Amazon ECS

Di seguito sono riportati i possibili messaggi di errore che potresti ricevere quando l'attività si interrompe in modo imprevisto.

Per verificare la presenza di un messaggio di errore tra le attività interrotte utilizzando il AWS Management Console, vedere. Visualizzazione degli errori delle attività interrotte da Amazon ECS

Ai codici di errore delle attività interrotte è associata una categoria, ad esempio "ResourceInitializationErrore». Per ulteriori informazioni su ciascuna categoria, consulta quanto segue:

Categoria	Ulteriori informazioni
TaskFailedToStart	Risoluzione degli errori di Amazon ECS TaskFailedToStart
ResourceInitializationErrore	Risoluzione degli errori di Amazon ECS ResourceInitializationError
ResourceNotFoundException	Risoluzione degli errori di Amazon ECS ResourceNotFoundException
SpotInterruptionErrore	Risoluzione degli errori di Amazon ECS SpotInterruption
InternalError	Risoluzione degli errori di Amazon ECS InternalError
OutOfMemoryError	Risoluzione degli errori di Amazon ECS OutOfMemoryError

Categoria	Ulteriori informazioni
ContainerRuntimeErrore	Risoluzione degli errori di Amazon ECS Container RuntimeError
ContainerRuntimeTimeoutError	Risoluzione degli errori di Amazon ECS Container RuntimeTimeoutError
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
CannotPullContenitore	CannotPullContainer errori di attività in Amazon ECS

Risoluzione degli errori di Amazon ECS TaskFailedToStart

Di seguito sono riportati alcuni messaggi `TaskFailedToStart` di errore e azioni che puoi intraprendere per correggere gli errori.

`Errore EC2 imprevisto durante il tentativo di creare un'interfaccia di rete con l'assegnazione dell'IP pubblico abilitata nella sottorete 'subnet-id'`

Ciò accade quando un'operazione Fargate che utilizza la modalità di `awsvpc` rete ed è eseguita in una sottorete con un indirizzo IP pubblico e la sottorete non dispone di indirizzi IP sufficienti.

Il numero di indirizzi IP disponibili è disponibile nella pagina dei dettagli della sottorete nella console Amazon EC2 o utilizzando. `describe-subnets` Per ulteriori informazioni, consulta Visualizza la sottorete nella Amazon VPC User Guide.

Per risolvere questo problema, puoi creare una nuova sottorete in cui eseguire l'attività.

InternalError: <reason>

Questo errore si verifica quando viene richiesto un allegato ENI. Amazon EC2 gestisce in modo asincrono il provisioning dell'ENI. Il processo di provisioning richiede del tempo. Amazon ECS raggiunge un valore di timeout in caso di lunghi tempi di attesa o errori non segnalati. Ci sono momenti in cui viene eseguito il provisioning dell'ENI, ma il report arriva ad Amazon ECS dopo il timeout dell'errore. In questo caso, Amazon ECS rileva l'errore segnalato dell'attività con un'ENI in uso.

La definizione dell'attività selezionata non è compatibile con la strategia di elaborazione selezionata

Questo errore si verifica quando si sceglie una definizione di attività con un tipo di avvio che non corrisponde al tipo di capacità del cluster. Per ulteriori informazioni, consulta Tipi di avvio di Amazon ECS. È necessario selezionare una definizione di attività che corrisponda al provider di capacità assegnato al cluster.

Risoluzione degli errori di Amazon ECS ResourceInitializationError

Di seguito sono riportati alcuni messaggi `ResourceInitialization` di errore e le azioni che è possibile eseguire per correggere gli errori.

impossibile recuperare i segreti o l'autenticazione del registro: l'attività non può recuperare l'autenticazione del registro da Amazon ECR

Questo errore si verifica quando yourtask non riesce a recuperare l'immagine definita nella definizione dell'attività.

Questo problema è causato da uno dei seguenti motivi:

Causa dell'errore..	Esegui questa operazione…	
Problema di connettività di rete tra l'endpoint VPC Amazon ECR e l'attività.	Verifica la connettività tra l'attività e l'endpoint Amazon VPC:. Verifica dell'interruzione	

Causa dell'errore..	Esegui questa operazione…	
Il problema è un problema di rete quando viene visualizzata una delle seguenti stringhe nel messaggio di errore: • digita tcp • componi udp • <ip>:<port>: timeout di I/O • net/http: timeout dell'hand shake TLS • read: timeout della connessione • Client.Timeout superato in attesa degli header • net/http: richiesta annullata in attesa della connessione • segnale: ucciso • scadenza del contesto superata	della connettività delle attività da parte di Amazon ECS	

Causa dell'errore..	Esegui questa operazione…	
Il ruolo definito nella definizione dell'attività non dispone delle autorizzazioni per Amazon ECR.	Aggiungi le autorizzazioni richieste al ruolo di esecuzione dell'attività. L'attività utilizza uno dei seguenti ruoli: • Per le attività con il tipo di avvio Fargate, questo è il ruolo di esecuzione dell'attività. Per informazioni, consulta Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia. • Per le attività con il tipo di avvio EC2, questo è il ruolo dell'istanza del contenitore. Per informazioni, consulta Autorizzazioni Amazon ECR.	

Causa dell'errore..	Esegui questa operazione…	
L'ARN dell'immagine non esiste	Visualizzate l'immagine, quindi verificate quanto segue: Per informazioni sulla visualizzazione delle immagini, consulta Visualizzazione dei dettagli delle immagini in Amazon ECR nella Amazon Elastic Container Registry User Guide. • L'immagine si trova nella stessa regione dell'attività. Spingi l'immagine nella regione corretta. Quindi, aggiorna l'attività con la nuova immagine ARN. Per informazioni sull'invio di un'immagine, consulta Pushing an image to an Amazon ECR repository nella Amazon ECR User Guide Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition nel riferimento all'API di Amazon Elastic Container Service.	

Causa dell'errore..	Esegui questa operazione…	
	• La definizione dell'attività ha l'immagine ARN errata. Aggiorna la definizione dell'attività. Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition nel riferimento all'API di Amazon Elastic Container Service.	

impossibile recuperare i segreti o l'autenticazione del registro: impossibile recuperare i segreti da ssm: l'attività non può estrarre il '***SecretName***' segreto da Systems Manager

Questo errore si verifica quando l'attività non è in grado di estrarre l'immagine definita nella definizione dell'attività utilizzando le credenziali in Systems Manager.

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 Systems Manager e l'attività. Il problema è un problema di rete quando viene visualizzata una delle seguenti stringhe nel messaggio di errore: • digita tcp • componi udp	Verificare la connettività tra l'attività e l'endpoint Systems Manager:Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS.	

Causa dell'errore..	Esegui questa operazione…	
• <ip>:<port>: timeout di I/O • net/http: timeout dell'hand shake TLS • read: timeout della connessione • Client.Timeout superato in attesa degli header • net/http: richiesta annullata in attesa della connessione • segnale: ucciso • scadenza del contesto superata		
Il ruolo definito nella definizio ne dell'attività non dispone delle autorizzazioni per Secrets Manager.	Aggiungere le autorizza zioni necessarie di Systems Manager al ruolo di esecuzion e dell'attività. Per ulteriori informazioni, consulta Autorizzazioni Secrets Manager o Systems Manager.	
L'ARN segreto non esiste	Verifica che l'ARN esista. Per informazioni, vedere Searching for Systems Manager parameters nella Guida per l'AWS Systems Manager utente.	

impossibile recuperare segreti o autenticazione del registro: impossibile recuperare segreti da asm: l'attività non può estrarre il segreto '***SecretarN***' da Secrets Manager

Questo errore si verifica quando l'attività Fargate non riesce a recuperare l'immagine definita nella definizione dell'attività utilizzando le credenziali in Secrets Manager.

Questo problema è causato da uno dei seguenti motivi:

Causa dell'errore..	Esegui questa operazione…	
Problema di connettività di rete tra l'endpoint VPC di Secrets Manager e l'attività. Il problema è un problema di rete quando viene visualizzata una delle seguenti stringhe nel messaggio di errore: • digita tcp • componi udp • <ip>:<port>: timeout di I/O • net/http: timeout dell'hand shake TLS • read: timeout della connessione • Client.Timeout superato in attesa degli header • net/http: richiesta annullata in attesa della connessione • segnale: ucciso • scadenza del contesto superata	Verifica la connettività tra l'attività e l'endpoint Secrets Manager. Per ulteriori informazioni, consulta Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS.	
Il ruolo di esecuzione dell'attività definito nella definizione dell'attività non dispone delle autorizzazioni per Secrets Manager.	Aggiungi le autorizzazioni richieste per Secrets Manager al ruolo di esecuzione dell'attività. Per ulteriori informazioni, consulta Autorizzazioni Secrets Manager o Systems Manager.	

Causa dell'errore..	Esegui questa operazione…	
L'ARN segreto non esiste	Verifica che l'ARN esista in Secrets Manager. Per informazioni sulla visualizz azione delle immagini, consulta Find secrets in Secrets Manager nella Secrets Manager Developer Guide.	

impossibile recuperare i segreti o l'autenticazione del registro: l'attività non può recuperare il segreto '*SecretarN*' da Secrets Manager

Questo errore si verifica quando l'attività non è in grado di estrarre l'immagine definita nella definizione dell'attività utilizzando le credenziali in Secrets Manager.

L'errore indica che esiste un problema di connettività di rete tra l'endpoint VPC di Systems Manager e l'attività.

Per informazioni su come verificare la connettività tra l'attività e l'endpoint, vedere. Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS

impossibile scaricare i file env: l'attività non può scaricare i file delle variabili di ambiente da Amazon S3

Questo errore si verifica quando l'attività non è in grado di scaricare il file di ambiente da Amazon S3.

Causa dell'errore..	Esegui questa operazione…	
Problema di connettività di rete tra l'attività e Amazon S3.	Verifica la connettività tra l'attività e l'endpoint Amazon S3:. Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS	
Il ruolo definito nella definizio ne dell'attività non dispone	Aggiungi l'autorizzazione Amazon S3 al ruolo. Per ulteriori informazioni, consulta	

Causa dell'errore..	Esegui questa operazione…
delle autorizzazioni per Amazon S3.	Autorizzazioni per lo storage di file Amazon S3.

impossibile convalidare logger args: l'attività non riesce a trovare il nome del ***gruppo CloudWatch*** Logs definito nella definizione dell'attività. C'è un problema di connessione tra l'attività e. CloudWatch

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.

Per risolvere il problema, è possibile eseguire una delle seguenti opzioni:

Per utilizzare questa opzione...	Esegui questa operazione…
Aggiorna la definizione dell'attività per includere la configurazione del gruppo di log nella definizione del contenitore.	Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition nel riferimento all'API di Amazon Elastic Container Service.
Crea il gruppo di log in CloudWatch	a. Esegui il comando seguente per ottenere il nome del gruppo di log. `aws ecs describe-task-definition \ --task-definition task-def-name | jq -r.taskDefinition.containerDefinitions[].logConfiguration`

Per utilizzare questa opzione...	Esegui questa operazione…	
	b. Crea il gruppo di log. Per ulteriori informazioni, consulta Creare un gruppo di log in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.	

impossibile inizializzare il driver di registrazione

Questo errore si verifica quando l'attività non riesce a trovare il gruppo di CloudWatch log definito nella definizione dell'attività.

L'errore indica che il CloudWatch gruppo nella definizione dell'attività non esiste.

Per risolvere il problema, è possibile eseguire una delle seguenti opzioni:

Per utilizzare questa opzione...	Esegui questa operazione…	
Aggiorna la definizione dell'attività per includere la configurazione del gruppo di log nella definizione del contenitore.	Per informazioni sull'aggiornamento della definizione del task, consulta Aggiornamento di una definizione di attività Amazon ECS tramite la console o RegisterTaskDefinition nel riferimento all'API di Amazon Elastic Container Service.	
Crea il gruppo di log in CloudWatch	a. Esegui il comando seguente per ottenere il nome del gruppo di log. `aws ecs describe-task-definition \ --task-definition task-def-name | jq -r.taskDe`	

Per utilizzare questa opzione...	Esegui questa operazione…
	`finition.container Definitions[].logC onfiguration` b. Crea il gruppo di log. Per ulteriori informazioni, consulta Creare un gruppo di log in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

non è riuscito a richiamare i comandi EFS utils per configurare i volumi EFS

I seguenti problemi potrebbero impedirti di montare i volumi Amazon EFS sulle tue richieste:

- Il file system Amazon EFS non è configurato correttamente.
- L'attività non dispone delle autorizzazioni richieste.
- Esistono problemi relativi alle configurazioni di rete e VPC.

Per informazioni su come eseguire il debug e risolvere questo problema, consulta Perché non posso montare i miei volumi Amazon EFS sulle mie AWS Fargate attività su AWS re:POST.

Risoluzione degli errori di Amazon ECS ResourceNotFoundException

Di seguito sono riportati alcuni messaggi `ResourceNotFoundException` di errore e le azioni che è possibile eseguire per correggere gli errori.

L'operazione non può recuperare il segreto con l'ARN *`'SercRetarn' from.`* AWS Secrets Manager Controlla se il segreto esiste nella regione specificata.

Questo errore si verifica quando l'attività non può recuperare il segreto da Secrets Manager. Ciò significa che il segreto specificato nella definizione dell'attività (e contenuto nel messaggio di errore) non esiste in Secrets Manager.

La regione è presente nel messaggio di errore.

Recupero di dati segreti da una AWS Secrets Manager *`regione`*: secret *`SercreTarn`*:: Secrets ResourceNotFoundException Manager non riesce a trovare il segreto specificato.

Per informazioni su come trovare un segreto, consulta Trova segreti AWS Secrets Manager nella Guida per l'utente.AWS Secrets Manager

Utilizzate la tabella seguente per determinare e risolvere l'errore.

Problema	Azioni
Il segreto si trova in una regione diversa dalla definizione dell'attività.	a. Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni, consulta Creare un AWS Secrets Manager segreto. b. Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta la Aggiornamento di una definizione di attività Amazon ECS tramite la console nostra RegisterTaskdefinizione nel riferimento all'API di Amazon Elastic Container Service.
La definizione dell'attività ha un ARN segreto errato. Il segreto corretto esiste in Secrets Manager.	Aggiorna la definizione dell'attività con il segreto corretto. Per ulteriori informazioni, consulta la Aggiornamento di una definizione di attività Amazon ECS tramite la console nostra RegisterTaskdefinizione nel riferimento all'API di Amazon Elastic Container Service.
Il segreto non esiste più.	a. Crea il segreto nella stessa regione dell'attività. Per ulteriori informazioni,

Problema	Azioni	
	consulta Creare un AWS Secrets Manager segreto. b. Aggiorna la definizione dell'attività con il nuovo segreto. Per ulteriori informazioni, consulta la Aggiornamento di una definizione di attività Amazon ECS tramite la console nostra RegisterTaskdefinizione nel riferimento all'API di Amazon Elastic Container Service.	

Risoluzione degli errori di Amazon ECS SpotInterruption

L'`SpotInterruption`errore ha diverse ragioni per i tipi di lancio di Fargate ed EC2.

Tipo di avvio di Fargate

L'`SpotInterruption`errore si verifica quando non c'è capacità Fargate Spot o quando Fargate riprende la capacità Spot.

È possibile eseguire le attività in più zone di disponibilità per consentire una maggiore capacità.

Tipo di avvio EC2

Questo errore si verifica quando non ci sono istanze Spot disponibili o EC2 recupera la capacità delle istanze Spot.

Puoi far funzionare le tue istanze in più zone di disponibilità per consentire una maggiore capacità.

Risoluzione degli errori di Amazon ECS InternalError

Vale per: Tipo di lancio Fargate

L'`InternalError`errore che 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 eseguire il debug e risolvere questo problema, consulta Come posso risolvere un task Amazon ECS che non è stato avviato in un cluster ECS su re:post. AWS

Risoluzione degli errori di Amazon ECS OutOfMemoryError

Di seguito sono riportati alcuni messaggi OutOfMemoryError di errore e le azioni che è possibile eseguire per correggere gli errori.

contenitore interrotto a causa dell'utilizzo della memoria

Questo errore si verifica quando un container si chiude a causa di processi al suo interno che consumano più memoria di quella allocata nella definizione di attività.

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.

ContainerRuntimeErrore

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.

Impossibile passare alla modalità di esecuzione; timeout scaduto dopo un'attesa di 1 minuto o errore di timeout Docker

Questo errore si verifica quando un container non è in grado di passare a uno stato `RUNNING` o `STOPPED` entro il periodo di timeout. Il motivo e il valore di timeout vengono forniti nel messaggio di errore.

Risoluzione degli errori di Amazon ECS CannotStartContainerError

Di seguito sono riportati alcuni messaggi CannotStartContainerError di errore e le azioni che è possibile eseguire per correggere gli errori.

impossibile ottenere lo stato del contenitore: <reason>

Questo errore si verifica quando un container non può essere avviato.

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.

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.

CannotInspectContainerError

Questo errore si verifica quando l'agente del container non può descrivere il container tramite il runtime del container.

Quando si utilizza una versione della piattaforma `1.3` o precedente, l'agente Amazon ECS restituisce il motivo da Docker.

Quando si utilizza la versione della piattaforma `1.4.0` o successiva (Linux) `1.0.0` o successiva (Windows), l'agente Fargate restituisce il motivo da. `containerd`

Per informazioni su come eseguire il debug e risolvere questo problema, consulta Why is my Amazon ECS task Stopped on AWS re:post.

Risoluzione degli errori di Amazon ECS CannotCreateVolumeError

Di seguito sono riportati alcuni messaggi CannotCreateVolumeError di errore e le azioni che è possibile eseguire per correggere gli errori.

CannotCreateVolumeError

Questo errore si verifica quando l'agente non è in grado di creare il montaggio del volume specificato nella definizione di attività.

Questo errore si verifica solo se si utilizza la versione `1.4.0` della piattaforma o versioni successive (Linux) o `1.0.0` o versioni successive (Windows).

Per informazioni su come eseguire il debug e risolvere questo problema, consulta Why is my Amazon ECS task Stopped on AWS re:post.

CannotPullContainer errori di attività in Amazon ECS

I seguenti errori indicano che l'attività non è stata avviata perché Amazon ECS non è in grado di recuperare l'immagine del contenitore specificata.

> **Note**
>
> La piattaforma Fargate versione 1.4 tronca i messaggi di errore troppo lunghi.

Errori

- L'attività non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni per estrarre immagini dal registro
- L'attività non può estrarre l'immagine. Controlla la configurazione della tua rete
- Errore API (500): ottieni 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/ GetImage Blob111111111: non è rimasto spazio sul dispositivo
- ERRORE: toomanyrequests: Troppe richieste o hai raggiunto il limite di pull rate.
- Risposta all'errore dal demone: Get url: net/http: richiesta annullata in attesa della connessione
- ref pull è stato riprovato 1 volta/e: copia non riuscita: apertura fallita httpReaderSeeker: codice di stato imprevisto
- pull access negato
- comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido o dereferenziazione del puntatore nullo

- errore durante l'estrazione dell'immagine conf/errore durante l'estrazione della configurazione dell'immagine
- Contesto annullato

L'attività non può estrarre l'immagine. Verifica che il ruolo disponga delle autorizzazioni per estrarre immagini dal registro

Questo errore indica che l'attività non può recuperare l'immagine specificata nella definizione dell'attività a causa di problemi di autorizzazione. Il messaggio di errore che fornisce l'immagine o il ruolo che causa il problema contiene informazioni aggiuntive.

«Risposta di errore dal demone: l'accesso pull negato per il *`repository`* non esiste o potrebbe richiedere 'docker login': denied: User: *`ROLearn`* is not authorized to perform: ecr: BatchGetImage on resource: *`image`* perché nessuna policy basata sull'identità consente l'azione ecr:». BatchGetImage

Per risolvere il problema:

1. Verifica che *`l'`*immagine esista nell'irepository. Per informazioni sulla visualizzazione delle immagini, consulta Visualizzazione dei dettagli delle immagini in Amazon ECR nella Amazon Elastic Container Registry User Guide.
2. Verifica che il *`role-arn`* disponga delle autorizzazioni corrette per estrarre l'immagine.

   Per informazioni su come visualizzare e modificare i ruoli, consulta Modificare un ruolo nella Guida per l'uso.AWS Identity and Access Management

   L'attività utilizza uno dei seguenti ruoli:

   - Per le attività con il tipo di avvio Fargate, questo è il ruolo di esecuzione dell'attività. Per informazioni sulle autorizzazioni aggiuntive per Amazon ECR,. Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia
   - Per le attività con il tipo di avvio 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 l'immagine. Controlla la configurazione della tua rete

Questo errore indica che l'attività non può connettersi ad Amazon ECR.

Per informazioni su come verificare e risolvere il problema, consultaVerifica dell'interruzione della connettività delle attività da parte di Amazon ECS.

Errore API (500): ottieni https://111122223333.dkr.ecr.us-east-1.amazonaws.com/v2/: net/http: richiesta annullata in attesa della connessione

Questo errore indica che la connessione è scaduta perché non esiste un percorso verso Internet.

Per risolvere questo problema, puoi procedere in questi modi:

- Per le attività in sottoreti pubbliche, puoi specificare ENABLED (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta Esecuzione di un'applicazione come attività Amazon ECS.
- Per le attività in sottoreti private, puoi specificare DISABLED (DISABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo e quindi configurare un gateway NAT nel VPC per instradare le richieste a Internet. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.

Errori API

Questo errore indica che c'è un problema di connessione con l'endpoint Amazon ECR.

Per informazioni su come risolvere questo problema, consulta How can I resolve the Amazon ECR error "CannotPullContainerError: API error» in Amazon ECS sul AWS Support sito Web.

write /var/lib/docker/tmp/ GetImage Blob111111111: non è rimasto spazio sul dispositivo

Questo errore indica che lo spazio su disco è insufficiente.

Per risolvere questo problema, libera spazio sul disco.

Se utilizzi l'AMI ottimizzata per Amazon ECS, puoi utilizzare il seguente comando per recuperare i 20 file più grandi sul tuo file system:

```
du -Sh / | sort -rh | head -20
```

Output di esempio:

```
5.7G    /var/lib/docker/
containers/50501b5f4cbf90b406e0ca60bf4e6d4ec8f773a6c1d2b451ed8e0195418ad0d2
1.2G    /var/log/ecs
```

```
594M    /var/lib/docker/devicemapper/mnt/
c8e3010e36ce4c089bf286a623699f5233097ca126ebd5a700af023a5127633d/rootfs/data/logs
...
```

In alcuni casi, il volume root potrebbe essere compilato da un contenitore in esecuzione. Se il container sta utilizzando il driver di log `json-file` predefinito senza un limite `max-size`, è probabile che il file di log sia responsabile della maggior parte dello spazio occupato. Puoi utilizzare il comando `docker ps` per verificare quale container sta utilizzando lo spazio mappando il nome della directory dall'output sopra l'ID del container. Ad esempio:

```
CONTAINER ID   IMAGE                                COMMAND             CREATED
 STATUS              PORTS                                NAMES
50501b5f4cbf   amazon/amazon-ecs-agent:latest   "/agent"            4 days ago
 Up 4 days                                            ecs-agent
```

Di default, quando utilizzi il driver di log `json-file`, Docker acquisisce l'output standard (e l'errore standard) di tutti i container e li scrive in file utilizzando il formato JSON. Puoi impostare `max-size` come opzione del driver di log per evitare che il file di log occupi troppo spazio. Per ulteriori informazioni, consulta la pagina Configura driver di log nella documentazione Docker.

La seguente è uno snippet di definizione di container che mostra come usare questa opzione:

```
{
    "log-driver": "json-file",
    "log-opts": {
        "max-size": "256m"
    }
}
```

Un'alternativa, se i log del contenitore occupano troppo spazio su disco, consiste nell'utilizzare il driver di `awslogs` registro. Il driver di registro invia i `awslogs` log a CloudWatch, il che libera lo spazio su disco che verrebbe altrimenti utilizzato per i registri del contenitore sull'istanza del contenitore. Per ulteriori informazioni, consulta Invia i log di Amazon ECS a CloudWatch .

ERRORE: toomanyrequests: Troppe richieste o hai raggiunto il limite di pull rate.

Questo errore indica che esiste una limitazione della velocità di Docker Hub.

Se ricevi uno degli errori seguenti, è probabile che si stiano per raggiungere i limiti della frequenza di Docker Hub:

Per ulteriori informazioni sui limiti di frequenza di Docker Hub, consulta Informazioni sulla limitazione della frequenza di Docker Hub.

Se hai aumentato il limite di velocità di Docker Hub e devi autenticare i file Docker pull per le istanze di container, vedi Autenticazione del registro privato per le istanze di container.

Risposta all'errore dal demone: Get ***url***: net/http: richiesta annullata in attesa della connessione

Questo errore indica che la connessione è scaduta, perché non esiste un percorso verso Internet.

Per risolvere questo problema, puoi procedere in questi modi:

- Per le attività in sottoreti pubbliche, puoi specificare ENABLED (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta Esecuzione di un'applicazione come attività Amazon ECS.
- Per le attività in sottoreti private, puoi specificare DISABLED (DISABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo e quindi configurare un gateway NAT nel VPC per instradare le richieste a Internet. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.

ref pull è stato riprovato 1 volta/e: copia non riuscita: apertura fallita httpReaderSeeker: codice di stato imprevisto

Questo errore indica che si è verificato un errore durante la copia di un'immagine.

Per risolvere questo problema, consulta uno dei seguenti articoli:

- Per i processi Fargate, consulta Come posso risolvere l'errore "cannotpullcontainererror" per i miei processi Amazon ECS su Fargate.
- Per altri processi, consulta Come posso risolvere l'errore "cannotpullcontainererror" per i miei processi Amazon ECS.

pull access negato

Questo errore indica che non è possibile accedere all'immagine.

Per risolvere questo problema, potrebbe essere necessario autenticare il client Docker con Amazon ECR. Per ulteriori informazioni, consulta Autenticazione del registro privato nella Amazon ECR User Guide.

comando pull non riuscito: panic: errore di runtime: indirizzo di memoria non valido o dereferenziazione del puntatore nullo

Questo errore indica che non è possibile accedere all'immagine a causa di un indirizzo di memoria non valido o di una dereferenziazione del puntatore pari a zero.

Per risolvere il problema:

- Verifica di disporre delle regole del gruppo di sicurezza per accedere ad Amazon S3.
- Quando si utilizzano gli endpoint del gateway, è necessario aggiungere una route nella tabella delle rotte per accedere all'endpoint.

errore durante l'estrazione dell'immagine conf/errore durante l'estrazione della configurazione dell'immagine

Questo errore indica che è stato raggiunto un limite di velocità o che c'è un errore di rete:

Per risolvere questo problema, consulta Come posso risolvere l'errore "CannotPullContainerError" nel mio task di tipo di avvio di Amazon ECS EC2.

Contesto annullato

Questo errore indica che il contesto è stato annullato.

La causa comune di questo errore è che il VPC utilizzato dal processo che non ha un percorso per estrarre l'immagine del container da Amazon ECR.

Verifica dell'interruzione della connettività delle attività da parte di Amazon ECS

Ci sono momenti in cui un'attività si interrompe a causa di un problema di connettività di rete. Potrebbe trattarsi di un problema intermittente, ma molto probabilmente è causato dal fatto che l'attività non riesce a connettersi a un endpoint.

Verifica della connettività dell'attività

È possibile utilizzare `AWSSupport-TroubleshootECSTaskFailedToStart` runbook per testare la connettività delle attività. Quando si utilizza il runbook, sono necessarie le seguenti informazioni sulle risorse:

- L'ID dell'attività

  Utilizza l'ID dell'ultima operazione non riuscita.
- Il cluster in cui si trovava l'attività

Per informazioni su come utilizzare il runnbook, consulta `AWSSupport-TroubleshootECSTaskFailedToStart`l'AWS Systems Manager Automation runbook reference.

Il runbook analizza il task. È possibile visualizzare i risultati nella sezione Output per i seguenti problemi che possono impedire l'avvio di un'attività:

- Connettività di rete al registro dei contenitori configurato
- Connettività endpoint VPC
- Configurazione delle regole del gruppo di sicurezza

Risoluzione dei problemi relativi agli endpoint VPC

Quando il risultato del `AWSSupport-TroubleshootECSTaskFailedToStart` runbook indica il problema dell'endpoint VPC, controlla la seguente configurazione:

- Il VPC in cui si crea l'endpoint deve utilizzare il DNS privato.
- Assicurati di avere un AWS PrivateLink endpoint per il servizio a cui l'attività non può connettersi nello stesso VPC dell'attività. Per ulteriori informazioni, consulta una delle seguenti pagine:

Servizio	Informazioni sugli endpoint VPC per il servizio
Amazon ECR	Endpoint VPC con interfaccia Amazon ECR ()AWS PrivateLink
Systems Manager	Crea un endpoint VPC
Secrets Manager	Utilizzo di un AWS Systems Manager endpoint VPC
CloudWatch	CloudWatch Endpoint VPC

Servizio	Informazioni sugli endpoint VPC per il servizio	
Amazon S3	AWS PrivateLink per Amazon S3	

- Configura una regola in uscita per la sottorete delle attività che consente il traffico HTTPS sulla porta 443 DNS (UDP e TCP). Per ulteriori informazioni, consulta Aggiungere regole a un gruppo di sicurezza nella Amazon Elastic Compute Cloud User Guide.
- Se la sottorete dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
  - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.
  - Una regola in entrata che consente il traffico TCP sulla porta 443.

  Per informazioni su come configurare le regole, consulta Controllare il traffico verso le sottoreti utilizzando gli ACL di rete nella Amazon Virtual Private Cloud User Guide.

Risoluzione dei problemi di rete

Quando il risultato del `AWSSupport-TroubleshootECSTaskFailedToStart` runbook indica un problema di rete, controlla la seguente configurazione:

Attività che utilizzano la modalità di rete awsvpc in una sottorete pubblica

Esegui la seguente configurazione in base al runbook:

- Per le attività in sottoreti pubbliche, puoi specificare ENABLED (ABILITATO) per Assegna automaticamente IP pubblico all'avvio del processo. Per ulteriori informazioni, consulta Esecuzione di un'applicazione come attività Amazon ECS.
- È necessario un gateway per gestire il traffico Internet. La tabella di routing per la sottorete delle attività deve avere un percorso per il traffico verso il gateway.

  Per ulteriori informazioni, consulta Aggiungere e rimuovere percorsi da una tabella di routing nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
Internet Gateway	0.0.0.0/0	ID del gateway Internet

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
  - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.
  - Una regola in entrata che consente il traffico TCP sulla porta 443.

  Per informazioni su come configurare le regole, consulta Controllare il traffico verso le sottoreti utilizzando gli ACL di rete nella Amazon Virtual Private Cloud User Guide.

Attività che utilizzano la modalità di rete awsvpc in una sottorete privata

Esegui la seguente configurazione in base al runbook:

- Scegli DISABLED per assegnare automaticamente l'IP pubblico all'avvio dell'attività.
- Configura un gateway NAT nel tuo VPC per indirizzare le richieste a Internet. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.
- La tabella di routing per la sottorete delle attività deve avere un percorso per il traffico verso il gateway NAT.

  Per ulteriori informazioni, consulta Aggiungere e rimuovere percorsi da una tabella di routing nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
  - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.
  - Una regola in entrata che consente il traffico TCP sulla porta 443.

  Per informazioni su come configurare le regole, consulta Controllare il traffico verso le sottoreti utilizzando gli ACL di rete nella Amazon Virtual Private Cloud User Guide.

Attività che non utilizzano la modalità di rete awsvpc in una sottorete pubblica

Esegui la seguente configurazione in base al runbook:

- Scegli Attiva per l'assegnazione automatica dell'IP in Networking for Amazon EC2 istanze quando crei il cluster.

  Questa opzione assegna un indirizzo IP pubblico all'interfaccia di rete primaria dell'istanza.
- È necessario un gateway per gestire il traffico Internet. La tabella di routing per la sottorete dell'istanza deve avere un percorso per il traffico verso il gateway.

  Per ulteriori informazioni, consulta Aggiungere e rimuovere percorsi da una tabella di routing nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT
Internet Gateway	0.0.0.0/0	ID del gateway Internet

- Se la sottorete dell'istanza dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
  - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.
  - Una regola in entrata che consente il traffico TCP sulla porta 443.

  Per informazioni su come configurare le regole, consulta Controllare il traffico verso le sottoreti utilizzando gli ACL di rete nella Amazon Virtual Private Cloud User Guide.

Attività che utilizzano la modalità di rete awsvpc in una sottorete privata

Esegui la seguente configurazione in base al runbook:

- Scegli Disattiva per l'assegnazione automatica dell'IP in Networking for Amazon EC2 istanze quando crei il cluster.
- Configura un gateway NAT nel tuo VPC per indirizzare le richieste a Internet. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC.
- La tabella di routing per la sottorete dell'istanza deve avere un percorso per il traffico verso il gateway NAT.

Per ulteriori informazioni, consulta Aggiungere e rimuovere percorsi da una tabella di routing nella Amazon Virtual Private Cloud User Guide.

Tipo di gateway	Destinazione della tabella delle rotte	Obiettivo della tabella di routing
NAT	0.0.0.0/0	ID del gateway NAT

- Se la sottorete delle attività dispone di un ACL di rete, sono necessarie le seguenti regole ACL:
  - Una regola in uscita che consente il traffico che consente il traffico sulle porte 1024-65535.
  - Una regola in entrata che consente il traffico TCP sulla porta 443.

  Per informazioni su come configurare le regole, consulta Controllare il traffico verso le sottoreti utilizzando gli ACL di rete nella Amazon Virtual Private Cloud User Guide.

Visualizzazione delle richieste di ruolo IAM per le attività di Amazon ECS

Quando utilizzi un provider per le credenziali dell'attività in un ruolo IAM, le richieste del provider vengono salvate in un registro di controllo. Il log di audit eredita le stesse impostazioni di rotazione del log dell'agente del container. È possibile impostare le variabili di configurazione dell'agente del container `ECS_LOG_ROLLOVER_TYPE`, `ECS_LOG_MAX_FILE_SIZE_MB` `ECS_LOG_MAX_ROLL_COUNT` in modo da influire sul comportamento del log di audit. Per ulteriori informazioni, consulta Parametri di configurazione del registro dell'agente container Amazon ECS.

Per l'agente del container versione 1.36.0 e successive, il log di audit si trova in `/var/log/ecs/audit.log`. Quando il log viene ruotato, viene aggiunto un timestamp nel formato *`YYYY`*`-`*`MM`*`-`*`DD`*`-`*`HH`* alla fine del nome del file di log.

Per l'agente del container versione 1.35.0 e precedenti, il log di audit si trova in `/var/log/ecs/audit.log.`*`YYYY`*`-`*`MM`*`-`*`DD`*`-`*`HH`*.

Il formato delle voci di log è il seguente:

- Timestamp
- Codice di risposta HTTP

- Indirizzo IP e numero di porta dell'origine della richiesta
- URI relativo del provider di credenziali
- L'agente utente che ha effettuato la richiesta
- L'ARN dell'attività a cui appartiene il container richiedente
- Il nome API `GetCredentials` e il numero di versione
- Il nome del cluster Amazon ECS su cui è registrata l'istanza di container
- L'ARN dell'istanza di container

Puoi utilizzare il comando seguente per visualizzare i file di log.

```
cat /var/log/ecs/audit.log.2016-07-13-16
```

Output:

```
2016-07-13T16:11:53Z 200 172.17.0.5:52444 "/v1/credentials" "python-requests/2.7.0
 CPython/2.7.6 Linux/4.4.14-24.50.amzn1.x86_64" TASK_ARN GetCredentials
 1 CLUSTER_NAME CONTAINER_INSTANCE_ARN
```

Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS

Se stai risolvendo un problema relativo a un servizio, il primo luogo in cui cercare le informazioni di diagnostica è il log di eventi dei servizi. Puoi visualizzare gli eventi di servizio utilizzando l'`DescribeServices`API AWS CLI, o utilizzando il AWS Management Console.

Quando si visualizzano i messaggi degli eventi di servizio utilizzando l'API Amazon ECS, vengono restituiti solo gli eventi dell'utilità di pianificazione del servizio. Questi includono il posizionamento dei processi e gli eventi di integrità delle istanze più recenti. Tuttavia, la console Amazon ECS visualizza gli eventi di servizio dalle seguenti origini.

- Posizionamento dei processi ed eventi di integrità delle istanze dal pianificatore di servizi Amazon ECS. Questi eventi hanno il prefisso service ***(service-name***). Per garantire che questa visualizzazione degli eventi sia utile, mostriamo solo i `100` eventi più recenti e 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, si riceve un altro messaggio relativo a un evento di servizio relativo a tale causa.

- Eventi di scalabilità automatica dei servizi Questi eventi hanno il prefisso Message. Sono visualizzati i `10` eventi di dimensionamento più recenti. Questi eventi si verificano solo quando un servizio è configurato con una policy di dimensionamento di Application Auto Scaling.

Utilizza la procedura seguente per visualizzare i messaggi relativi a eventi del servizio corrente.

Console

1. Apri la console all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Nel pannello di navigazione scegliere Clusters (Cluster).
3. Nella pagina Clusters (Cluster), scegli il cluster.
4. Scegli il servizio da ispezionare.
5. Scegli Deployments and events (Implementazioni ed eventi), in Events (Eventi), quindi visualizza i messaggi.

AWS CLI

Utilizza il comando describe-services per visualizzare i messaggi di evento del servizio per un servizio specificato.

L' AWS CLI esempio seguente descrive il servizio *`service-name`* nel cluster *`predefinito`*, 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:

Il servizio (*nome-servizio*) ha raggiunto uno stato costante.

Il service scheduler invia un evento di `service (`*service-name*`) has reached a steady state.` servizio quando il servizio è integro e ha il numero di attività desiderato, raggiungendo così uno stato stazionario.

Il pianificatore del servizio segnala periodicamente lo stato, quindi potresti ricevere questo messaggio più volte.

(service *service-name*) was unable to place a task because no container instance met all of its requirements.

Il service scheduler invia questo messaggio di evento quando non riesce a trovare le risorse disponibili per aggiungere un'altra attività. Le possibili cause sono:

Nessuna istanza di container trovata nel cluster

Se nessuna istanza del contenitore è registrata nel cluster in cui si tenta di eseguire un'attività, viene visualizzato questo errore. È opportuno aggiungere istanze di container al cluster. Per ulteriori informazioni, consulta Avvio di un'istanza di container Linux di Amazon ECS.

Porte insufficienti

Se la tua attività utilizza una mappatura fissa per le porte host (ad esempio, utilizza la porta 80 sull'host per un server Web), devi disporre almeno di un'istanza di container per ogni attività, poiché solo un container può utilizzare una sola porta host alla volta. È consigliabile aggiungere istanze di container al tuo cluster o ridurre il numero di attività desiderate.

Troppe porte registrate

L'istanza di contenitore più vicina per il posizionamento delle attività non può superare il limite massimo consentito di porte riservate di 100 porte host per istanza di contenitore. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione dell'attività di questa attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza del contenitore scelta. Il messaggio dell'evento del servizio avrebbe l'ID dell'istanza di container scelto come parte del messaggio riportato di seguito.

```
The closest matching container-instance is already using a port required by your
 task.
```

Memoria insufficiente

Se nella definizione di attività sono specificati 1.000 MiB di memoria e le istanze di container nel cluster hanno ognuna 1.024 MiB di memoria, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire la memoria nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

> **Note**
>
> Per ottimizzare l'utilizzo delle risorse, assegnando all'attività quanta più memoria possibile per un determinato tipo di istanza, consulta Riservare la memoria delle istanze del contenitore Amazon ECS Linux .

CPU insufficiente

Un'istanza di container dispone di 1.024 unità CPU per ogni core CPU. Se nella definizione di attività sono specificate 1.000 unità CPU e le istanze di container nel cluster hanno ognuna 1.024 unità CPU, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire le unità CPU nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

Punti di collegamento insufficienti per le interfacce di rete elastiche

Le attività che utilizzano la modalità di rete `awsvpc` ricevono ognuna una propria interfaccia di rete elastica (ENI), che è collegata all'istanza di container che la ospita. Il numero di ENI che possono essere collegate alle istanze Amazon EC2 è limitato e nel cluster non esistono istanze di 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 la sezione relativa agli Indirizzi IP per interfaccia di rete per tipo di istanza nella Guida per l'utente di Amazon EC2.
- Se hai attivato l'impostazione dell'`awsvpcTrunking`account ma non hai avviato nuove istanze di contenitore utilizzando un tipo di istanza supportato dopo l'attivazione, il limite ENI per ogni istanza di contenitore è ancora al valore predefinito. Per ulteriori informazioni, consulta la

sezione relativa agli Indirizzi IP per interfaccia di rete per tipo di istanza nella Guida per l'utente di Amazon EC2.

- Se hai fornito il consenso esplicito all'impostazione dell'account `awsvpcTrunking` e hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo il consenso esplicito, sono disponibili ENI aggiuntive. Per ulteriori informazioni, consulta Istanze supportate per interfacce di rete di container Amazon ECS potenziate.

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsvpcTrunking`, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.

Puoi aggiungere istanze di container al tuo cluster per fornire più adattatori di rete disponibili.

Attributo richiesto mancante nell'istanza di container

Alcuni parametri di definizione di attività richiedono che nell'istanza di container sia installata una specifica versione dell'API remota Docker. Altri, come le opzioni relative ai driver di log, richiedono alle istanze di container di registrare tali driver con la variabile di configurazione dell'agente `ECS_AVAILABLE_LOGGING_DRIVERS`. Se la definizione dell'attività contiene un parametro che richiede uno specifico attributo di istanza del contenitore e non disponi di istanze di contenitore disponibili in grado di soddisfare questo requisito, l'attività non può essere inserita.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di `awsvpc` rete e il tipo di avvio EC2. Il cluster che hai specificato non ha un'istanza di contenitore registrata nella stessa sottorete specificata al `awsvpcConfiguration` momento della creazione del servizio.

Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS e Configurazione dell'agente del container Amazon ECS.

(service ***`service-name`***) was unable to place a task because no container instance met all of its requirements. L'istanza di container ***`container-instance-id`*** più vicina non ha sufficienti unità CPU disponibili.

L'istanza del contenitore più vicina per il posizionamento dell'attività non contiene unità CPU sufficienti per soddisfare i requisiti nella definizione dell'attività. Esaminare i requisiti di CPU in entrambi i parametri dimensioni attività e definizione del container della definizione di attività.

(service ***service-name***) was unable to place a task because no container instance met all of its requirements. L'istanza di container ***container-instance-id*** più vicina ha riscontrato l'errore "AGENT".

L'agente del container Amazon ECS sull'istanza di container con la corrispondenza maggiore per il posizionamento del processo viene disconnesso. Se riesci a connetterti all'istanza di container tramite SSH, puoi esaminare i log dell'agente; per ulteriori informazioni, consulta Parametri di configurazione del registro dell'agente container Amazon ECS. Devi anche verificare che l'agente sia in esecuzione sull'istanza. Se stai utilizzando l'AMI ottimizzata per Amazon ECS, puoi provare ad arrestare e riavviare l'agente con il comando seguente.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS e l'AMI Amazon Linux 2023 ottimizzata per Amazon ECS

  ```
  sudo systemctl restart ecs
  ```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS

  ```
  sudo stop ecs && sudo start ecs
  ```

service (***service-name***) (instance ***instance-id***) non è integro in (elb ***elb-name***) a causa di (motivo: Instance ha superato almeno il numero di controlli sanitari consecutivi). UnhealthyThreshold

Questo servizio è registrato con un load balancer i cui controlli dello stato hanno esito negativo. Per ulteriori informazioni, consulta Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS.

(service ***service-name***) is unable to consistently start tasks successfully.

Questo servizio contiene attività che non sono state avviate dopo vari tentativi consecutivi. A questo punto, il pianificatore del servizio inizia ad aumentare in modo incrementale il tempo tra i nuovi tentativi. È consigliabile risolvere il motivo per cui le attività non vengono avviate. Per ulteriori informazioni, consulta Logica di accelerazione del servizio Amazon ECS.

Una volta aggiornato il servizio, ad esempio con una definizione aggiornata dell'attività, il pianificatore del servizio riprende il normale funzionamento.

Le operazioni di service (***nome-servizio***) sono sottoposte a limitazione. 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 AWS Support Center, accedi se necessario e scegli Crea caso. Selezionare Service limit increase (Aumento limiti del servizio). Compilare e inviare il modulo.

service (***nome-servizio***) non è riuscito ad arrestare o avviare i processi durante un'implementazione a causa della configurazione della implementazione del servizio. Aggiorna il valore minimumHealthyPercent o MaximumPercent e riprova.

Questo servizio non è riuscito ad arrestare o avviare i processi durante un'implementazione a causa della configurazione dell'implementazione. La configurazione di distribuzione è costituita dai `maximumPercent` valori `minimumHealthyPercent` e, che vengono definiti al momento della creazione del servizio. Questi valori possono essere aggiornati anche su un servizio esistente.

`minimumHealthyPercent`Rappresenta il limite inferiore al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo. È una percentuale del numero desiderato di attività per il servizio. Questo valore viene arrotondato per eccesso. Ad esempio, se la percentuale minima di integrità è `50` e il numero di attività desiderato è quattro, lo scheduler può interrompere due attività esistenti prima di iniziarne due nuove. Allo stesso modo, se la percentuale di integrità minima è 75% e il numero di processi desiderato è due, il pianificatore non può interrompere alcun processo a causa del valore risultante che è anche due.

`maximumPercent`Rappresenta il limite massimo al numero di attività che devono essere eseguite per un servizio durante una distribuzione o quando un'istanza del contenitore si sta esaurendo. È una percentuale del numero desiderato di attività per un servizio. Questo valore viene arrotondato per difetto. Ad esempio, se la percentuale massima è `200` e il numero di attività desiderato è quattro, lo scheduler può iniziare quattro nuove attività prima di interrompere quattro attività esistenti. Allo stesso modo, se la percentuale di integrità massima è `125` e il numero di processi desiderato è tre, il pianificatore non può interrompere alcun processo a causa del valore risultante che è esso stesso tre.

Quando si imposta una percentuale di integrità minima o massima, è necessario assicurarsi che il pianificatore possa arrestare o avviare almeno un processo quando viene attivata un'implementazione.

service (*nome-servizio*) non riuscito a posizionare un processo. Motivo: hai raggiunto il numero limite di attività che puoi eseguire in un dato momento

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta Quote del servizio. Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

service (*nome-servizio*) non riuscito a posizionare un processo. Motivo: errore interno.

Di seguito sono descritte le possibili cause 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.
- Stai cercando di eseguire una definizione di attività che utilizza l'architettura ARM su Fargate Spot.

service (*nome-servizio*) non riuscito a posizionare un processo. Motivo: la configurazione della CPU richiesta è al di sopra del limite.

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta Quote del servizio. Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

service (*nome-servizio*) non riuscito a posizionare un processo. Motivo: la configurazione di memoria richiesta è al di sopra del limite.

È possibile richiedere un aumento della quota per la risorsa che ha causato l'errore. Per ulteriori informazioni, consulta Quote del servizio. Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

service (*nome-servizio*) non riuscito a posizionare un processo. Motivo: hai raggiunto il numero limite di CPU che puoi eseguire in un dato momento

AWS Fargate sta passando da quote basate sul conteggio delle attività a quote basate su vCPU.

Puoi richiedere un aumento della quota per la quota basata su vCPU Fargate. Per ulteriori informazioni, consulta Quote del servizio. Per richiedere un aumento delle quote Fargate, consulta Richiesta di aumento delle quote nella Guida per l'utente di Service Quotas.

servizio (*service-name*) non è riuscito a raggiungere lo stato stazionario perché la serie di attività (*TaskSet-ID*) non è stata in grado di scalare. 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 utilizza l'`UpdateTaskProtection`API con l'`protectionEnabled`opzione impostata su `false` per annullare la protezione per queste attività.
- Aumenta il numero di attività desiderate del servizio portandolo a un numero maggiore del numero di attività protette.

Il servizio (*service-name*) non è riuscito a raggiungere lo stato stazionario. Motivo: non è stata trovata alcuna istanza di container nel provider di capacità.

Il service scheduler invia questo messaggio di evento quando non riesce a trovare le risorse disponibili per aggiungere un'altra attività. Le possibili cause sono:

Non esiste alcun provider di capacità associato al cluster

`describe-services`Da utilizzare per verificare che al cluster sia associato un provider di capacità. È possibile aggiornare la strategia del provider di capacità per il servizio.

Verifica che ci sia capacità disponibile nel provider di capacità. Nel caso del tipo di avvio EC2, assicurati che le istanze del contenitore soddisfino i requisiti di definizione delle attività.

Nessuna istanza di container trovata nel cluster

Se nessuna istanza del contenitore è registrata nel cluster in cui 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 l'attività utilizza la mappatura fissa delle porte dell'host (ad esempio, l'attività utilizza la porta 80 sull'host per un server Web), è necessario disporre di almeno un'istanza di contenitore per attività. Solo un contenitore può utilizzare una singola porta host alla volta. È consigliabile aggiungere istanze di container al tuo cluster o ridurre il numero di attività desiderate.

Troppe porte registrate

L'istanza di container più vicina per il posizionamento delle attività non può superare il limite massimo consentito di porte riservate di 100 porte host per istanza di contenitore. L'utilizzo della mappatura dinamica delle porte dell'host potrebbe risolvere il problema.

Porta già in uso

La definizione dell'attività di questa attività utilizza la stessa porta nella mappatura delle porte di un'attività già in esecuzione sull'istanza del contenitore scelta. Il messaggio dell'evento del servizio avrebbe l'ID dell'istanza di container scelto come parte del messaggio riportato di seguito.

```
The closest matching container-instance is already using a port required by your
 task.
```

Memoria insufficiente

Se nella definizione di attività sono specificati 1.000 MiB di memoria e le istanze di container nel cluster hanno ognuna 1.024 MiB di memoria, puoi eseguire una sola copia di questa attività per ogni istanza di container. Puoi provare a diminuire la memoria nella definizione di attività in modo da poter avviare più di un'attività per ogni istanza di container o avviare più istanze di container nel cluster.

> **Note**
>
> Per ottimizzare l'utilizzo delle risorse, assegnando al processo quanta più memoria possibile per un determinato tipo di istanza, consulta Riservare la memoria delle istanze del contenitore Amazon ECS Linux .

Punti di collegamento insufficienti per le interfacce di rete elastiche

Le attività che utilizzano la modalità di rete `awsvpc` ricevono ognuna una propria interfaccia di rete elastica (ENI), che è collegata all'istanza di container che la ospita. Le istanze Amazon EC2 hanno un limite al numero di ENI che possono essere collegate a esse e non ci sono istanze di 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 la sezione relativa agli Indirizzi IP per interfaccia di rete per tipo di istanza nella Guida per l'utente di Amazon EC2.
- Se hai attivato l'impostazione dell'`awsvpcTrunking`account ma non hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo l'attivazione, il limite ENI per ogni istanza di contenitore è ancora al valore predefinito. Per ulteriori informazioni, consulta la sezione relativa agli Indirizzi IP per interfaccia di rete per tipo di istanza nella Guida per l'utente di Amazon EC2.
- Se hai fornito il consenso esplicito all'impostazione dell'account `awsvpcTrunking` e hai avviato nuove istanze di container utilizzando un tipo di istanza supportato dopo il consenso esplicito, sono disponibili ENI aggiuntive. Per ulteriori informazioni, consulta Istanze supportate per interfacce di rete di container Amazon ECS potenziate.

Per ulteriori informazioni sul consenso esplicito all'impostazione dell'account `awsvpcTrunking`, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.

Puoi aggiungere istanze di container al tuo cluster per fornire più adattatori di rete disponibili.

Attributo richiesto mancante nell'istanza di container

Alcuni parametri di definizione di attività richiedono che nell'istanza di container sia installata una specifica versione dell'API remota Docker. Altri, come le opzioni relative ai driver di log, richiedono alle istanze di container di registrare tali driver con la variabile di configurazione dell'agente `ECS_AVAILABLE_LOGGING_DRIVERS`. Se la definizione dell'attività contiene un parametro che richiede uno specifico attributo di istanza del contenitore e non sono disponibili istanze di contenitore in grado di soddisfare questo requisito, l'attività non può essere inserita.

Una causa comune di questo errore è se il servizio utilizza attività che utilizzano la modalità di `awsvpc` rete e il tipo di avvio EC2 e il cluster specificato non ha un'istanza di contenitore

registrata nella stessa sottorete specificata al `awsvpcConfiguration` momento della creazione del servizio.

Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS e Configurazione dell'agente del container Amazon ECS.

service (***nome-servizio***) non riuscito a posizionare un processo. 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 Fargate o le istanze di container EC2 diventino disponibili.
- Riavvia il servizio e specifica sottoreti aggiuntive.

distribuzione del ***servizio (nome-servizio***) non riuscita: le attività non sono state avviate.

Le attività del servizio non sono state avviate.

Per informazioni su come eseguire il debug delle attività interrotte, vedere. Messaggi di errore delle attività interrotte da Amazon ECS

service (***service-name***) Timeout in attesa dell'avvio di Amazon ECS Agent. Controlla i log su /var/log/ecs/ecs-agent.log».

L'agente del container Amazon ECS sull'istanza di container con la corrispondenza maggiore per il posizionamento del processo viene disconnesso. Se riesci a connetterti all'istanza del contenitore con SSH, puoi esaminare i log dell'agente. Per ulteriori informazioni, consulta Parametri di configurazione del registro dell'agente container Amazon ECS. Devi anche verificare che l'agente sia in esecuzione sull'istanza. Se stai utilizzando l'AMI ottimizzata per Amazon ECS, puoi provare ad arrestare e riavviare l'agente con il comando seguente.

- Per l'AMI Amazon Linux 2 ottimizzata per Amazon ECS

```
sudo systemctl restart ecs
```

- Per l'AMI Amazon Linux ottimizzata per Amazon ECS

```
sudo stop ecs && sudo start ecs
```

*service (service-name) task set (taskset-ID) non è integro in target-group (targetGroup-ARN)) a causa di.*TARGET GROUP IS NOT FOUND

L'attività impostata per il servizio non supera i controlli di integrità perché il gruppo target non è stato trovato. È necessario eliminare e ricreare il servizio. Non eliminare alcun gruppo target Elastic Load Balancing a meno che il servizio Amazon ECS corrispondente non sia già stato eliminato.

*service (service-name) task set (taskset-ID) non è integro in target-group (targetGroup-ARN)) a causa di.*TARGET IS NOT FOUND

L'attività impostata per il servizio non supera i controlli di integrità perché la destinazione non è stata trovata.

Risoluzione dei problemi relativi ai servizi di bilanciamento del carico in Amazon ECS

I servizi Amazon ECS possono registrare i processi mediante un load balancer Elastic Load Balancing. Gli errori di configurazione del load balancer sono una causa comune dell'interruzione delle attività. Se le attività interrotte erano state avviate da servizi che utilizzano un load balancer, considera le seguenti possibili cause.

Il ruolo collegato al servizio Amazon ECS non esiste

> Il ruolo collegato ai servizi Amazon ECS consente ai servizi Amazon ECS di registrare le istanze di container con Elastic Load Balancing. Il ruolo collegato ai servizi deve essere creato nel tuo account. Per ulteriori informazioni, consulta Uso di ruoli collegati ai servizi per Amazon ECS.

Gruppo di sicurezza delle istanze del contenitore

> Se il tuo container è associato alla porta 80 della tua istanza di container, il gruppo di sicurezza per le istanze di container deve consentire il traffico in ingresso sulla porta 80 per garantire il superamento dei controlli dello stato del load balancer.

Il sistema di bilanciamento del carico Elastic Load Balancing non è configurato per tutte le zone di disponibilità

Il tuo load balancer deve essere configurato in modo da utilizzare tutte le zone di disponibilità all'interno di una regione o almeno tutte le zone di disponibilità in cui risiedono le tue istanze di container. Se un servizio utilizza un sistema di bilanciamento del carico e avvia un'attività su un'istanza di contenitore che risiede in una zona di disponibilità per la quale il bilanciamento del carico non è configurato per l'utilizzo, l'attività non supera mai il controllo di integrità. Ciò comporta l'interruzione dell'attività.

Il controllo dello stato del load balancer Elastic Load Balancing non è configurato correttamente

I parametri di controllo dell'integrità del load balancer possono essere troppo restrittivi o puntare a risorse inesistenti. Se un'istanza del contenitore viene ritenuta non integra, viene rimossa dal sistema di bilanciamento del carico. Assicurati di verificare che i parametri seguenti siano configurati correttamente per il load balancer del tuo servizio.

Ping Port (Porta ping)

Il valore Ping Port per il controllo dello stato di un load balancer corrisponde alla porta delle istanze di container verificate dal load balancer per determinare se sono integre. Se questa porta non è configurata correttamente, il load balancer probabilmente revocherà la registrazione dell'istanza di container da se stessa. Questa porta deve essere configurata in modo da utilizzare il valore `hostPort` per il container nella definizione di attività del servizio che stai utilizzando con il controllo dell'integrità.

Ping Path (Percorso ping)

Questo fa parte dello stato di salute del sistema di bilanciamento del carico. È un endpoint dell'applicazione in grado di restituire un codice di stato corretto (ad esempio 200) quando l'applicazione è integra. Questo valore viene spesso impostato su `index.html`, ma se il tuo servizio non risponde a tale richiesta, il controllo dell'integrità ha esito negativo. Se il tuo container non dispone di un file `index.html`, puoi impostare tale valore su `/` per indirizzare l'URL di base per l'istanza di container.

Response Timeout (Timeout di risposta)

Questo è il tempo entro il quale il tuo container deve restituire una risposta al ping del controllo dello stato. Se questo valore è inferiore al tempo necessario per una risposta, il controllo dello stato ha esito negativo.

Health Check Interval (Intervallo tra controlli dello stato)

Questo è il tempo compreso tra i ping del controllo dello stato. Più brevi sono gli intervalli tra i controlli dello stato, più rapidamente la tua istanza di container sarà in grado di raggiungere il valore Unhealthy Threshold (Soglia di mancata integrità).

Unhealthy Threshold (Soglia di mancata integrità)

Questo è il numero di volte che il controllo dello stato può avere esito negativo prima che la tua istanza di container venga considerata non integra. Se hai una soglia non corretta di 2 e un intervallo di controllo dello stato di 30 secondi, l'attività ha 60 secondi per rispondere al ping del controllo dello stato 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 nome del servizio: 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 AWS CLI o l'SDK per modificare la configurazione del bilanciamento del carico. Per informazioni su come modificare la configurazione, consulta il riferimento UpdateServiceall'API di Amazon Elastic Container Service. Se aggiorni la definizione di attività per il servizio, il nome e la porta del container specificati nella configurazione del sistema di bilanciamento del carico devono rimanere nella definizione di attività.

Hai raggiunto il limite del numero di attività che puoi eseguire contemporaneamente.

Per un nuovo account, le quote potrebbero essere inferiori a quelle di servizio. La quota di servizio per l'account può essere visualizzata nella console Service Quotas. Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

Risoluzione dei problemi relativi alla scalabilità automatica del servizio in Amazon ECS

Application Auto Scaling disattiva i processi di scalabilità mentre le implementazioni di Amazon ECS sono in corso e riprendono una volta completata l'implementazione. Tuttavia, durante l'implementazione i processi di dimensionamento orizzontale continuano a verificarsi, a meno che non siano sospesi. Per ulteriori informazioni, consulta Sospensione e ripresa del dimensionamento per Application Auto Scaling.

Risolvi gli errori di CPU o memoria non validi relativi alla definizione delle attività di Amazon ECS

Quando si registra una definizione di attività utilizzando l'API Amazon ECS o AWS CLI, se si specifica un `memory` valore `cpu` o non valido, viene restituito il seguente errore.

```
An error occurred (ClientException) when calling the RegisterTaskDefinition operation:
 Invalid 'cpu' setting for task.
```

> **Note**
>
> Quando si utilizza Terraform, potrebbe essere restituito il seguente errore.
>
> ```
> Error: ClientException: No Fargate configuration exists for given values.
> ```

Per risolvere questo problema, devi specificare un valore di CPU e memoria supportato nella definizione di attività. Il `cpu` valore può essere espresso in unità CPU o vCPU in una definizione di attività. Viene convertito in un numero intero che indica le unità CPU al momento della registrazione della definizione dell'attività. Il `memory` valore può essere espresso in MiB o GB in una definizione di attività. Viene convertito in un numero intero che indica il MiB quando viene registrata la definizione dell'attività.

Per le definizioni di processo che specificano solo `EC2` per la `requiresCompatibilities`, i valori della CPU supportati sono compresi tra `256` unità CPU (`0.25` vCPU) e `16384` unità CPU (16 vCPU). Il valore della memoria deve essere un numero intero e il limite dipende dalla quantità di memoria disponibile sull'istanza Amazon EC2 sottostante che utilizzi.

Per le definizioni delle attività che `FARGATE` specificano il `requiresCompatibilities` parametro (anche se `EC2` è specificato), devi utilizzare uno dei valori nella tabella seguente. Questi valori determinano l'intervallo di valori supportati per il parametro CPU e memoria.

Per i processi ospitati su Fargate, nella tabella seguente sono riportate le combinazioni valide di CPU e memoria. I valori di memoria nel file JSON sono specificati in MiB. Puoi convertire il valore in GB in MiB moltiplicando il valore per 1.024. Ad esempio 1 GB = 1.024 MiB.

Valore CPU	Valore memoria	Sistemi operativi supportati per AWS Fargate
256 (0,25 vCPU)	512 MiB, 1 GB, 2 GB	Linux
512 (0,5 vCPU)	1 GB, 2 GB, 3 GB, 4 GB	Linux
1024 (1 vCPU)	2 GB, 3 GB, 4 GB, 5 GB, 6 GB, 7 GB, 8 GB	Linux, Windows
2048 (2 vCPU)	Tra 4 GB e 16 GB in incrementi di 1 GB	Linux, Windows
4096 (4 vCPU)	Tra 8 GB e 30 GB in incrementi di 1 GB	Linux, Windows
8192 (8 vCPU) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 16 GB e 60 GB in incrementi di 4 GB	Linux
16384 (16vCPU) **Note** Questa opzione richiede la piattaforma Linux `1.4.0` o successiva.	Tra 32 GB e 120 GB in incrementi di 8 GB	Linux

Per le attività ospitate su Amazon EC2, i valori della CPU delle attività supportati sono compresi tra 0,25 vCPU e 192 vCPU.

> **Note**
>
> I parametri della CPU e della memoria a livello di processo vengono ignorati per i container Windows.

Visualizzazione dei log degli agenti container Amazon ECS

Amazon ECS archivia i log nella cartella `/var/log/ecs` delle tue istanze di container. Sono disponibili log dell'agente dal container Amazon ECS e dal servizio `ecs-init` che controlla lo stato (di avvio/arresto) dell'agente sull'istanza di container. Puoi visualizzare tali file di log connettendoti a un'istanza di container tramite SSH.

> **Note**
>
> Se hai dubbi su come raccogliere tutti i log nelle tue istanze di container, puoi utilizzare il programma di raccolta log di Amazon ECS. Per ulteriori informazioni, consulta Raccolta dei log dei container con Amazon ECS logs collector .

Sistema operativo Linux

Il processo `ecs-init` archivia i log in `/var/log/ecs/ecs-init.log`.

Il `ecs-init.log` file contiene informazioni sulla gestione, la configurazione e il bootstrap del ciclo di vita degli agenti container.

Puoi utilizzare il comando seguente per visualizzare i file di log.

```
cat /var/log/ecs/ecs-init.log
```

Output:

```
2018-02-16T18:13:54Z [INFO] pre-start
2018-02-16T18:13:56Z [INFO] start
2018-02-16T18:13:56Z [INFO] No existing agent container to remove.
2018-02-16T18:13:56Z [INFO] Starting Amazon Elastic Container Service Agent
```

Sistema operativo Windows

Puoi usare Amazon ECS logs collector per Windows. Per ulteriori informazioni, consulta Amazon ECS Logs Collector per Windows su Github.

1. Connettiti alla tua istanza.
2. Apri PowerShell ed esegui i seguenti comandi con privilegi amministrativi. I comandi scaricano lo script e raccolgono i log.

```
Invoke-WebRequest -OutFile ecs-logs-collector.ps1 https://
raw.githubusercontent.com/awslabs/aws-ecs-logs-collector-for-windows/master/ecs-
logs-collector.ps1
.\ecs-logs-collector.ps1
```

Puoi attivare la registrazione di debug per l'agente Amazon ECS e il daemon Docker. Questa opzione consente allo script di raccogliere i log prima di attivare la modalità di debug. Lo script riavvia il daemon Docker e l'agente Amazon ECS, quindi termina tutti i contenitori in esecuzione sull'istanza. Prima di eseguire il comando seguente, svuota l'istanza del contenitore e sposta tutte le attività importanti su altre istanze del contenitore.

Eseguite il comando seguente per attivare la registrazione.

```
.\ecs-logs-collector.ps1 -RunMode debug
```

Raccolta dei log dei container con Amazon ECS logs collector

Se hai dubbi su come raccogliere tutti i vari log nelle tue istanze di container, puoi utilizzare il programma di raccolta log di Amazon ECS. È disponibile sia per Linux che GitHub per Windows. Lo script raccoglie i log generali del sistema operativo, nonché i log degli agenti container Docker e Amazon ECS, che possono essere utili per la risoluzione dei casi. AWS Support Quindi comprime e archivia le informazioni raccolte in un singolo file che può essere facilmente condiviso per scopi di diagnostica. Supporta inoltre l'abilitazione della modalità di debug per il daemon Docker e l'agente del container di Amazon ECS nelle varianti di Amazon Linux, ad esempio l'AMI ottimizzata per Amazon ECS. Al momento, il programma di raccolta log di Amazon ECS supporta i seguenti sistemi operativi:

- Amazon Linux
- Red Hat Enterprise Linux 7

- Debian 8
- Ubuntu 14.04
- Ubuntu 16.04
- Ubuntu 18.04
- Windows Server 2016

> **Note**
>
> Il codice sorgente per il raccoglitore di log di Amazon ECS è disponibile sia GitHub per Linux che per Windows. Consigliamo di inviare le richieste pull per le modifiche che desideri siano incluse. Tuttavia, Amazon Web Services al momento non supporta l'esecuzione di copie modificate di questo software.

Come scaricare ed eseguire il programma di raccolta log di Amazon ECS per Linux

1. Connettiti alla tua istanza di container.
2. Scarica lo script del programma di raccolta di log di Amazon ECS.

```
curl -O https://raw.githubusercontent.com/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 demone Docker e l'agente contenitore Amazon ECS, aggiungi l'`--mode=enable-debug`opzione al comando seguente. Ciò potrebbe riavviare il demone Docker, che uccide tutti i contenitori in esecuzione sull'istanza. Prendi in considerazione di svuotare l'istanza di container e di spostare eventuali attività importanti su altre istanze di container prima di abilitare la modalità di debug. Per ulteriori informazioni, consulta Drenaggio delle istanze di container Amazon ECS.

```
[ec2-user ~]$ sudo bash ./ecs-logs-collector.sh
```

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 un aiuto diagnostico. AWS Support

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 Connessione all'istanza Windows nella Guida per l'utente di Amazon EC2.
2. Scarica lo script di raccolta dei log di Amazon ECS utilizzando. PowerShell

```
Invoke-WebRequest -OutFile ecs-logs-collector.ps1 https://
raw.githubusercontent.com/awslabs/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 demone Docker e l'agente contenitore Amazon ECS, aggiungi l'`-RunMode debug`opzione al comando seguente. Questo comporta il riavvio del daemon Docker, con conseguente interruzione di tutti i container in esecuzione nell'istanza. Prendi in considerazione di svuotare l'istanza di container e di spostare eventuali attività importanti su altre istanze di container prima di abilitare la modalità di debug. Per ulteriori informazioni, consulta Drenaggio delle istanze di container Amazon ECS.

```
.\ecs-logs-collector.ps1
```

Dopo aver eseguito lo script, puoi esaminare i log raccolti nella cartella `collect` creata dallo script. Il `collect.tgz` file è un archivio compresso di tutti i registri, che puoi condividere con Support per ricevere AWS assistenza diagnostica.

Recupera i dettagli diagnostici di Amazon ECS con l'introspezione degli agenti

L'API di introspezione dell'agente Amazon ECS fornisce informazioni sullo stato generale dell'agente Amazon ECS e delle istanze del contenitore.

Puoi utilizzare l'API di introspezione dell'agente per ottenere l'ID Docker per un contenitore nell'ambito della tua attività. Puoi utilizzare l'API di introspezione dell'agente connettendoti a un'istanza di container tramite SSH.

> **Important**
>
> L'istanza di container deve disporre di un ruolo IAM che consenta l'accesso ad Amazon ECS per raggiungere l'API di introspezione. Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container Amazon ECS.

L'esempio seguente mostra due attività, una attualmente in esecuzione e una che è stata interrotta.

> **Note**
>
> Il comando seguente viene reindirizzato tramite il python -mjson.tool per una maggiore leggibilità.

```
curl http://localhost:51678/v1/tasks | python -mjson.tool
```

Output:

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1095  100  1095    0     0   117k      0 --:--:-- --:--:-- --:--:--  133k
{
    "Tasks": [
        {
            "Arn": "arn:aws:ecs:us-west-2:aws_account_id:task/090eff9b-1ce3-4db6-848a-
a8d14064fd24",
            "Containers": [
                {
```

```
                    "DockerId":
 "189a8ff4b5f04affe40e5160a5ffadca395136eb5faf4950c57963c06f82c76d",
                    "DockerName": "ecs-console-sample-app-static-6-simple-
app-86caf9bcabe3e9c61600",
                    "Name": "simple-app"
                },
                {
                    "DockerId":
 "f7f1f8a7a245c5da83aa92729bd28c6bcb004d1f6a35409e4207e1d34030e966",
                    "DockerName": "ecs-console-sample-app-static-6-busybox-
ce83ce978a87a890ab01",
                    "Name": "busybox"
                }
            ],
            "Family": "console-sample-app-static",
            "KnownStatus": "STOPPED",
            "Version": "6"
        },
        {
            "Arn": "arn:aws:ecs:us-west-2:aws_account_id:task/1810e302-eaea-4da9-
a638-097bea534740",
            "Containers": [
                {
                    "DockerId":
 "dc7240fe892ab233dbbcee5044d95e1456c120dba9a6b56ec513da45c38e3aeb",
                    "DockerName": "ecs-console-sample-app-static-6-simple-app-
f0e5859699a7aecfb101",
                    "Name": "simple-app"
                },
                {
                    "DockerId":
 "096d685fb85a1ff3e021c8254672ab8497e3c13986b9cf005cbae9460b7b901e",
                    "DockerName": "ecs-console-sample-app-static-6-
busybox-92e4b8d0ecd0cce69a01",
                    "Name": "busybox"
                }
            ],
            "DesiredStatus": "RUNNING",
            "Family": "console-sample-app-static",
            "KnownStatus": "RUNNING",
            "Version": "6"
        }
    ]
```

```
}
```

Nell'esempio precedente, l'operazione interrotta (*090eff9b-1ce3-4db6-848a-a8d14064fd24*) ha due contenitori. Puoi utilizzare docker inspect ***container-ID*** per visualizzare informazioni dettagliate su ogni container. Per ulteriori informazioni, consulta Introspezione dei container Amazon ECS.

Diagnostica Docker in Amazon ECS

Docker fornisce diversi strumenti di diagnostica che ti aiutano a risolvere i problemi relativi a container e attività. Per ulteriori informazioni su tutte le utility a riga di comando Docker disponibili, consulta l'argomento relativo alla riga di comando Docker nella documentazione Docker. Puoi accedere alle utility a riga di comando Docker connettendoti a un'istanza di container tramite SSH.

Anche i codici di uscita riportati dai container Docker possono fornire informazioni di diagnostica (ad esempio, il codice di uscita 137 indica che il container ha ricevuto un segnale `SIGKILL`). Per ulteriori informazioni, consulta la sezione relativa allo stato di uscita nella documentazione Docker.

Elenca i contenitori Docker in Amazon ECS

Puoi utilizzare il comando docker ps sulla tua istanza di container per visualizzare l'elenco dei container in esecuzione. Nell'esempio seguente, è in esecuzione solo l'agente container Amazon ECS. Per ulteriori informazioni, consulta la sezione relativa al comando docker ps nella documentazione Docker.

```
docker ps
```

Output:

```
CONTAINER ID        IMAGE                            COMMAND             CREATED
      STATUS              PORTS                          NAMES
cee0d6986de0        amazon/amazon-ecs-agent:latest   "/agent"            22 hours ago
      Up 22 hours         127.0.0.1:51678->51678/tcp   ecs-agent
```

Puoi utilizzare il comando docker ps -a per visualizzare tutti i container (anche quelli sospesi o interrotti). Questo comando è utile per visualizzare l'elenco dei container che si arrestano inaspettatamente. Nell'esempio seguente, il container `f7f1f8a7a245` è stato chiuso 9 secondi fa, pertanto non compare nell'output del comando docker ps senza il flag `-a`.

```
docker ps -a
```

Output:

```
CONTAINER ID        IMAGE                                                COMMAND
 CREATED             STATUS                          PORTS                          NAMES
db4d48e411b1        amazon/ecs-emptyvolume-base:autogenerated   "not-applicable"
  19 seconds ago                                                                     ecs-
console-sample-app-static-6-internalecs-emptyvolume-source-c09288a6b0cba8a53700
f7f1f8a7a245        busybox:buildroot-2014.02                    "\"sh -c '/bin/sh -c
  22 hours ago        Exited (137) 9 seconds ago                                     ecs-
console-sample-app-static-6-busybox-ce83ce978a87a890ab01
189a8ff4b5f0        httpd:2                                      "httpd-foreground"
  22 hours ago        Exited (137) 40 seconds ago                                    ecs-
console-sample-app-static-6-simple-app-86caf9bcabe3e9c61600
0c7dca9321e3        amazon/ecs-emptyvolume-base:autogenerated   "not-applicable"
  22 hours ago                                                                       ecs-
console-sample-app-static-6-internalecs-emptyvolume-source-90fefaa68498a8a80700
cee0d6986de0        amazon/amazon-ecs-agent:latest               "/agent"
  22 hours ago        Up 22 hours                 127.0.0.1:51678->51678/tcp   ecs-
agent
```

Visualizza i log Docker in Amazon ECS

Puoi visualizzare i flussi STDOUT e STDERR relativi a un container mediante il comando docker logs. In questo esempio, i log sono visualizzati per il container *dc7240fe892a* e reindirizzati tramite il comando head 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 hai configurato le tue attività per utilizzare il driver di `awslogs` registro, i log del contenitore sono disponibili in Logs. CloudWatch Per ulteriori informazioni, consulta Invia i log di Amazon ECS a CloudWatch .

```
docker logs dc7240fe892a | head
```

Output:

```
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
 using 172.17.0.11. Set the 'ServerName' directive globally to suppress this message
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
 using 172.17.0.11. Set the 'ServerName' directive globally to suppress this message
[Thu Apr 23 19:48:36.956682 2015] [mpm_event:notice] [pid 1:tid 140327115417472]
 AH00489: Apache/2.4.12 (Unix) configured -- resuming normal operations
[Thu Apr 23 19:48:36.956827 2015] [core:notice] [pid 1:tid 140327115417472] AH00094:
 Command line: 'httpd -D FOREGROUND'
10.0.1.86 - - [23/Apr/2015:19:48:59 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:48:59 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:49:28 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:49:29 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:49:50 +0000] "-" 408 -
10.0.0.154 - - [23/Apr/2015:19:49:50 +0000] "-" 408 -
10.0.1.86 - - [23/Apr/2015:19:49:58 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:49:59 +0000] "GET / HTTP/1.1" 200 348
10.0.1.86 - - [23/Apr/2015:19:50:28 +0000] "GET / HTTP/1.1" 200 348
10.0.0.154 - - [23/Apr/2015:19:50:29 +0000] "GET / HTTP/1.1" 200 348
time="2015-04-23T20:11:20Z" level="fatal" msg="write /dev/stdout: broken pipe"
```

Ispeziona i contenitori Docker in Amazon ECS

Se disponi dell'ID Docker di un container, puoi ispezionarlo mediante il comando docker inspect. L'ispezione dei container fornisce le informazioni più dettagliate sull'ambiente in cui è stato avviato un container. Per ulteriori informazioni, consulta la sezione relativa al comando docker inspect nella documentazione Docker.

```
docker inspect dc7240fe892a
```

Output:

```
[{
    "AppArmorProfile": "",
    "Args": [],
    "Config": {
        "AttachStderr": false,
        "AttachStdin": false,
        "AttachStdout": false,
        "Cmd": [
            "httpd-foreground"
        ],
```

```
        "CpuShares": 10,
        "Cpuset": "",
        "Domainname": "",
        "Entrypoint": null,
        "Env": [
            "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/
local/apache2/bin",
            "HTTPD_PREFIX=/usr/local/apache2",
            "HTTPD_VERSION=2.4.12",
            "HTTPD_BZ2_URL=https://www.apache.org/dist/httpd/httpd-2.4.12.tar.bz2"
        ],
        "ExposedPorts": {
            "80/tcp": {}
        },
        "Hostname": "dc7240fe892a",
...
```

Configurazione dell'output verboso dal demone Docker in Amazon ECS

Se hai problemi con i contenitori o le immagini Docker, puoi attivare la modalità di debug sul tuo demone Docker. L'uso del debug fornisce un output più dettagliato dal demone. È possibile utilizzarlo per recuperare i messaggi di errore inviati dai registri dei container, come Amazon ECR.

> **Important**
>
> Questa procedura è scritta per l'AMI Amazon Linux ottimizzata per Amazon ECS. Per altri sistemi operativi, consulta Enable debugging and Control and configure with nella documentazione Docker. Dockersystemd

Per utilizzare la modalità di debug del demone Docker sull'AMI Amazon Linux ottimizzata per Amazon ECS

1. Connettiti alla tua istanza di container.
2. Aprire il file delle opzioni Docker con un editor di testo, ad esempio vi. Per l'AMI Amazon Linux ottimizzata per Amazon ECS, il file delle opzioni Docker si trova in `/etc/sysconfig/docker`.
3. Trova l'istruzione delle opzioni Docker e aggiungi l'opzione `-D` alla stringa, tra virgolette.

> **Note**
>
> Se l'istruzione delle opzioni Docker inizia con #, rimuovi questo carattere dall'istruzione e abilita le opzioni.

Per l'AMI Amazon Linux ottimizzata per Amazon ECS, l'istruzione delle opzioni Docker si chiama `OPTIONS`. Ad esempio:

```
# Additional startup options for the Docker daemon, for example:
# OPTIONS="--ip-forward=true --iptables=true"
# By default we limit the number of open files per container
OPTIONS="-D --default-ulimit nofile=1024:4096"
```

4. Salva il file ed esci dall'editor di testo.
5. Riavvia il daemon Docker.

```
sudo service docker restart
```

L'output è il seguente:

```
Stopping docker:                                           [  OK  ]
Starting docker: .                                   [  OK  ]
```

6. Riavvia l'agente Amazon ECS.

```
sudo service ecs restart
```

Il tuo log di Docker dovrebbe ora mostrare un output più dettagliato.

```
time="2015-12-30T21:48:21.907640838Z" level=debug msg="Unexpected response from
 server: \"{\\\"errors\\\":[{\\\"code\\\":\\\"DENIED\\\",\\\"message\\\":\\\"User:
 arn:aws:sts::1111:assumed-role/ecrReadOnly/i-abcdefg is not authorized to perform:
 ecr:InitiateLayerUpload on resource: arn:aws:ecr:us-east-1:1111:repository/nginx_test
\\\"}]}\\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 al Docker in Amazon **`API error (500): devmapper`** ECS

Il seguente errore Docker indica che lo spazio di storage del thin pool della tua istanza di container è pieno e che il daemon Docker non può creare nuovi container:

```
CannotCreateContainerError: API error (500): devmapper: Thin Pool has 4350 free data
 blocks which is less than minimum required 4454 free data blocks. Create more free
 space in thin pool or use dm.min_free_space option to change behavior
```

di default, le AMI Amazon Linux ottimizzate per Amazon ECS dalla versione `2015.09.d` e successiva vengono avviate con un volume di 8 GiB per il sistema operativo collegato a `/dev/xvda` e sono montate come root 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 AMI Linux ottimizzate per Amazon ECS.

Se lo storage dell'istanza di container si sta riempiendo troppo rapidamente, puoi eseguire alcune operazioni per ridurre questo effetto:

- Per visualizzare le informazioni di thin poll, esegui il comando seguente nella tua istanza di container:

  ```
  docker info
  ```

- (Agente container Amazon ECS 1.8.0 e versioni successive) Puoi ridurre la quantità di tempo in cui i container fermi o usciti rimangono sulle tue istanze di container. La variabile di configurazione dell'agente `ECS_ENGINE_TASK_CLEANUP_WAIT_DURATION` imposta il tempo di attesa dall'interruzione di un'attività fino alla rimozione del container Docker (di default, questo valore è 3 ore). Questo rimuove i dati del container di Docker. Se questo valore è impostato su un valore troppo basso, potresti non essere in grado di ispezionare i contenitori fermi o visualizzare i log prima che vengano rimossi. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

- Puoi rimuovere i contenitori non in esecuzione e le immagini non utilizzate dalle istanze dei contenitori. Puoi utilizzare i seguenti comandi di esempio per rimuovere manualmente i container interrotti e le immagini inutilizzate. I container eliminati non possono essere esaminati in un secondo momento e le immagini eliminate devono essere recuperate nuovamente prima di poter avviare i nuovi container da esse.

  Per rimuovere i container non in esecuzione, esegui il comando riportato nell'istanza di container:

  ```
  docker rm $(docker ps -aq)
  ```

  Per rimuovere le immagini inutilizzate, emetti il comando seguente nell'istanza di container:

  ```
  docker rmi $(docker images -q)
  ```

- Puoi rimuovere i blocchi di dati non utilizzati all'interno dei contenitori. Puoi utilizzare il seguente comando per eseguire fstrim in qualsiasi container in esecuzione e ignorare tutti i blocchi di dati inutilizzati dal file system del container.

  ```
  sudo sh -c "docker ps -q | xargs docker inspect --format='{{ .State.Pid }}' | xargs -IZ fstrim /proc/Z/root/"
  ```

Risolvi i problemi di Amazon ECS Exec

Di seguito sono riportate note sulla risoluzione dei problemi che consentono di diagnosticare il motivo per cui potrebbe essere visualizzato un errore durante l'utilizzo di ECS Exec.

Verifica utilizzando Exec Checker

Lo script ECS Exec Checker fornisce un modo per verificare e convalidare che il cluster e l'attività Amazon ECS soddisfino i prerequisiti per l'utilizzo della funzionalità ECS Exec. Lo script ECS Exec Checker verifica che l' AWS CLI ambiente e il cluster e le attività siano pronte per ECS Exec, chiamando diverse API per tuo conto. Lo strumento richiede la versione più recente di e che sia disponibile. AWS CLI `jq` Per ulteriori informazioni, consulta ECS Exec Checker on. GitHub

Errore durante la chiamata a **`execute-command`**

Di seguito sono riportate le cause possibili se si verifica un errore `The execute command failed`.

- Il processo non dispone delle autorizzazioni necessarie. Verifica che nella definizione di attività utilizzata per avviare il processo sia definito un ruolo IAM del processo e che il ruolo disponga delle autorizzazioni richieste. Per ulteriori informazioni, consulta Autorizzazioni ECS Exec.
- L'agente SSM non è installato o non è in esecuzione.
- Esiste un'interfaccia Amazon VPC endpoint per Amazon ECS, ma non ce n'è una per Systems Manager Session Manager.

Risolvi i problemi relativi ad Amazon ECS Anywhere

Amazon ECS Anywhere fornisce supporto per la registrazione di un'istanza esterna come un server locale o una macchina virtuale (VM) nel tuo cluster Amazon ECS. Di seguito sono riportati i problemi più comuni riscontrati e i suggerimenti generali per la risoluzione dei problemi.

Argomenti

Problemi di registrazione delle istanze esterne

Durante la registrazione di un'istanza esterna nel tuo cluster Amazon ECS, è necessario soddisfare i seguenti requisiti:

- È necessario AWS Systems Manager recuperare un'attivazione, che consiste in un ID di attivazione e un codice di attivazione. Si utilizza per registrare l'istanza esterna come istanza gestita da Systems Manager. Quando viene richiesta l'attivazione di Systems Manager, specificare un limite di registrazione e una data di scadenza. Il limite di registrazione specifica il numero massimo di istanze che possono essere registrate mediante l'attivazione. Il valore predefinito per il limite di registrazione è 1 instance. La data di scadenza specifica la data di scadenza dell'attivazione. Il valore di default è 24 ore. Se l'attivazione di Systems Manager utilizzata per registrare l'istanza esterna non è valida, richiederne una nuova. Per ulteriori informazioni, consulta Registrazione di un'istanza esterna in un cluster Amazon ECS.
- Una policy IAM viene utilizzata per fornire all'istanza esterna le autorizzazioni necessarie per comunicare con le operazioni AWS API. Se questa policy gestita non viene creato correttamente e

non contiene le autorizzazioni richieste, la registrazione dell'istanza esterna non riesce. Per ulteriori informazioni, consulta Ruolo IAM di Amazon ECS Anywhere.

- Amazon ECS fornisce uno script di installazione che installa Docker, l'agente del container Amazon ECS e Systems Manager Agent nell'istanza esterna. Se lo script di installazione non riesce, è probabile che non possa essere eseguito nuovamente sulla stessa istanza senza che si verifichi un errore. In tal caso, segui la procedura di pulizia per eliminare AWS le risorse dall'istanza in modo da poter eseguire nuovamente lo script di installazione. Per ulteriori informazioni, consulta Annullamento della registrazione di un'istanza esterna Amazon ECS.

  > **Note**
  >
  > Tieni presente che, se lo script di installazione ha richiesto e utilizzato correttamente l'attivazione di Systems Manager, l'esecuzione dello script di installazione una seconda volta utilizza nuovamente l'attivazione di Systems Manager. Ciò potrebbe a sua volta causare il raggiungimento del limite di registrazione per tale attivazione. Se questo limite viene raggiunto, sarà necessario creare una nuova attivazione.

- Quando si esegue lo script di installazione su un'istanza esterna per carichi di lavoro GPU, se il driver NVIDIA non viene rilevato o configurato correttamente, si verificherà un errore. Lo script di installazione utilizza il comando `nvidia-smi` per confermare l'esistenza del driver NVIDIA.

Problemi di rete delle istanze esterne

Per comunicare eventuali modifiche, l'istanza esterna richiede una connessione di rete a AWS. Se l'istanza esterna perde la connessione di rete a AWS, le attività in esecuzione sulle istanze continuano comunque a essere eseguite a meno che non vengano interrotte manualmente. Dopo il ripristino della AWS connessione a, le AWS credenziali utilizzate dall'agente contenitore Amazon ECS e dall'agente Systems Manager sull'istanza esterna si rinnovano automaticamente. Per ulteriori informazioni sui AWS domini utilizzati per la comunicazione tra l'istanza esterna e, consulta. AWSRete

Problemi durante l'esecuzione di processi sull'istanza esterna

Se i processi o i container non vengono eseguiti nell'istanza esterna, le cause più comuni sono relative alla rete o alle autorizzazioni. Se i tuoi container stanno estraendo le loro immagini da Amazon ECR o sono configurati per inviare i log dei container a CloudWatch Logs, la definizione dell'attività deve specificare un ruolo IAM valido per l'esecuzione delle attività. Senza un ruolo IAM di

esecuzione processo valido, i container non saranno avviati. Per ulteriori informazioni sui problemi della rete, consulta Problemi di rete delle istanze esterne.

> **Important**
>
> Amazon ECS fornisce lo strumento di raccolta dei log di Amazon ECS. È possibile utilizzarlo per raccogliere i log dalle istanze esterne a scopo di risoluzione dei problemi. Per ulteriori informazioni, consulta Raccolta dei log dei container con Amazon ECS logs collector .

AWS Fargate limitazione delle quote

AWS Fargate limita le attività di Amazon ECS e i tassi di lancio dei pod Amazon EKS a quote (precedentemente denominate limiti) utilizzando un algoritmo token bucket per ogni AWS account in base alla regione. Con questo algoritmo, il tuo account ha un bucket che contiene un numero specifico di token. Il numero di token nel bucket rappresenta la tua quota di velocità in un dato secondo. Ogni account cliente ha un bucket di token per le attività e per i pod che si esaurisce in base al numero di attività e pod avviati dall'account cliente. Questo bucket di token prevede un massimo, che consente di effettuare periodicamente un numero maggiore di richieste, e una velocità di ricarica, che consente di sostenere una velocità costante di richieste per tutto il tempo necessario.

Ad esempio, la dimensione del bucket di token di attività e pod per un account cliente Fargate è di 100 token e la velocità di ricarica è di 20 token al secondo. Pertanto, puoi avviare immediatamente fino a 100 attività Amazon ECS e pod Amazon EKS per ogni account cliente, con una velocità di avvio sostenuta di 20 attività Amazon ECS e pod Amazon EKS al secondo.

Azioni	Capacità massima del bucket (o velocità di espansione)	Velocità di ricarica del bucket (o velocità sostenuta)
Quota di velocità delle risorse di Fargate per le attività Amazon ECS on demand e i pod Amazon EKS[1]	100	20
Quota di velocità delle risorse di Fargate per le attività Spot Amazon ECS	100	20

[1]Gli 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'**RunTask**API in Fargate

Inoltre, Fargate limita la velocità delle richieste durante l'avvio di attività tramite l'API `RunTask` di Amazon ECS, che utilizza una quota separata. Fargate limita le richieste `RunTask` API Amazon ECS per ogni AWS account in base alla regione. Ogni richiesta effettuata rimuove un token dal bucket. Ciò consente di aiutare le prestazioni del servizio e garantire un utilizzo equo a tutti i clienti di Fargate. Le chiamate API sono soggette alle quote di richieste, indipendentemente dal fatto che abbiano origine dalla console Amazon Elastic Container Service, da uno strumento a riga di comando o da un'applicazione di terze parti. La quota di velocità per le chiamate verso l'API `RunTask` di Amazon ECS è di 20 chiamate al secondo (di espansione e sostenuta). Tuttavia, ogni chiamata a questa API può avviare fino a 10 attività. Ciò significa che è possibile avviare 100 attività in un secondo effettuando 10 chiamate a questa API, richiedendo l'avvio di 10 attività in ogni chiamata. Allo stesso modo, è possibile effettuare 20 chiamate a questa API richiedendo l'avvio di 5 attività in ogni chiamata. Per ulteriori informazioni sulla limitazione delle API per l'API Amazon ECS, consulta la sezione `RunTask` Limitazione delle richieste API nell'Amazon ECS API Reference.

Nella pratica, le velocità di avvio delle attività e dei pod dipendono anche da altri fattori, come le immagini dei container da scaricare e decomprimere, i controlli dell'integrità e altre integrazioni abilitate, come la registrazione di attività o di pod con un load balancer. I clienti riscontrano variazioni nei tassi di avvio di task e pod rispetto alle quote rappresentate in precedenza, in base alle funzionalità abilitate dai clienti.

Adeguamento delle quote tariffarie a Fargate

Puoi richiedere un aumento delle quote di limitazione di velocità di Fargate per il tuo account AWS . Per richiedere un adeguamento delle quote, contatta il centro AWS Support.

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 di limitazione si verifica in genere quando si chiamano le API di Amazon ECS durante l'esecuzione di attività o la creazione di servizi. Per ulteriori informazioni sulla limitazione coinvolta e sui relativi limiti di accelerazione, consulta Request throttling for the Amazon ECS API.

Quando l'applicazione avvia richieste API, ad esempio utilizzando l'SDK AWS CLI o un AWS SDK, puoi porre rimedio alla limitazione delle API. Puoi farlo progettando l'applicazione in modo da gestire gli errori o implementando una strategia esponenziale di backoff e jitter con logica di ripetizione per le chiamate API. Per ulteriori informazioni, consulta Timeouts, retries e backoff with jitter.

Se utilizzi un AWS SDK, la logica di riprova automatica è già integrata e configurabile.

Limitazione asincrona in Amazon ECS

La limitazione asincrona si verifica a causa di flussi di lavoro asincroni in cui Amazon ECS o AWS CloudFormation potrebbe chiamare API per tuo conto per fornire risorse. È importante sapere quali AWS API vengono richiamate da Amazon ECS per tuo conto. Ad esempio, l'`CreateNetworkInterface`API viene richiamata per attività che utilizzano la modalità di `awsvpc` rete e l'`DescribeTargetHealth`API viene richiamata quando si eseguono controlli di integrità per attività registrate su un sistema di bilanciamento del carico.

Quando i carichi di lavoro raggiungono una scala considerevole, queste operazioni API potrebbero essere limitate. Cioè, potrebbero essere sufficientemente limitati da violare i limiti imposti da Amazon ECS o da Servizio AWS quello che viene chiamato. Ad esempio, se distribuisci centinaia di servizi, ciascuno con centinaia di attività contemporaneamente che utilizzano la modalità di `awsvpc` rete, Amazon ECS richiama le operazioni API di Amazon EC2 come le operazioni dell'API `CreateNetworkInterface` Elastic Load Balancing come o `RegisterTarget` per `DescribeTargetHealth` registrare rispettivamente l'interfaccia di rete elastica e 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 di evento del servizio.

```
{
   "userIdentity":{
      "arn":"arn:aws:sts::111122223333:assumed-role/AWSServiceRoleForECS/ecs-service-
scheduler",
      "eventTime":"2022-03-21T08:11:24Z",
```

```
      "eventSource":"elasticloadbalancing.amazonaws.com",
      "eventName":" DescribeTargetHealth ",
      "awsRegion":"us-east-1",
      "sourceIPAddress":"ecs.amazonaws.com",
      "userAgent":"ecs.amazonaws.com",
      "errorCode":"ThrottlingException",
      "errorMessage":"Rate exceeded",
      "eventID":"0aeb38fc-229b-4912-8b0d-2e8315193e9c"
   }
}
```

Quando queste chiamate API condividono i limiti con altro traffico API nel tuo account, potrebbero essere difficili da monitorare anche se vengono emesse come eventi di servizio.

Monitora la limitazione

È importante identificare quali richieste API sono limitate e chi le emette. Puoi utilizzare AWS CloudTrail Which per monitorare la limitazione e integrarsi con Amazon CloudWatch Athena e Amazon. EventBridge Puoi configurare l'invio di eventi specifici CloudTrail ai registri. CloudWatch CloudWatch Logs, log insights, analizza e analizza gli eventi. Questo identifica i dettagli negli eventi di limitazione, come l'utente o il ruolo IAM che ha effettuato la chiamata e il numero di chiamate API effettuate. Per ulteriori informazioni, consulta Monitoraggio dei file di CloudTrail registro con Logs. CloudWatch

Per ulteriori informazioni su CloudWatch Logs Insights e istruzioni su come interrogare i file di registro, vedi Analisi dei dati di log con CloudWatch Logs Insights.

Con Amazon Athena, puoi creare query e analizzare dati utilizzando SQL standard. Ad esempio, puoi creare una tabella Athena per analizzare CloudTrail gli eventi. Per ulteriori informazioni, vedere Utilizzo della CloudTrail console per creare una tabella Athena per i CloudTrail log.

Dopo aver creato una tabella Athena, è possibile utilizzare semplici query SQL come la seguente per analizzare gli errori. `ThrottlingException`

```
select eventname, errorcode,eventsource,awsregion, useragent,COUNT(*) count
FROM cloudtrail-table-name
where errorcode = 'ThrottlingException'
AND eventtime between '2022-01-14T03:00:08Z' and '2022-01-23T07:15:08Z'
group by errorcode, awsregion, eventsource, username, eventname
order by count desc;
```

Amazon ECS invia anche notifiche di eventi ad Amazon. EventBridge Esistono eventi di modifica dello stato delle risorse ed eventi di azione del servizio. Includono eventi di limitazione delle API come eECS_OPERATION_THROTTLED. SERVICE_DISCOVERY_OPERATION_THROTTLED Per ulteriori informazioni, consulta Eventi di azione del servizio Amazon ECS.

Questi eventi possono essere utilizzati da un servizio, ad esempio AWS Lambda per eseguire azioni in risposta. Per ulteriori informazioni, consulta Gestione degli eventi Amazon ECS.

Se esegui attività autonome, alcune operazioni API, ad esempio, `RunTask` sono asincrone e le operazioni di riprova non vengono eseguite automaticamente. In questi casi, puoi utilizzare servizi come AWS Step Functions with EventBridge integration per riprovare le operazioni limitate o non riuscite. Per ulteriori informazioni, consulta Manage a container task (Amazon ECS, Amazon SNS).

CloudWatch Da utilizzare per monitorare la limitazione

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

CloudWatch offre anche il rilevamento delle anomalie. Questa funzionalità utilizza l'apprendimento automatico per analizzare e stabilire linee di base in base al comportamento particolare della metrica su cui l'hai abilitata. In caso di attività insolite dell'API, puoi utilizzare questa funzionalità insieme agli allarmi. CloudWatch Per ulteriori informazioni, consulta Utilizzo del rilevamento CloudWatch delle anomalie.

Monitorando in modo proattivo gli errori di limitazione, potete contattarci AWS Support per aumentare i limiti di limitazione pertinenti e ricevere anche indicazioni per le vostre esigenze applicative specifiche.

Motivi di errore dell'API Amazon ECS

Quando un'azione API che hai attivato tramite l'API, la console o la console di Amazon ECS AWS CLI esce con un messaggio di `failures` errore, quanto segue può aiutarti a risolvere la causa. L'errore restituisce un motivo e l'Amazon Resource Name (ARN) della risorsa associata all'errore.

Molte risorse sono specifiche della regione, quindi accertati che la console sia impostata sulla regione corretta per le tue risorse. Quando usi il AWS CLI, assicurati che i AWS CLI comandi vengano inviati alla regione corretta con il `--region` *`region`* parametro.

Per ulteriori informazioni sulla struttura del tipo di dati `Failure`, consulta Errore nella Documentazione di riferimento dell'API di Amazon Elastic Container Service.

Di seguito sono riportati alcuni esempi di messaggi di errore che potreste ricevere durante l'esecuzione di comandi API.

Azione API	Motivo dell'errore o dell'interruzione	Causa
`DescribeClusters`	`MISSING`	Il cluster specificato non è stato trovato. Verifica l'ortografia del nome del cluster.
`DescribeInstances`	`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.
`DescribeServices`	`MISSING`	Il servizio specificato non è stato trovato. Verifica che sia specificato il cluster o la regione corretta e che l'ARN del servizio o il nome sia valido.
`DescribeTasks`	`MISSING`	Il processo specificato non è stato trovato. Verifica che sia specificato il cluster o la regione corretta e che l'ARN del processo o il nome sia valido.
`DescribeTasks`	`TaskFailedToStart: RESOURCE:*`	Per gli errori `RESOURCE:CPU` indica che il numero di CPU richiesto dall'attività non è

Azione API	Motivo dell'errore o dell'interruzione	Causa
		disponibile sulle istanze di container. Ciò si verifica in genere quando il requisito dell'unità CPU nella definizione dell'attività è maggiore della dimensione della CPU delle istanze Amazon EC2 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 sulle istanze Amazon EC2 definita 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
	`TaskFailedToStart: 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).
	`TaskFailedToStart: MemberOf placement constraint unsatisfied`	Non esiste un'istanza di contenitore che soddisfi i vincoli di posizionamento definiti nella definizione dell'attività.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	`TaskFailedToStart: 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 `ecs.capability.task-eni` . Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS e Configurazione dell'agente del container Amazon ECS.
	`TaskFailedToStart: NO ACTIVE INSTANCES`	Non ci sono istanze attive nel provider di capacità. Per informazioni sulla gestione dei gruppi con dimensionamento automatico, consulta Gruppi con dimensionamento automatico nella Guida per l'utente di Dimensionamento automatico Amazon EC2.

Azione API	Motivo dell'errore o dell'interruzione	Causa
	`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 sulla gestione dei gruppi con dimensionamento automatico, consulta Gruppi con dimensionamento automatico nella Guida per l'utente di Dimensionamento automatico Amazon EC2.
`GetTaskProtection`	`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 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 `awsvpc`. Il numero di interfacce di rete che possono essere collegate alle istanze Amazon EC2 è limitato e l'interfaccia di rete primaria conta come una di queste. 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. Per errori `RESOURCE:GPU` , il numero di GPU richieste dal processo non è disponibile

Azione API	Motivo dell'errore o dell'interruzione	Causa
		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 `awsVpcConfiguration` .

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 `ecs.capability.task-eni` . Per ulteriori informazioni sugli attributi richiesti per determinati parametri di definizione di attività e variabili di configurazione dell'agente, consulta Parametri di definizione delle attività di Amazon ECS e Configurazione dell'agente del container Amazon ECS.
`StartTask`	`MISSING`	L'istanza del contenitore su cui hai tentato di avviare l'attività non è stata trovata. Controlla se è stato specificato il cluster o la regione errati o se l'ARN o l'ID dell'istanza del contenitore non è stato digitato correttamente.

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.
`UpdateTaskProtection`	`DEPLOYMENT_BLOCKED`	Impossibile impostare la protezione delle attività poiché una o più attività protette impediscono alla distribuzione del servizio di raggiungere uno stato stazionario. Annulla l'impostazione della protezione delle attività sulle attività esistenti o attendi la scadenza della protezione.
	`MISSING`	Il processo specificato non è stato trovato. Verifica che il nome o l'ARN del cluster e l'ARN o l'ID dell'attività siano validi.
	`TASK_NOT_VALID`	L'attività specificata non fa parte di un servizio Amazon ECS. Solo le attività gestite dai servizi Amazon ECS possono essere protette. Verifica l'ARN o l'ID dell'attività e riprova.

> **Note**
>
> Oltre agli scenari di errore descritti qui, le operazioni API possono fallire anche a causa di eccezioni, con conseguenti risposte di errore. Per un elenco di tali eccezioni, consulta Errori comuni.

Sicurezza in Amazon Elastic Container Service

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il modello di responsabilità condivisa descrive questo come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori di terze parti testano e verificano regolarmente l'efficacia della sicurezza come parte dei programmi di conformitàAWS. Per ulteriori informazioni sui programmi di conformità che si applicano ad Amazon Elastic Container Service, consulta Servizi AWS coperti dal programma di conformità.
- Sicurezza nel cloud: la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questa documentazione aiuta a comprendere come applicare il modello di responsabilità condivisa quando si utilizza Amazon ECS. Gli argomenti seguenti descrivono come configurare Amazon ECS per soddisfare gli obiettivi di sicurezza e conformità. Scopri anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse Amazon ECS.

Argomenti

Identity and Access Management per Amazon Elastic Container Registry

AWS Identity and Access Management (IAM) è un software Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle AWS risorse. Gli amministratori IAM controllano chi è autenticato (accesso effettuato) e autorizzato (dispone di autorizzazioni) a utilizzare risorse Amazon ECS. IAM è un software Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto in Amazon ECS.

Utente del servizio: se utilizzi il servizio Amazon ECS per svolgere il tuo lavoro, l'amministratore fornisce le credenziali e le autorizzazioni necessarie. All'aumentare del numero di funzionalità di Amazon ECS utilizzate per il lavoro, potrebbero essere necessarie ulteriori autorizzazioni. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità in Amazon ECS, consulta Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service.

Amministratore dei servizi: se sei il responsabile delle risorse Amazon ECS presso la tua azienda, probabilmente disponi dell'accesso completo ad Amazon ECS. Il compito dell'utente è determinare le

caratteristiche e le risorse di Amazon MQ a cui gli utenti del servizio devono accedere. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per ulteriori informazioni su come la tua azienda può utilizzare IAM con Amazon ECS, consulta Come funziona Amazon Elastic Container Service con IAM.

Amministratore IAM: se sei un amministratore IAM, potresti essere interessato a ottenere informazioni su come scrivere policy per gestire l'accesso ad Amazon ECS. Per visualizzare policy basate su identità di Amazon ECS di esempio che possono essere utilizzate in IAM, consulta Esempi di policy basate su identità per Amazon Elastic Container Service.

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWSnella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti

alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente IAM.

- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
  - Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
  - Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella Guida per l'utente IAM.
  - Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations

AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .

- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire 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 utilizzabili con Amazon Elastic Container Service

Funzionalità IAM	Supporto di Amazon ECS
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Parziale
Chiavi di condizione delle policy	Sì

Funzionalità IAM	Supporto di Amazon ECS
Liste di controllo degli accessi (ACL)	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Inoltro delle sessioni di accesso (FAS)	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una visione di alto livello di come Amazon ECS e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Policy basate su identità per Amazon ECS

Supporta le policy basate su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate su identità per Amazon ECS

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta Esempi di policy basate su identità per Amazon Elastic Container Service.

Policy basate su risorse all'interno di Amazon ECS

Supporta le policy basate su risorse	No

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Per consentire l'accesso multi-account, puoi specificare un intero account o entità IAM in un altro account come principale in una policy basata sulle risorse. L'aggiunta di un principale multi-account a una policy basata sulle risorse rappresenta solo una parte della relazione di trust. Quando il principale e la risorsa sono diversi Account AWS, un amministratore IAM dell'account affidabile deve inoltre concedere all'entità principale (utente o ruolo) l'autorizzazione ad accedere alla risorsa. L'autorizzazione viene concessa collegando all'entità una policy basata sull'identità. Tuttavia, se una policy basata su risorse concede l'accesso a un principale nello stesso account, non sono richieste ulteriori policy basate su identità. Per ulteriori informazioni, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di 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. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di operazioni di Amazon ECS, consulta Operazioni definite da Amazon Elastic Container Service nella documentazione di riferimento alla autorizzazione di servizio.

Le operazioni delle policy in Amazon ECS utilizzano il seguente prefisso prima dell'operazione:

```
ecs
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "ecs:action1",
      "ecs:action2"
         ]
```

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `Describe`, includi la seguente azione:

```
"Action": "ecs:Describe*"
```

Per visualizzare esempi di policy basate su identità Amazon ECS, consulta Esempi di policy basate su identità per Amazon Elastic Container Service.

Risorse di policy per Amazon ECS

Supporta le risorse di policy	Parziale

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale 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. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco di tipi di risorse di Amazon ECS, consulta Risorse definite da Amazon Elastic Container Service nella documentazione di riferimento alla autorizzazione di servizio. 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, separa gli ARN con le 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 il cluster `my-cluster-1` e `my-cluster-2` nell'istruzione, utilizza i seguenti ARN:

```
"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 $ {}: TaskDefinition RevisionNumber

```
"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. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Amazon ECS supporta le seguenti chiavi di condizione specifiche del servizio, utilizzabili per fornire filtri granulari per le tue policy IAM:

Chiave di condizione	Descrizione	Tipi di valutazione
aws: RequestTag /$ {} TagKey	La chiave di contesto è formattata `"aws:RequestTag/ tag-key":"tag-value "` laddove *tag-key* e *tag-value* sono la chiave del tag e la coppia di valori. Verifica che la coppia chiave-valore del tag sia presente in una richiesta. AWS Ad esempio, è possibile controllare che la richiesta includa la chiave di un tag `"Dept"` e che abbia il valore `"Accounting"` .	Stringa
aws: ResourceTag /$ {} TagKey	La chiave di contesto è formattata `"aws:ResourceTag/ tag-key":"tag-value "` laddove *tag-key* e *tag-value* sono la chiave del tag e la coppia di valori. 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 ha il formato `"aws:TagKeys":" tag-key"`, dove *tag-key* è una lista 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/ tag-key":"tag-value "` laddove *tag-key* e *tag-value* sono la chiave del tag e la coppia di valori. Controlla che il tag collegato alla risorsa dell'identità (utente o ruolo) corrisponda al nome e al valore della chiave specificata.	Stringa

Chiave di condizione	Descrizione	Tipi di valutazione
ecs:cluster	La chiave di contesto ha il formato `"ecs:cluster":"` *`cluster-arn`* `"`, dove *`cluster-arn`* è l'ARN del cluster Amazon ECS.	ARN, Null
ecs:container-instances	La chiave di contesto ha il formato `"ecs:container-instances":"` *`container-instance-arns`* `"`, dove *`container-instance-arns`* è uno o più ARN di istanza di container.	ARN, Null
ecs:container-name	La chiave di contesto ha il formato `"ecs:container-name":"` *`container-name`* `"`, dove *`container-instance-`* è il nome di un container Amazon ECS definito nella definizione di attività.	Stringa
ecs:enable-execute-command	La chiave di contesto ha il formato `"ecs:enable-execute-command":"` *`value`*`"`, dove *`value-`* è `"true"` o `"false"`.	Stringa
ecs:enable-service-connect	La chiave di contesto ha il formato `"ecs:enable-service-connect":"` *`value`*`"`, dove *`value`* è `"true"` o `"false"`.	Stringa
ecs: abilita i volumi ebs	La chiave di contesto ha il formato `"ecs:enable-ebs-volumes":"` *`value`*`"`, dove *`value`* è `"true"` o `"false"`.	Stringa
ecs:namespace	La chiave di contesto ha il formato `"ecs:namespace":"` *`namespace-arn`* `"`, dove *`namespace-arn`* è l'ARN dello spazio dei nomi AWS Cloud Map .	ARN, Null
ecs:service	La chiave di contesto ha il formato `"ecs:service":"` *`service-arn`* `"`, dove *`service-arn`* è l'ARN del servizio Amazon ECS.	ARN, Null

Chiave di condizione	Descrizione	Tipi di valutazione
ecs:task-definition	La chiave di contesto ha il formato `"ecs:task-definition":"` *`task-definition-arn`* `"`, dove *`task-definition-arn`* è l'ARN per la definizione di attività di Amazon ECS.	ARN, Null
ecs:account-setting	La chiave di contesto ha il formato `"ecs:account-setting":"` *`account-setting`* `"`, dove *`account-setting`* è il nome di un'impostazione dell'account Amazon ECS.	Stringa

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.

Liste di controllo degli accessi (ACL) in Amazon ECS

Supporta le ACL	No

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Controllo degli accessi basato su attributi (ABAC) con Amazon ECS

Important

Amazon ECS supporta il controllo degli accessi basato su attributi per tutte le risorse Amazon ECS. Per determinare se puoi utilizzare gli attributi per definire l'ambito di un'azione, utilizza la tabella Operazioni definite da Amazon ECS in Service Authorization Reference. Verifica

innanzitutto che vi sia una risorsa nella colonna Risorsa. Quindi, utilizza la colonna Chiavi di condizione per vedere le chiavi per la combinazione operazione/risorsa.

Supporta ABAC (tag nelle policy)	Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente IAM. Per visualizzare un tutorial con le fasi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Per ulteriori informazioni sull'aggiunta di tag alle risorse di Amazon ECS, consulta Taggare le risorse Amazon ECS.

Per visualizzare una policy basata sulle identità di esempio per limitare l'accesso a una risorsa basata su tag su tale risorsa, consulta Descrizione dei servizi Amazon ECS basati su tag.

Utilizzo di credenziali temporanee con Amazon ECS

Supporta le credenziali temporanee	Sì

Alcune Servizi AWS non funzionano quando accedi utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedi AWS utilizzando il link Single Sign-On (SSO) della tua azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Sessioni di accesso diretto per Amazon ECS

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per Amazon ECS

Supporta i ruoli di servizio	Sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per

ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella 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'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per maggiori dettagli su come creare e gestire i ruoli collegati ai servizi Amazon ECS, consulta Uso di ruoli collegati ai servizi per Amazon ECS.

Esempi di policy basate su identità per Amazon Elastic Container Service

Di default, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse di Amazon ECS. Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o l' AWS API. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle operazioni e sui tipi di risorse definiti da Amazon ECS, incluso il formato degli ARN per ogni tipo di risorse, consulta Operazioni, risorse e chiavi di condizione per Amazon Elastic Container Service in Service Authorization Reference (Guida di riferimento per l'autorizzazione del servizio).

Argomenti

Best practice relative alle policy di Amazon ECS

Le policy basate su identità determinano se qualcuno può creare, accedere o eliminare risorse Amazon ECS nell'account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche AWS gestite che concedono le autorizzazioni per molti casi d'uso comuni. Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile

scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Consenti agli utenti di Amazon ECS di visualizzare le proprie autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono cpllegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando programmaticamente l'API o. AWS CLI AWS

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
```

```
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Esempi di cluster Amazon ECS

La seguente policy IAM concede le autorizzazioni per creare ed elencare i cluster. Le operazioni `CreateCluster` e `ListClusters` non accettano risorse, pertanto la definizione delle risorse è impostata su `*` per tutte le risorse.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:ListClusters"
            ],
            "Resource": ["*"]
        }
    ]
}
```

La seguente policy IAM concede le autorizzazioni per descrivere ed eliminare un determinato cluster. Le operazioni `DescribeClusters` e `DeleteCluster` accettano gli ARN del cluster come risorse.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DescribeClusters",
                "ecs:DeleteCluster"
            ],
            "Resource": ["arn:aws:ecs:us-east-1:<aws_account_id>:cluster/
<cluster_name>"]
        }
    ]
}
```

La seguente policy IAM può essere collegata a un utente o a un gruppo per permettere solo a tale utente o gruppo di eseguire operazioni su un determinato cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Action": [
                "ecs:Describe*",
                "ecs:List*"
            ],
            "Effect": "Allow",
            "Resource": "*"
        },
        {
            "Action": [
                "ecs:DeleteCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:ListContainerInstances",
                "ecs:RegisterContainerInstance",
                "ecs:SubmitContainerStateChange",
                "ecs:SubmitTaskStateChange"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:ecs:us-east-1:<aws_account_id>:cluster/default"
        },
        {
            "Action": [
```

```
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTasks",
                "ecs:ListTasks",
                "ecs:UpdateContainerAgent",
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:RunTask"
            ],
            "Effect": "Allow",
            "Resource": "*",
            "Condition": {
                "ArnEquals": {"ecs:cluster": "arn:aws:ecs:us-
east-1:<aws_account_id>:cluster/default"}
            }
        }
    ]
}
```

Esempi di istanze di container Amazon ECS

La registrazione delle istanze di container viene gestita dall'agente Amazon ECS ma, in alcuni casi, puoi permettere a un utente di revocare manualmente la registrazione di un'istanza in un cluster. Ad esempio, se l'istanza di container è stata accidentalmente registrata nel cluster sbagliato o se l'istanza è stata interrotta con attività ancora in esecuzione.

La seguente policy IAM permette a un utente di elencare e revocare la registrazione delle istanze di container in un determinato cluster:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:DeregisterContainerInstance",
                "ecs:ListContainerInstances"
            ],
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>"]
        }
    ]
}
```

La seguente policy IAM permette a un utente di descrivere una specifica istanza di container in un determinato cluster. Per estendere questa autorizzazione a tutte le istanze di container in un cluster, puoi sostituire l'UUID dell'istanza di container con *.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:DescribeContainerInstances"],
            "Condition": {
                "ArnEquals": {"ecs:cluster":
 "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
            },
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:container-instance/
<cluster_name>/<container_instance_UUID>"]
        }
    ]
}
```

Esempi di definizione delle attività in Amazon ECS

Le policy IAM delle definizioni di attività non supportano le autorizzazioni a livello di risorsa, ma le seguenti policy IAM permettono a un utente di registrare, elencare e descrivere le definizioni di attività:

Se utilizzi la console, devi aggiungere `CloudFormation: CreateStack` come `Action`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:RegisterTaskDefinition",
                "ecs:ListTaskDefinitions",
                "ecs:DescribeTaskDefinition"
            ],
            "Resource": ["*"]
        }
    ]
}
```

Esempio di attività di esecuzione di Amazon ECS

Le risorse per `RunTask` sono definizioni di attività. Per limitare i cluster su cui un utente può eseguire le definizioni di attività, puoi specificarli nel blocco `Condition`. Il vantaggio è che non dovrai elencare sia le definizioni di attività che i cluster nelle tue risorse per consentire un accesso appropriato. Puoi utilizzarne uno o entrambi.

La seguente policy IAM concede l'autorizzazione per eseguire qualsiasi revisione di una specifica definizione di attività in un determinato cluster:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:RunTask"],
            "Condition": {
                "ArnEquals": {"ecs:cluster":
 "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
            },
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task-definition/
<task_family>:*"]
        }
    ]
}
```

Esempio di attività di avvio di Amazon ECS

Le risorse per `StartTask` sono definizioni di attività. Per limitare i cluster e le istanze di container su cui un utente può avviare le definizioni di attività, puoi specificarli nel blocco `Condition`. Il vantaggio è che non dovrai elencare sia le definizioni di attività che i cluster nelle tue risorse per consentire un accesso appropriato. Puoi utilizzarne uno o entrambi.

La seguente policy IAM concede l'autorizzazione per avviare qualsiasi revisione di una specifica definizione di attività in un cluster o un'istanza di container determinati.

> **Note**
>
> Per questo esempio, quando chiami l'`StartTask`API con il AWS CLI o un altro AWS SDK, devi specificare la revisione della definizione dell'attività in modo che la `Resource` mappatura corrisponda.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:StartTask"],
            "Condition": {
                "ArnEquals": {
                    "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>",
                    "ecs:container-instances":
 ["arn:aws:ecs:<region>:<aws_account_id>:container-instance/<cluster_name>/
<container_instance_UUID>"]
                }
            },
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task-definition/
<task_family>:*"]
        }
    ]
}
```

Elenca e descrivi esempi di attività di Amazon ECS

La seguente policy IAM permette a un utente di elencare i processi per un determinato cluster:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:ListTasks"],
            "Condition": {
                "ArnEquals": {"ecs:cluster":
 "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
            },
```

```
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>"]
        }
    ]
}
```

La seguente policy IAM permette a un utente di descrivere uno specifico processo in un determinato cluster:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:DescribeTasks"],
            "Condition": {
                "ArnEquals": {"ecs:cluster":
 "arn:aws:ecs:<region>:<aws_account_id>:cluster/<cluster_name>"}
            },
            "Resource": ["arn:aws:ecs:<region>:<aws_account_id>:task/<cluster_name>/
<task_UUID>"]
        }
    ]
}
```

Crea un esempio di servizio Amazon ECS

La seguente policy IAM permette a un utente di creare servizi Amazon ECS nella AWS Management Console:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:Describe*",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "ecs:List*",
```

```
                "ecs:Describe*",
                "ecs:CreateService",
                "elasticloadbalancing:Describe*",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:ListRoles",
                "iam:ListGroups",
                "iam:ListUsers"
            ],
            "Resource": ["*"]
        }
    ]
}
```

Esempio di aggiornamento del servizio Amazon ECS

La seguente policy IAM permette a un utente di aggiornare i servizi Amazon ECS nella AWS Management Console:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:Describe*",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "ecs:List*",
                "ecs:Describe*",
                "ecs:UpdateService",
                "iam:GetPolicy",
                "iam:GetPolicyVersion",
                "iam:GetRole",
                "iam:ListAttachedRolePolicies",
                "iam:ListRoles",
                "iam:ListGroups",
                "iam:ListUsers"
```

```
            ],
            "Resource": ["*"]
        }
    ]
}
```

Descrizione dei servizi Amazon ECS basati su tag

Puoi utilizzare condizioni nella policy basata su identità per controllare l'accesso alle risorse Amazon ECS in base ai tag. Questo esempio illustra il modo in cui è possibile creare una policy che consente di descrivere i servizi. Tuttavia, l'autorizzazione viene concessa solo se il valore del tag del servizio `Owner` è quello del nome utente dell'utente. Questa policy concede anche le autorizzazioni necessarie per completare questa azione nella console.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "DescribeServices",
            "Effect": "Allow",
            "Action": "ecs:DescribeServices",
            "Resource": "*"
        },
        {
            "Sid": "ViewServiceIfOwner",
            "Effect": "Allow",
            "Action": "ecs:DescribeServices",
            "Resource": "arn:aws:ecs:*:*:service/*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Owner": "${aws:username}"}
            }
        }
    ]
}
```

Puoi allegare questa policy agli utenti IAM nel tuo account. Se un utente denominato `richard-roe` prova a descrivere un servizio Amazon ECS, il servizio deve avere il tag `Owner=richard-roe` o `owner=richard-roe`. In caso contrario l'accesso è negato. La chiave di tag di condizione `Owner` corrisponde a `Owner` e `owner` perché i nomi delle chiavi di condizione non effettuano la distinzione tra maiuscole e minuscole. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.

Esempio di sostituzione dello spazio dei nomi di Amazon ECS Service Connect Deny

La seguente policy IAM impedisce a un utente di sovrascrivere lo spazio dei nomi predefinito di Service Connect in una configurazione di servizio. Lo spazio dei nomi predefinito è impostato nel cluster. Tuttavia, puoi sovrascrivere il parametro in una configurazione del servizio. Per coerenza, valuta la possibilità di impostare tutti i nuovi servizi in modo che utilizzino lo stesso spazio dei nomi. Utilizza le seguenti chiavi di contesto per richiedere ai servizi di utilizzare uno spazio dei nomi specifico. Nel seguente esempio, sostituisci `<region>`, `<aws_account_id>`, `<cluster_name>` e `<namespace_id>` con i tuoi valori.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:CreateService",
                "ecs:UpdateService"
            ],
            "Condition": {
                "ArnEquals": {
                    "ecs:cluster": "arn:aws:ecs:<region>:<aws_account_id>:cluster/
<cluster_name>",
                    "ecs:namespace":
 "arn:aws:servicediscovery:<region>:<aws_account_id>:namespace/<namespace_id>"
                }
            },
            "Resource": "*"
        }
    ]
}
```

AWS politiche gestite per Amazon Elastic Container Service

Per aggiungere autorizzazioni a utenti, gruppi e ruoli, è più facile utilizzare le politiche AWS gestite piuttosto che scrivere le politiche autonomamente. Creare policy gestite dal cliente IAM per fornire al tuo team solo le autorizzazioni di cui ha bisogno richiede tempo e competenza. Per iniziare rapidamente, puoi utilizzare le nostre politiche AWS gestite. Queste politiche coprono casi d'uso

comuni e sono disponibili nel tuo AWS account. Per ulteriori informazioni sulle policy AWS gestite, consulta le policy AWS gestite nella IAM User Guide.

AWS i servizi mantengono e aggiornano le politiche AWS gestite. Non è possibile modificare le autorizzazioni nelle politiche AWS gestite. I servizi aggiungono occasionalmente autorizzazioni aggiuntive a una policy AWS gestita per supportare nuove funzionalità. Questo tipo di aggiornamento interessa tutte le identità (utenti, gruppi e ruoli) a cui è collegata la policy. È più probabile che i servizi aggiornino una politica AWS gestita quando viene lanciata una nuova funzionalità o quando diventano disponibili nuove operazioni. I servizi non rimuovono le autorizzazioni da una policy AWS gestita, quindi gli aggiornamenti delle policy non comprometteranno le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy AWS gestita di ReadOnlyAccess fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consulta la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

Amazon ECS e Amazon ECR forniscono diverse policy gestite e relazioni di attendibilità che possono essere collegate a utenti, gruppi, ruoli, istanze Amazon EC2 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.

AmazonECS_ FullAccess

È possibile allegare la policy `AmazonECS_FullAccess`alle identità IAM.

Questa policy concede l'accesso amministrativo alle risorse di Amazon ECS e concede a un'identità IAM (ad esempio un utente, un gruppo o un ruolo) l'accesso ai servizi con AWS cui è integrato Amazon ECS per utilizzare tutte le funzionalità di Amazon ECS. L'utilizzo di questa policy consente di accedere a tutte le funzioni Amazon ECS disponibili nella AWS Management Console.

Dettagli dell'autorizzazione

La policy IAM gestita da `AmazonECS_FullAccess` deve includere le seguenti autorizzazioni: 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.

- `ecs`— Consente ai responsabili l'accesso completo a tutte le operazioni API di Amazon ECS.
- `application-autoscaling`: consente ai principali di creare, descrivere e gestire risorse di Application Auto Scaling. Questo è necessario quando si abilita il dimensionamento automatico del servizio per i servizi Amazon ECS.
- `appmesh`: consente ai principali di elencare le mesh del servizio App Mesh e i nodi virtuali e di descrivere i nodi virtuali App Mesh. Ciò è necessario quando si integrano i servizi Amazon ECS con App Mesh.
- `autoscaling`: consente ai principali di creare, gestire e descrivere le risorse di Amazon EC2 Auto Scaling. Ciò è necessario quando si gestiscono i gruppi di Amazon EC2 Auto Scaling quando si utilizza la funzionalità di scalabilità automatica del cluster.
- `cloudformation`— Consente ai principali di creare e gestire gli stack. AWS CloudFormation Ciò è necessario quando si creano cluster Amazon ECS tramite la AWS Management Console e la successiva gestione di tali cluster.
- `cloudwatch`— Consente ai responsabili di creare, gestire e descrivere gli CloudWatch allarmi Amazon.
- `codedeploy`— Consente ai responsabili di creare e gestire le distribuzioni delle applicazioni e di visualizzarne le configurazioni, le revisioni e gli obiettivi di distribuzione.
- `sns`: consente ai principali di visualizzare un elenco di argomenti di Amazon SNS.
- `lambda`: consente ai principali di visualizzare un elenco di funzioni AWS Lambda e le relative configurazioni specifiche della versione.
- `ec2`— Consente ai responsabili di eseguire istanze Amazon EC2 e creare e gestire percorsi, tabelle di routing, gateway Internet, gruppi di avvio, gruppi di sicurezza, cloud privati virtuali, flotte Spot e sottoreti.
- `elasticloadbalancing`: consente ai principali di creare, descrivere ed eliminare i load balancer di Elastic Load Balancing. I principali saranno inoltre in grado di gestire completamente i gruppi di destinazione, gli ascoltatori e le regole dell'ascoltatore per i sistemi di bilanciamento del carico.
- `events`— Consente ai mandanti di creare, gestire ed eliminare EventBridge le regole di Amazon e i relativi obiettivi.
- `iam`: consente ai principali di elencare i ruoli IAM e le relative policy associate. I principali possono anche elencare i profili dell'istanza disponibili per le istanze Amazon EC2.
- `logs`— Consente ai responsabili di creare e descrivere i gruppi di log di Amazon CloudWatch Logs. I principali possono elencare anche gli eventi di log per questi gruppi di log.
- `route53`: consente ai principali di creare, gestire ed eliminare le zone ospitate di Amazon Route 53. I principali possono anche visualizzare anche le informazioni e la configurazione del controllo

dell'integrità di Amazon Route 53. Per ulteriori informazioni sulle zone ospitate, consulta Utilizzo di zone ospitate.

- `servicediscovery`— Consente ai responsabili di creare, gestire ed eliminare AWS Cloud Map servizi e creare namespace DNS privati.

Di seguito è riportata una policy `AmazonECS_FullAccess` di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:DeleteScalingPolicy",
                "application-autoscaling:DeregisterScalableTarget",
                "application-autoscaling:DescribeScalableTargets",
                "application-autoscaling:DescribeScalingActivities",
                "application-autoscaling:DescribeScalingPolicies",
                "application-autoscaling:PutScalingPolicy",
                "application-autoscaling:RegisterScalableTarget",
                "appmesh:DescribeVirtualGateway",
                "appmesh:DescribeVirtualNode",
                "appmesh:ListMeshes",
                "appmesh:ListVirtualGateways",
                "appmesh:ListVirtualNodes",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:Describe*",
                "autoscaling:UpdateAutoScalingGroup",
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:PutMetricAlarm",
                "codedeploy:BatchGetApplicationRevisions",
                "codedeploy:BatchGetApplications",
                "codedeploy:BatchGetDeploymentGroups",
```

```
                "codedeploy:BatchGetDeployments",
                "codedeploy:ContinueDeployment",
                "codedeploy:CreateApplication",
                "codedeploy:CreateDeployment",
                "codedeploy:CreateDeploymentGroup",
                "codedeploy:GetApplication",
                "codedeploy:GetApplicationRevision",
                "codedeploy:GetDeployment",
                "codedeploy:GetDeploymentConfig",
                "codedeploy:GetDeploymentGroup",
                "codedeploy:GetDeploymentTarget",
                "codedeploy:ListApplicationRevisions",
                "codedeploy:ListApplications",
                "codedeploy:ListDeploymentConfigs",
                "codedeploy:ListDeploymentGroups",
                "codedeploy:ListDeployments",
                "codedeploy:ListDeploymentTargets",
                "codedeploy:RegisterApplicationRevision",
                "codedeploy:StopDeployment",
                "ec2:AssociateRouteTable",
                "ec2:AttachInternetGateway",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:CancelSpotFleetRequests",
                "ec2:CreateInternetGateway",
                "ec2:CreateLaunchTemplate",
                "ec2:CreateRoute",
                "ec2:CreateRouteTable",
                "ec2:CreateSecurityGroup",
                "ec2:CreateSubnet",
                "ec2:CreateVpc",
                "ec2:DeleteLaunchTemplate",
                "ec2:DeleteSubnet",
                "ec2:DeleteVpc",
                "ec2:Describe*",
                "ec2:DetachInternetGateway",
                "ec2:DisassociateRouteTable",
                "ec2:ModifySubnetAttribute",
                "ec2:ModifyVpcAttribute",
                "ec2:RequestSpotFleet",
                "ec2:RunInstances",
                "ecs:*",
                "elasticfilesystem:DescribeAccessPoints",
                "elasticfilesystem:DescribeFileSystems",
                "elasticloadbalancing:CreateListener",
```

```
                "elasticloadbalancing:CreateLoadBalancer",
                "elasticloadbalancing:CreateRule",
                "elasticloadbalancing:CreateTargetGroup",
                "elasticloadbalancing:DeleteListener",
                "elasticloadbalancing:DeleteLoadBalancer",
                "elasticloadbalancing:DeleteRule",
                "elasticloadbalancing:DeleteTargetGroup",
                "elasticloadbalancing:DescribeListeners",
                "elasticloadbalancing:DescribeLoadBalancers",
                "elasticloadbalancing:DescribeRules",
                "elasticloadbalancing:DescribeTargetGroups",
                "events:DeleteRule",
                "events:DescribeRule",
                "events:ListRuleNamesByTarget",
                "events:ListTargetsByRule",
                "events:PutRule",
                "events:PutTargets",
                "events:RemoveTargets",
                "fsx:DescribeFileSystems",
                "iam:ListAttachedRolePolicies",
                "iam:ListInstanceProfiles",
                "iam:ListRoles",
                "lambda:ListFunctions",
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:FilterLogEvents",
                "route53:CreateHostedZone",
                "route53:DeleteHostedZone",
                "route53:GetHealthCheck",
                "route53:GetHostedZone",
                "route53:ListHostedZonesByName",
                "servicediscovery:CreatePrivateDnsNamespace",
                "servicediscovery:CreateService",
                "servicediscovery:DeleteService",
                "servicediscovery:GetNamespace",
                "servicediscovery:GetOperation",
                "servicediscovery:GetService",
                "servicediscovery:ListNamespaces",
                "servicediscovery:ListServices",
                "servicediscovery:UpdateService",
                "sns:ListTopics"
            ],
            "Resource": ["*"]
        },
```

```
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameter",
                "ssm:GetParameters",
                "ssm:GetParametersByPath"
            ],
            "Resource": "arn:aws:ssm:*:*:parameter/aws/service/ecs*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DeleteInternetGateway",
                "ec2:DeleteRoute",
                "ec2:DeleteRouteTable",
                "ec2:DeleteSecurityGroup"
            ],
            "Resource": ["*"],
            "Condition": {
                "StringLike": {"ec2:ResourceTag/aws:cloudformation:stack-name":
"EC2ContainerService-*"}
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": ["*"],
            "Condition": {
                "StringLike": {"iam:PassedToService": "ecs-tasks.amazonaws.com"}
            }
        },
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": ["arn:aws:iam::*:role/ecsInstanceRole*"],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn"
                    ]
                }
            }
        },
```

```
        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": ["arn:aws:iam::*:role/ecsAutoscaleRole*"],
            "Condition": {
                "StringLike": {
                    "iam:PassedToService": [
                        "application-autoscaling.amazonaws.com",
                        "application-autoscaling.amazonaws.com.cn"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "iam:AWSServiceName": [
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com",
                        "ecs.application-autoscaling.amazonaws.com",
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": ["elasticloadbalancing:AddTags"],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "elasticloadbalancing:CreateAction": [
                        "CreateTargetGroup",
                        "CreateRule",
                        "CreateListener",
                        "CreateLoadBalancer"
                    ]
                }
            }
        }
```

```
    ]
}
```

Volumi AmazonECS InfrastructureRole PolicyFor

La policy IAM `AmazonECSInfrastructureRolePolicyForVolumes` gestita concede le autorizzazioni necessarie ad Amazon ECS per effettuare chiamate AWS API per tuo conto. Puoi collegare questa policy al ruolo IAM che fornisci con la configurazione del volume quando avvii attività e servizi Amazon ECS. Il ruolo consente ad Amazon ECS di gestire i volumi collegati alle tue attività. Per ulteriori informazioni, consulta il ruolo IAM dell'infrastruttura Amazon ECS.

Dettagli dell'autorizzazione

La policy IAM gestita da `AmazonECSInfrastructureRolePolicyForVolumes` deve includere le seguenti autorizzazioni: Seguendo i consigli di sicurezza standard che prevedono la concessione del privilegio minimo, è possibile utilizzare la politica `AmazonECSInfrastructureRolePolicyForVolumes` gestita come modello per creare una politica personalizzata che includa solo le autorizzazioni necessarie.

- `ec2:CreateVolume`— Consente a un principale di creare un volume Amazon EBS se e solo se è etichettato con i `AmazonECSManaged` tag `AmazonECSCreated` and. Questa autorizzazione è necessaria per creare volumi Amazon EBS collegati alle attività di Amazon ECS e ridurre al minimo le autorizzazioni fornite ad Amazon ECS da questa politica.
- `ec2:CreateTags`— Consente a un principale di aggiungere tag a un volume Amazon EBS come parte di`ec2:CreateVolume`. Questa autorizzazione è richiesta da Amazon ECS per aggiungere tag specificati dal cliente ai volumi Amazon EBS creati per tuo conto.
- `ec2:AttachVolume`— Consente a un principale di collegare un volume Amazon EBS a un'istanza Amazon EC2. Questa autorizzazione è richiesta da Amazon ECS per collegare i volumi Amazon EBS all'istanza Amazon EC2 che ospita l'attività Amazon ECS associata.
- `ec2:DescribeVolume`— Consente a un responsabile di recuperare informazioni sui volumi Amazon EBS. Questa autorizzazione è necessaria per gestire il ciclo di vita dei volumi Amazon EBS.
- `ec2:DescribeAvailabilityZones`— Consente a un preside di recuperare informazioni sulle zone di disponibilità nel tuo account. Ciò è necessario per gestire il ciclo di vita dei volumi EBS.
- `ec2:DetachVolume`— Consente a un principale di scollegare un volume Amazon EBS da un'istanza Amazon EC2. Questa autorizzazione è richiesta da Amazon ECS per scollegare il

volume Amazon EBS dall'istanza Amazon EC2 che ospita l'attività Amazon ECS associata quando l'attività termina.

- `ec2:DeleteVolume`— Consente a un principale di eliminare un volume Amazon EBS. Questa autorizzazione è richiesta da Amazon ECS per eliminare i volumi Amazon EBS che non sono più utilizzati dall'attività Amazon ECS.
- `ec2:DeleteTags`— Consente a un principale di eliminare il `AmazonECSManaged` tag da un volume Amazon EBS. Questa autorizzazione è richiesta da Amazon ECS per rimuovere l'accesso a un volume Amazon EBS dopo che non è più associato a un carico di lavoro Amazon ECS. Questo è applicabile solo quando un volume Amazon EBS non viene eliminato dopo l'arresto dell'attività.

Di seguito è riportata una policy `AmazonECSInfrastructureRolePolicyForVolumes` di esempio.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Sid": "CreateEBSManagedVolume",
   "Effect": "Allow",
   "Action": "ec2:CreateVolume",
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "ArnLike": {
     "aws:RequestTag/AmazonECSCreated": "arn:aws:ecs:*:*:task/*"
    },
    "StringEquals": {
     "aws:RequestTag/AmazonECSManaged": "true"
    }
   }
  },
  {
   "Sid": "TagOnCreateVolume",
   "Effect": "Allow",
   "Action": "ec2:CreateTags",
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "ArnLike": {
     "aws:RequestTag/AmazonECSCreated": "arn:aws:ecs:*:*:task/*"
    },
```

```
    "StringEquals": {
     "ec2:CreateAction": "CreateVolume",
     "aws:RequestTag/AmazonECSManaged": "true"
    }
   }
  },
  {
   "Sid": "DescribeVolumesForLifecycle",
   "Effect": "Allow",
   "Action": [
    "ec2:DescribeVolumes",
    "ec2:DescribeAvailabilityZones"
   ],
   "Resource": "*"
  },
  {
   "Sid": "ManageEBSVolumeLifecycle",
   "Effect": "Allow",
   "Action": [
    "ec2:AttachVolume",
    "ec2:DetachVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
    "StringEquals": {
     "aws:ResourceTag/AmazonECSManaged": "true"
    }
   }
  },
  {
   "Sid": "ManageVolumeAttachmentsForEC2",
   "Effect": "Allow",
   "Action": [
    "ec2:AttachVolume",
    "ec2:DetachVolume"
   ],
   "Resource": "arn:aws:ec2:*:*:instance/*"
  },
  {
   "Sid": "DeleteEBSManagedVolume",
   "Effect": "Allow",
   "Action": "ec2:DeleteVolume",
   "Resource": "arn:aws:ec2:*:*:volume/*",
   "Condition": {
```

```
    "ArnLike": {
     "aws:ResourceTag/AmazonECSCreated": "arn:aws:ecs:*:*:task/*"
    },
    "StringEquals": {
     "aws:ResourceTag/AmazonECSManaged": "true"
    }
   }
  }
 ]
}
```

ContainerServiceforRuolo EC2 di Amazon EC2

Amazon ECS collega questa policy a un ruolo di servizio che consente ad Amazon ECS di eseguire operazioni per tuo conto rispetto a istanze Amazon EC2 o istanze esterne.

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.

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. Per ulteriori informazioni, consulta Dettagli dell'autorizzazione.
- 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
  ```

Dettagli dell'autorizzazione

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`: consente a un principale di descrivere i tag associati a un'istanza Amazon EC2. 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`: 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`: consente a un principale di annullare la registrazione di un'istanza di container di Amazon ECS da un cluster. L'agente container Amazon ECS non chiama questa operazione API, ma questa autorizzazione rimane per garantire la compatibilità con le versioni precedenti.
- `ecs:DiscoverPollEndpoint`: questa operazione restituisce gli endpoint utilizzati dall'agente del container di Amazon ECS per eseguire il polling per gli aggiornamenti.
- `ecs:Poll`: consente all'agente del container di Amazon ECS di comunicare con il piano di controllo Amazon ECS per segnalare le modifiche dello stato del processo.
- `ecs:RegisterContainerInstance`: 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'istanza Amazon EC2 in un cluster e per supportare la propagazione dei tag di risorse.
- `ecs:StartTelemetrySession`: 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 processo.
- `ecs:TagResource`: 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*`: 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'`SubmitContainerStateChange`autorizzazione non viene più utilizzata dall'agente container Amazon ECS, ma serve a garantire la compatibilità con le versioni precedenti.
- `ecr:GetAuthorizationToken`: 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`: 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`: 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`: 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`— Consente a un principale di creare un flusso di log di CloudWatch Logs per un gruppo di log specificato.
- `logs:PutLogEvents`: consente a un principale di caricare un batch di eventi di log in un flusso di log specificato.

Di seguito è riportata una policy `AmazonEC2ContainerServiceforEC2Role` di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTags",
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:UpdateContainerInstancesState",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": "ecs:TagResource",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ecs:CreateAction": [
                        "CreateCluster",
                        "RegisterContainerInstance"
                    ]
                }
            }
        }
    ]
}
```

Amazon EC2 ContainerService EventsRole

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.

Dettagli dell'autorizzazione

Questa policy include le seguenti autorizzazioni:

- `ecs`— Consente a un responsabile di un servizio di chiamare l' RunTask API Amazon ECS. Consente a un principale di un servizio di aggiungere tag (`TagResource`) quando chiama l' RunTask API Amazon ECS.
- `iam`: consente di trasmettere qualsiasi ruolo di servizio IAM a qualsiasi attività Amazon ECS.

Di seguito è riportata una policy `AmazonEC2ContainerServiceEventsRole` di esempio.

```
{
    "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"]
                }
            }
        }
    ]
}
```

Amazonecs TaskExecution RolePolicy

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.

Dettagli dell'autorizzazione

La policy IAM gestita da `AmazonECSTaskExecutionRolePolicy` deve includere le seguenti autorizzazioni: Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, la policy gestita `AmazonECSTaskExecutionRolePolicy`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.

- `ecr:GetAuthorizationToken`: 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`: 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 viene premuto, il livello dell'immagine viene ignorato.
- `ecr:GetDownloadUrlForLayer`: 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`: 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`— Consente a un principale di creare un flusso di log di CloudWatch Logs per un gruppo di log specificato.
- `logs:PutLogEvents`: consente a un principale di caricare un batch di eventi di log in un flusso di log specificato.

Di seguito è riportata una policy `AmazonECSTaskExecutionRolePolicy` di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

Politica AmazonECS ServiceRole

La policy IAM gestita `AmazonECSServiceRolePolicy` consente ad Amazon Elastic Container Service di gestire il cluster. Questa policy può essere aggiunta al ruolo IAM di esecuzione dell''attività. Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

Dettagli dell'autorizzazione

La policy IAM gestita da `AmazonECSServiceRolePolicy` deve includere le seguenti autorizzazioni: Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, la policy gestita `AmazonECSServiceRolePolicy`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.

- `autoscaling`: consente ai principali di creare, gestire e descrivere le risorse di Dimensionamento automatico Amazon EC2. Ciò è necessario quando si gestiscono i gruppi di Amazon EC2 Auto Scaling quando si utilizza la funzionalità di scalabilità automatica del cluster.
- `autoscaling-plans`: consente ai principali di creare, eliminare e descrivere i piani di dimensionamento automatico.
- `cloudwatch`— Consente ai responsabili di creare, gestire e descrivere gli CloudWatch allarmi Amazon.
- `ec2`— Consente l'esecuzione dei principali su istanze Amazon EC2 e la creazione e la gestione di interfacce e tag di rete.
- `elasticloadbalancing`: consente ai principali di creare, descrivere ed eliminare i load balancer di Elastic Load Balancing. I presidi saranno inoltre in grado di aggiungere e descrivere i gruppi target.
- `logs`— Consente ai responsabili di creare e descrivere i gruppi di log di Amazon CloudWatch Logs. I principali possono elencare anche gli eventi di log per questi gruppi di log.
- `route53`: consente ai principali di creare, gestire ed eliminare le zone ospitate di Amazon Route 53. I principali possono anche visualizzare anche le informazioni e la configurazione del controllo dell'integrità di Amazon Route 53. Per ulteriori informazioni sulle zone ospitate, consulta Utilizzo di zone ospitate.
- `servicediscovery`— Consente ai responsabili di creare, gestire ed eliminare AWS Cloud Map servizi e creare namespace DNS privati.
- `events`— Consente ai mandanti di creare, gestire ed eliminare EventBridge le regole di Amazon e i relativi obiettivi.

Di seguito è riportata una policy `AmazonECSServiceRolePolicy` di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "ECSTaskManagement",
            "Effect": "Allow",
            "Action": [
                "ec2:AttachNetworkInterface",
                "ec2:CreateNetworkInterface",
                "ec2:CreateNetworkInterfacePermission",
                "ec2:DeleteNetworkInterface",
                "ec2:DeleteNetworkInterfacePermission",
                "ec2:Describe*",
                "ec2:DetachNetworkInterface",
                "elasticloadbalancing:DeregisterInstancesFromLoadBalancer",
                "elasticloadbalancing:DeregisterTargets",
                "elasticloadbalancing:Describe*",
                "elasticloadbalancing:RegisterInstancesWithLoadBalancer",
                "elasticloadbalancing:RegisterTargets",
                "route53:ChangeResourceRecordSets",
                "route53:CreateHealthCheck",
                "route53:DeleteHealthCheck",
                "route53:Get*",
                "route53:List*",
                "route53:UpdateHealthCheck",
                "servicediscovery:DeregisterInstance",
                "servicediscovery:Get*",
                "servicediscovery:List*",
                "servicediscovery:RegisterInstance",
                "servicediscovery:UpdateInstanceCustomHealthStatus"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AutoScaling",
            "Effect": "Allow",
            "Action": [
                "autoscaling:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AutoScalingManagement",
            "Effect": "Allow",
            "Action": [
                "autoscaling:DeletePolicy",
                "autoscaling:PutScalingPolicy",
```

```
                "autoscaling:SetInstanceProtection",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:PutLifecycleHook",
                "autoscaling:DeleteLifecycleHook",
                "autoscaling:CompleteLifecycleAction",
                "autoscaling:RecordLifecycleActionHeartbeat"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "autoscaling:ResourceTag/AmazonECSManaged": "false"
                }
            }
        },
        {
            "Sid": "AutoScalingPlanManagement",
            "Effect": "Allow",
            "Action": [
                "autoscaling-plans:CreateScalingPlan",
                "autoscaling-plans:DeleteScalingPlan",
                "autoscaling-plans:DescribeScalingPlans",
                "autoscaling-plans:DescribeScalingPlanResources"
            ],
            "Resource": "*"
        },
        {
            "Sid": "EventBridge",
            "Effect": "Allow",
            "Action": [
                "events:DescribeRule",
                "events:ListTargetsByRule"
            ],
            "Resource": "arn:aws:events:*:*:rule/ecs-managed-*"
        },
        {
            "Sid": "EventBridgeRuleManagement",
            "Effect": "Allow",
            "Action": [
                "events:PutRule",
                "events:PutTargets"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "events:ManagedBy": "ecs.amazonaws.com"
                }
            }
        },
        {
            "Sid": "CWAlarmManagement",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm"
            ],
            "Resource": "arn:aws:cloudwatch:*:*:alarm:*"
        },
        {
            "Sid": "ECSTagging",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": "arn:aws:ec2:*:*:network-interface/*"
        },
        {
            "Sid": "CWLogGroupManagement",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:DescribeLogGroups",
                "logs:PutRetentionPolicy"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/ecs/*"
        },
        {
            "Sid": "CWLogStreamManagement",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/ecs/*:log-stream:*"
        },
        {
            "Sid": "ExecuteCommandSessionManagement",
```

```
            "Effect": "Allow",
            "Action": [
                "ssm:DescribeSessions"
            ],
            "Resource": "*"
        },
        {
            "Sid": "ExecuteCommand",
            "Effect": "Allow",
            "Action": [
                "ssm:StartSession"
            ],
            "Resource": [
                "arn:aws:ecs:*:*:task/*",
                "arn:aws:ssm:*:*:document/AmazonECS-ExecuteInteractiveCommand"
            ]
        },
        {
            "Sid": "CloudMapResourceCreation",
            "Effect": "Allow",
            "Action": [
                "servicediscovery:CreateHttpNamespace",
                "servicediscovery:CreateService"
            ],
            "Resource": "*",
            "Condition": {
                "ForAllValues:StringEquals": {
                    "aws:TagKeys": [
                        "AmazonECSManaged"
                    ]
                }
            }
        },
        {
            "Sid": "CloudMapResourceTagging",
            "Effect": "Allow",
            "Action": "servicediscovery:TagResource",
            "Resource": "*",
            "Condition": {
                "StringLike": {
                    "aws:RequestTag/AmazonECSManaged": "*"
                }
            }
        },
```

```
        {
            "Sid": "CloudMapResourceDeletion",
            "Effect": "Allow",
            "Action": [
                "servicediscovery:DeleteService"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AmazonECSManaged": "false"
                }
            }
        },
        {
            "Sid": "CloudMapResourceDiscovery",
            "Effect": "Allow",
            "Action": [
                "servicediscovery:DiscoverInstances",
                "servicediscovery:DiscoverInstancesRevision"
            ],
            "Resource": "*"
        }
    ]
}
```

AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecur

Fornisce accesso amministrativo a AWS Private Certificate Authority Secrets Manager e ad altri AWS servizi necessari per gestire le funzionalità TLS di Amazon ECS Service Connect per tuo conto.

Dettagli dell'autorizzazione

La policy IAM gestita da `AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity` deve includere le seguenti autorizzazioni: Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, la policy gestita `AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity`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.

- `secretsmanager:CreateSecret`— Consente al preside di creare il segreto. È necessario per Service Connect TLS, Amazon ECS mantiene segreta la chiave privata del cliente nel Secrets Manager del cliente.
- `secretsmanager:TagResource`— Consente al preside di allegare un tag al segreto creato. È necessario per Service Connect TLS, perché Amazon ECS crea il segreto per conto del cliente e allega tag a risorsa. Questi tag forniscono al cliente un modo più semplice per identificare il segreto gestito e limitare le azioni su tali segreti.
- `secretsmanager:DescribeSecret`— Consenti al principale di descrivere il segreto e recuperare la fase della versione corrente. È necessario che Amazon ECS esegua la rotazione dei materiali TLS di Amazon ECS Service Connect.
- `secretsmanager:UpdateSecret`— Consenti al preside di aggiornare il segreto. È necessario che Amazon ECS esegua la rotazione dei materiali TLS di Amazon ECS Service Connect e aggiorni il segreto con nuovi materiali.
- `secretsmanager:GetSecretValue`— Consenti al preside di ottenere il valore segreto. È necessario che Amazon ECS esegua la rotazione dei materiali TLS di Amazon ECS Service Connect.
- `secretsmanager:PutSecretValue`— Consenti al preside di inserire il valore segreto. È necessario che Amazon ECS esegua la rotazione dei materiali TLS di Amazon ECS Service Connect.
- `secretsmanager:UpdateSecretVersionStage`— Consenti al principale di aggiornare la fase della versione segreta. È necessario che Amazon ECS esegua la rotazione dei materiali TLS di Amazon ECS Service Connect.
- `acm-pca:IssueCertificate`— Consenti al preside di IssueCertificate `End entity certificate` richiedere Amazon ECS Service Connect TLS. Era necessario che ECS generasse un certificato per il servizio upstream del cliente.
- `acm-pca:GetCertificate`— Consenti al preside di GetCertificate `End entity certificate` richiedere Amazon ECS Service Connect TLS.
- `acm-pca:GetCertificateAuthorityCertificate`— Consenti al preside di ottenere il certificato dell'autorità di certificazione. È necessario per Amazon ECS Service Connect TLS in modo che il servizio downstream del cliente possa fidarsi del certificato dell'entità finale upstream.
- `acm-pca:DescribeCertificateAuthority`— Consenti al preside di ottenere dettagli sull'autorità di certificazione. È necessario che Amazon ECS Service Connect TLS riutilizzi informazioni come l'algoritmo di firma per creare la CSR (Certificate Signing Request).

Di seguito è riportata una policy `AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity` di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CreateSecret",
            "Effect": "Allow",
            "Action": "secretsmanager:CreateSecret",
            "Resource": "arn:aws:secretsmanager:*:*:secret:ecs-sc!*",
            "Condition": {
                "ArnLike": {
                    "aws:RequestTag/AmazonECSCreated": [
                        "arn:aws:ecs:*:*:service/*/*",
                        "arn:aws:ecs:*:*:task-set/*/*"
                    ]
                },
                "StringEquals": {
                    "aws:RequestTag/AmazonECSManaged": "true",
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "TagOnCreateSecret",
            "Effect": "Allow",
            "Action": "secretsmanager:TagResource",
            "Resource": "arn:aws:secretsmanager:*:*:secret:ecs-sc!*",
            "Condition": {
                "ArnLike": {
                     "aws:RequestTag/AmazonECSCreated": [
                        "arn:aws:ecs:*:*:service/*/*",
                        "arn:aws:ecs:*:*:task-set/*/*"
                    ]
                },
                "StringEquals": {
                    "aws:RequestTag/AmazonECSManaged": "true",
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
```

```
        {
            "Sid": "RotateTLSCertificateSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:DescribeSecret",
                "secretsmanager:UpdateSecret",
                "secretsmanager:GetSecretValue",
                "secretsmanager:PutSecretValue",
                "secretsmanager:DeleteSecret",
                "secretsmanager:RotateSecret",
                "secretsmanager:UpdateSecretVersionStage"
            ],
            "Resource": "arn:aws:secretsmanager:*:*:secret:ecs-sc!*",
            "Condition": {
                "StringEquals": {
                    "secretsmanager:ResourceTag/aws:secretsmanager:owningService":
"ecs-sc",
                    "aws:ResourceAccount": "${aws:PrincipalAccount}"
                }
            }
        },
        {
            "Sid": "ManagePrivateCertificateAuthority",
            "Effect": "Allow",
            "Action": [
                "acm-pca:GetCertificate",
                "acm-pca:GetCertificateAuthorityCertificate",
                "acm-pca:DescribeCertificateAuthority"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/AmazonECSManaged": "true"
                }
            }
        },
        {
            "Sid": "ManagePrivateCertificateAuthorityForIssuingEndEntityCertificate",
            "Effect": "Allow",
            "Action": [
                "acm-pca:IssueCertificate"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "StringEquals": {
                    "aws:ResourceTag/AmazonECSManaged": "true",
                    "acm-pca:TemplateArn":"arn:aws:acm-pca:::template/
EndEntityCertificate/V1"
                }
            }
        }
    ]
}
```

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.

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.

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 conto dell'utente. Per ulteriori informazioni, consulta Creare un ruolo di servizio CodeDeploy nella Guida per l'AWS CodeDeploy utente.

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 una nuova policy di sicurezza AmazonECS InfrastructureRole PolicyFor ServiceConnect Transport Layer	È stata aggiunta una nuova policy Infrastru ctureRolePolicyForServiceCo nnectTransportLayerSecurity AmazonECS che fornisce l'accesso amministrativo a AWS Private Certificate Authority Secrets Manager e consente alle funzional ità TLS di Amazon ECS Service Connect di funzionare correttamente. AWS KMS	22 gennaio 2024
Aggiungi una nuova politica AmazonECS Volumes InfrastructureRole PolicyFor	La `AmazonECSInfrastru ctureRolePolicyFor Volumes` policy è stata aggiunta. La policy concede le autorizzazioni necessarie ad Amazon ECS per effettuar e chiamate AWS API per gestire i volumi Amazon EBS associati ai carichi di lavoro Amazon ECS.	11 gennaio 2024
Aggiungi autorizzazioni alla politica di ServiceRo leAmazonECS	La policy IAM `AmazonECS ServiceRolePolicy` gestita è stata aggiornata con nuove autorizzazioni e `events` autorizzazioni e aggiuntive. `autoscaling` `autoscaling-plans`	4 dicembre 2023
Aggiungi autorizzazioni ad Amazon EC2 Container Service EventsRole	La policy IAM `AmazonECS ServiceRolePolicy` gestita è stata aggiornat	4 ottobre 2023

Modifica	Descrizione	Data
	a per consentire l'accesso al funzionamento dell'API. AWS Cloud Map `DiscoverInstancesRevision`	
Aggiungi autorizzazioni a ContainerServiceforAmazonEC2 EC2Role	La `AmazonEC2ContainerServiceforEC2Role` politica è stata modificata per aggiungere l'`ecs:TagResource` autorizzazione, che include una condizione che limita l'autorizzazione solo ai cluster appena creati e alle istanze di container registrate.	6 marzo 2023
Aggiunta di autorizzazioni a the section called “Amazon ECS_ FullAccess”	La `AmazonECS_FullAccess` politica è stata modificata per aggiungere l'`elasticloadbalancing:AddTags` autorizzazione, che include una condizione che limita l'autorizzazione solo ai sistemi di bilanciamento del carico, ai gruppi target, alle regole e ai listener creati di recente. Questa autorizzazione non consente l'aggiunta di tag a risorse Elastic Load Balancing già create.	4 gennaio 2023
Amazon ECS ha cominciato a tenere traccia delle modifiche	Amazon ECS ha iniziato a tracciare le modifiche per le sue politiche AWS gestite.	8 giugno 2021

Politiche IAM AWS gestite eliminate gradualmente per Amazon Elastic Container Service

Le seguenti politiche IAM AWS gestite vengono eliminate gradualmente. Queste policy vengono ora sostituiti da policy aggiornate. È consigliabile aggiornare gli utenti o i ruoli in modo che utilizzino le policy aggiornate.

Amazon EC2 ContainerService FullAccess

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`.

Ruolo Amazon EC2 ContainerService

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 ContainerService AutoscaleRole

Important

La policy IAM gestita da `AmazonEC2ContainerServiceAutoscaleRole` viene gradualmente eliminata. Ora è sostituita dal ruolo collegato al servizio Application Auto Scaling per Amazon ECS. Per ulteriori informazioni, consulta Ruoli collegati ai servizi per Application Auto Scaling nella Guida per l'utente di Application Auto Scaling.

Migrazione alla policy gestita da **AmazonECS_FullAccess**

La policy IAM gestita da AmazonEC2ContainerServiceFullAccess è stata gradualmente eliminata a partire dal 29 gennaio 2021, in risposta a un risultato di sicurezza con l'autorizzazione iam:passRole. Questa autorizzazione consente di accedere a tutte le risorse, incluse le credenziali ai ruoli nell'account. Ora che la policy è stata eliminata gradualmente, non sarà più possibile collegarla a nuovi gruppi, utenti o ruoli. Qualsiasi gruppo, utente o ruolo che dispone già della policy collegata può continuare a utilizzarla. Tuttavia, consigliamo di aggiornare i gruppi, gli utenti o i ruoli perché utilizzino invece la policy gestita da AmazonECS_FullAccess.

Le autorizzazioni concesse dalla policy AmazonECS_FullAccess includono l'elenco completo di autorizzazioni necessarie per utilizzare ECS come amministratore. Se attualmente utilizzi autorizzazioni concesse dalla AmazonEC2ContainerServiceFullAccess politica che non sono incluse nella AmazonECS_FullAccess politica, puoi aggiungerle a una dichiarazione politica in linea. Per ulteriori informazioni, consulta AWS politiche gestite per Amazon Elastic Container Service.

Utilizza la seguente procedura per determinare se disponi di gruppi, utenti o ruoli che attualmente utilizzano la policy IAM gestita da AmazonEC2ContainerServiceFullAccess. Quindi, aggiornali per scollegare la policy precedente e collega la policy AmazonECS_FullAccess.

Per aggiornare un gruppo, un utente o un ruolo per utilizzare la policy di FullAccess AmazoneCS_ ()AWS Management Console

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy e cerca e seleziona la policy AmazonEC2ContainerServiceFullAccess.
3. Seleziona la scheda Utilizzo della policy che riporta qualsiasi ruolo IAM che sta attualmente utilizzando questa policy.
4. Per ogni ruolo IAM che attualmente utilizza la AmazonEC2ContainerServiceFullAccess policy, seleziona il ruolo e segui i seguenti passaggi per scollegare la policy eliminata gradualmente e allegarla. AmazonECS_FullAccess

    a. Nella scheda Autorizzazioni, scegli la X accanto alla politica di AmazonEC2. ContainerService FullAccess

    b. Scegli Aggiungi autorizzazioni.

    c. Scegli Allega direttamente le politiche esistenti, cerca e seleziona la FullAccess politica AmazonECS_, quindi scegli Avanti: revisione.

d. Rivedi le modifiche e scegli Aggiungi autorizzazioni.

e. Ripeti questa procedura per ogni gruppo, utente o ruolo che utilizza la policy `AmazonEC2ContainerServiceFullAccess`.

Aggiornamento di un gruppo, un utente o un ruolo per utilizzare la policy **`AmazonECS_FullAccess`** (AWS CLI)

1. Utilizza il generate-service-last-accessed-detailscomando per generare un report che includa dettagli sull'ultima volta in cui è stata utilizzata la politica di eliminazione graduale.

```
aws iam generate-service-last-accessed-details \
     --arn arn:aws:iam::aws:policy/AmazonEC2ContainerServiceFullAccess
```

   Output di esempio:

```
{
    "JobId": "32bb1fb0-1ee0-b08e-3626-ae83EXAMPLE"
}
```

2. Utilizzate l'ID del lavoro dell'output precedente con il get-service-last-accessed-detailscomando per recuperare l'ultimo report del servizio a cui si accede. Questo report mostra l'Amazon Resource Name (ARN) delle entità IAM che hanno utilizzato per l'ultima volta la politica di eliminazione graduale.

```
aws iam get-service-last-accessed-details \
      --job-id 32bb1fb0-1ee0-b08e-3626-ae83EXAMPLE
```

3. Usa uno dei seguenti comandi per scollegare la policy `AmazonEC2ContainerServiceFullAccess` da un gruppo, un utente o un ruolo.

   - detach-group-policy
   - detach-role-policy
   - detach-user-policy

4. Usa uno dei seguenti comandi per collegare la policy `AmazonECS_FullAccess` a un gruppo, un utente o un ruolo.

   - attach-group-policy
   - attach-role-policy

- attach-user-policy

Uso di ruoli collegati ai servizi per Amazon ECS

Amazon Elastic Container Service utilizza AWS Identity and Access Management ruoli collegati ai servizi (IAM). Un ruolo collegato ai servizi è un tipo di ruolo IAM univoco collegato direttamente ad Amazon ECS. I ruoli collegati ai servizi sono definiti automaticamente da Amazon ECS e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi AWS per tuo conto.

Un ruolo collegato ai servizi semplifica la configurazione di Amazon ECS perché ti permette di evitare l'aggiunta manuale delle autorizzazioni necessarie. Amazon ECS definisce le autorizzazioni dei ruoli collegati ai servizi e, salvo diversamente definito, solo Amazon ECS può assumerne i ruoli. Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere collegata a nessun'altra entità IAM.

Per informazioni sugli altri servizi che supportano i ruoli collegati al servizio, consulta Servizi AWS che funzionano con IAM e cerca i servizi che riportano Sì nella colonna Ruoli collegati al servizio. 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

Il ruolo AWSServiceRoleForECS collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `ecs.amazonaws.com`

La politica di autorizzazione dei ruoli denominata AmazonECS ServiceRolePolicy consente ad Amazon ECS di completare le seguenti azioni sulle risorse specificate:

- Azione: quando utilizzi la modalità di rete `awsvpc` per le attività Amazon ECS, Amazon ECS gestisce il ciclo di vita delle interfacce di rete elastiche associate all'attività. Ciò include anche i tag che Amazon ECS aggiunge alle interfacce di rete elastiche.
- Azione: quando si utilizza un sistema di bilanciamento del carico con il servizio Amazon ECS, Amazon ECS gestisce la registrazione e l'annullamento della registrazione delle risorse con il sistema di bilanciamento del carico.

- Azione: quando si utilizza Amazon ECS Service Discovery, Amazon ECS gestisce la Route 53 e AWS Cloud Map le risorse necessarie per il funzionamento del service discovery.
- Operazione: quando si utilizza il dimensionamento automatico del servizio Amazon ECS, Amazon ECS gestisce le risorse con dimensionamento automatico richieste.
- Azione: Amazon ECS crea e gestisce CloudWatch allarmi e flussi di log che aiutano nel monitoraggio delle tue risorse Amazon ECS.
- Operazione: quando si utilizza Amazon ECS Exec, Amazon ECS gestisce le autorizzazioni necessarie per avviare le sessioni Amazon ECS Exec per le attività.
- Operazione: quando si utilizza Amazon ECS Service Connect, Amazon ECS gestisce le risorse AWS Cloud Map richieste per utilizzare la funzionalità.
- Operazione: quando si utilizzano provider di capacità Amazon ECS, Amazon ECS gestisce le autorizzazioni necessarie per modificare il gruppo con dimensionamento automatico e le relative istanze Amazon EC2.

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per Amazon ECS

Nella maggior parte dei casi, non devi creare manualmente un ruolo collegato ai servizi. Quando crei un cluster o crei o aggiorni un servizio nell' AWS Management Console AWS API AWS CLI, Amazon ECS crea il ruolo collegato al servizio per te. Se non vedi il AWSServiceRoleForECSruolo dopo aver creato un cluster, esegui le seguenti operazioni per risolvere il problema:

- Verifica e configura le autorizzazioni per consentire ad Amazon ECS di creare, modificare o eliminare un ruolo collegato ai servizi per tuo conto. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.
- Ritenta l'operazione di creazione del cluster o crea manualmente il ruolo collegato al servizio.

  Puoi utilizzare la console IAM per creare il ruolo AWSServiceRoleForECScollegato al servizio. Nella AWS CLI o nell' AWS API, crea un ruolo collegato al servizio con il nome del servizio. `ecs.amazonaws.com` Per ulteriori informazioni, consulta Creazione di un ruolo collegato al servizio nella Guida per l'utente di IAM.

> **Important**
>
> Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo.

Se elimini questo ruolo collegato al servizio, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un cluster oppure quando crei o aggiorni un servizio, Amazon ECS crea il ruolo collegato al servizio per tuo conto.

Se elimini questo ruolo collegato al servizio, puoi utilizzare lo stesso processo IAM per crearlo nuovamente.

Modifica di un ruolo collegato al servizio per Amazon ECS

Amazon ECS non consente di modificare il ruolo collegato al AWSServiceRoleForECS servizio. Dopo aver creato un ruolo collegato al servizio, non potrai modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per Amazon ECS

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, ti consigliamo di eliminare il ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

> **Note**
>
> Se il servizio Amazon ECS utilizza tale ruolo quando provi a eliminare le risorse, è possibile che l'eliminazione non riesca. In questo caso, attendi alcuni minuti e quindi ripeti l'operazione.

Per verificare se il ruolo collegato al servizio dispone di una sessione attiva

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Ruoli e scegli il AWSServiceRoleForECS nome (non la casella di controllo).

3. Nella pagina Riepilogo, seleziona Consulente accessi ed esamina l'attività recente per il ruolo collegato al servizio.

> **Note**
>
> Se non sei sicuro che Amazon ECS stia utilizzando il AWSServiceRoleForECS ruolo, puoi provare a eliminarlo. Se il servizio sta utilizzando il ruolo, l'eliminazione non andrà a buon fine e potrai visualizzare le regioni in cui il ruolo viene utilizzato. Se il ruolo è in uso, prima di poterlo eliminare dovrai attendere il termine della sessione. Non puoi revocare la sessione per un ruolo collegato al servizio.

Per rimuovere le risorse Amazon ECS utilizzate dal ruolo collegato al AWSServiceRoleForECS servizio

È necessario eliminare tutti i cluster Amazon ECS in tutte le AWS regioni prima di poter eliminare il AWSServiceRoleForECS ruolo.

1. Dimensione a 0 il conteggio di tutti i servizi Amazon ECS in tutte le regioni, quindi elimina i servizi. Per ulteriori informazioni, consulta Aggiornamento di un servizio Amazon ECS tramite la console e Eliminazione di un servizio Amazon ECS tramite la console.
2. Forza l'annullamento della registrazione di tutte le istanze di container da tutti i cluster in tutte le regioni. Per ulteriori informazioni, consulta Annullamento della registrazione di un'istanza di container Amazon ECS.
3. Elimina tutti i cluster Amazon ECS in tutte le regioni. Per ulteriori informazioni, consulta Eliminazione di un cluster Amazon ECS.

Eliminazione manuale del ruolo collegato al servizio con IAM

Utilizza la console IAM AWS CLI, o l' AWS API per eliminare il ruolo collegato al AWSServiceRoleForECS servizio. Per ulteriori informazioni, consulta Eliminazione del ruolo collegato al servizio nella Guida per l'utente di IAM.

Regioni supportate per i ruoli collegati ai servizi di Amazon ECS

Amazon ECS supporta l'utilizzo di ruoli collegati ai servizi in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta Regioni ed endpoint di AWS.

Ruoli IAM per Amazon ECS

Un ruolo IAM è un'identità IAM che puoi creare nel tuo account e che dispone di autorizzazioni specifiche. In Amazon ECS, puoi creare ruoli per concedere autorizzazioni a risorse Amazon ECS come contenitori o servizi.

I ruoli richiesti da Amazon ECS dipendono dalla definizione dell'attività, dal tipo di avvio e dalle funzionalità utilizzate. Utilizza la tabella seguente per determinare quali ruoli IAM sono necessari per Amazon ECS.

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di esecuzione di attività	Questo ruolo consente ad Amazon ECS di utilizzare altri AWS servizi per tuo conto.	La tua attività è ospitata su AWS Fargateo su istanze esterne e: • estrae un'immagine del contenitore da un repository privato Amazon ECR. • estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività. • invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. `awslogs`	Ruolo IAM di esecuzione di attività Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
		La tua attività è ospitata su una delle AWS Fargate istanze Amazon EC2 e: • utilizza l'autenti cazione del registro privato. • utilizza Runtime Monitoring. • la definizione dell'attività fa riferimento a dati sensibili utilizzan 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 istanze EC2 o alle istanze esterne di registrarsi nel cluster.	La tua attività è ospitata su istanze Amazon EC2 o su un'istanza esterna.	Ruolo IAM delle istanze di container Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo di Amazon ECS Anywhere	Questo ruolo consente alle istanze esterne di accedere AWS alle API.	La tua attività è ospitata su istanze esterne.	Ruolo IAM di Amazon ECS Anywhere
Ruolo di Amazon ECS CodeDeploy	Questo ruolo consente di CodeDeploy apportare aggiornamenti ai 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 di EventBridge apportare aggiornamenti ai tuoi servizi.	Utilizzi le EventBridge regole e gli obiettivi per pianificare le tue attività.	Ruolo EventBridge IAM di Amazon ECS

Ruolo	Definizione	Quando richiesto	Ulteriori informazioni
Ruolo dell'infrastruttura Amazon ECS	Questo ruolo consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei tuoi cluster.	• Vuoi collegare volumi Amazon EBS alle tue attività Amazon ECS di tipo Fargate o EC2. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività. • Desideri utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect.	Ruolo IAM dell'infrastruttura Amazon ECS

Le migliori pratiche per i ruoli IAM in Amazon ECS

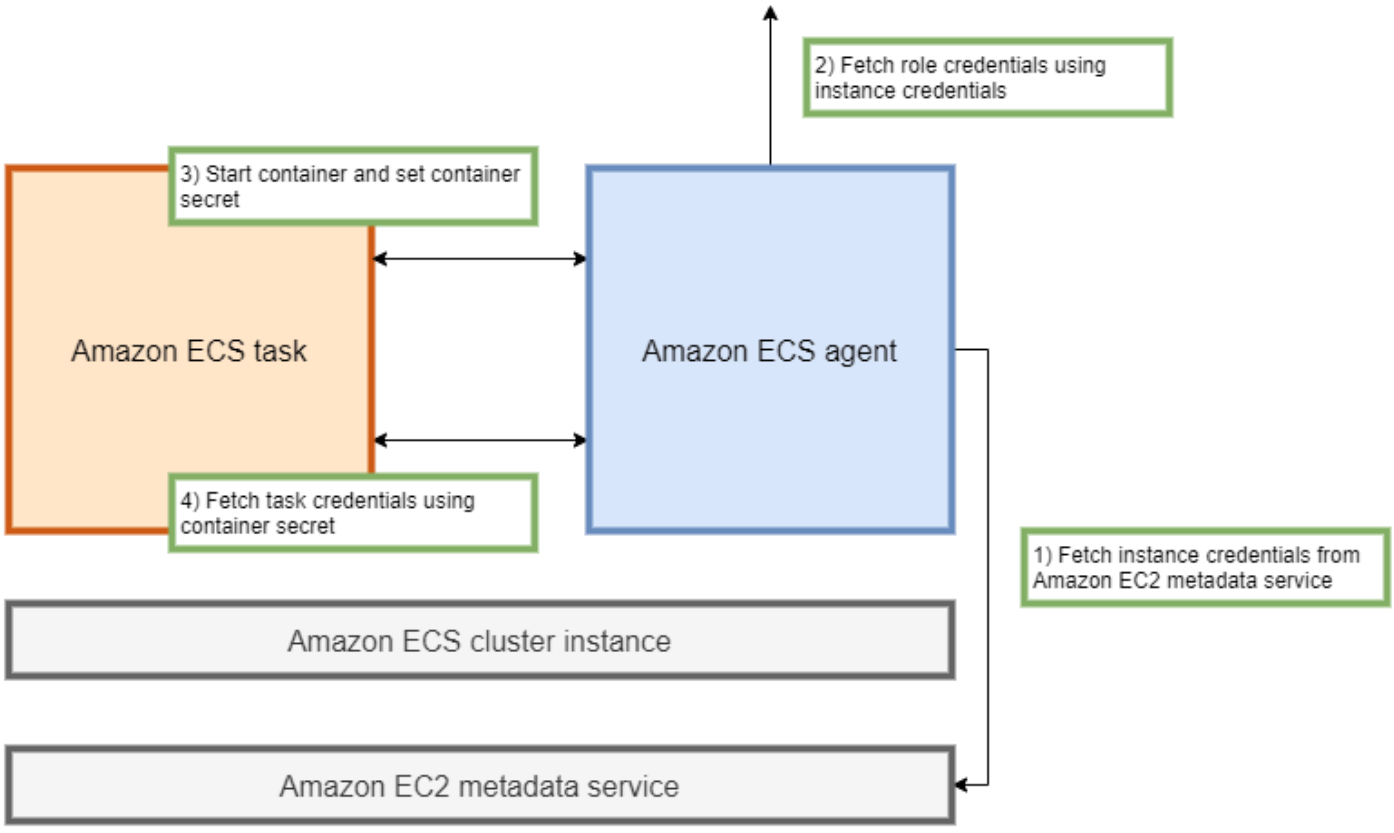
Ti consigliamo di assegnare un ruolo da svolgere. Il ruolo può essere diverso dal ruolo dell'istanza Amazon EC2 su cui è in esecuzione. L'assegnazione di un ruolo a ciascuna attività è conforme al principio dell'accesso con privilegi minimi e consente un controllo più granulare su operazioni e risorse.

Quando assegni i ruoli IAM per un'attività, devi utilizzare la seguente policy di attendibilità affinché ciascuna attività possa assumere un ruolo IAM diverso da quello utilizzato dall'istanza EC2. In tal modo, il processo non eredita il ruolo dell'istanza EC2.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
```

```
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Quando aggiungi un ruolo di attività a una definizione di attività, l'agente di container Amazon ECS crea in automatico un token con un ID di credenziali univoco (ad esempio `12345678-90ab-cdef-1234-567890abcdef`) per l'attività. Il token e le credenziali del ruolo vengono quindi aggiunti alla cache interna dell'agente. L'agente popola la variabile di ambiente `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` nel container con l'URI dell'ID delle credenziali (ad esempio `/v2/credentials/12345678-90ab-cdef-1234-567890abcdef`).



Puoi recuperare manualmente le credenziali temporanee del ruolo da un container aggiungendo la variabile di ambiente all'indirizzo IP dell'agente di container Amazon ECS ed eseguendo il comando `curl` sulla stringa risultante.

```
curl 169.254.170.2$AWS_CONTAINER_CREDENTIALS_RELATIVE_URI
```

L'output previsto è il seguente:

```
{
 "RoleArn": "arn:aws:iam::123456789012:role/SSMTaskRole-SSMFargateTaskIAMRole-
DASWWSF2WGD6",
 "AccessKeyId": "AKIAIOSFODNN7EXAMPLE",
 "SecretAccessKey": "wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
 "Token": "IQoJb3JpZ2luX2VjEEM/Example==",
 "Expiration": "2021-01-16T00:51:53Z"
}
```

Le versioni più recenti degli AWS SDK recuperano automaticamente queste credenziali dalla variabile di `AWS_CONTAINER_CREDENTIALS_RELATIVE_URI` ambiente quando effettuano chiamate API. AWS

L'output include una coppia di chiavi di accesso composta da un ID di chiave di accesso segreto e da una chiave segreta che l'applicazione utilizza 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 delle attività viene utilizzato per concedere all'agente container Amazon ECS l'autorizzazione a richiamare azioni AWS API specifiche per tuo conto. Ad esempio, quando si utilizza AWS Fargate, Fargate necessita di un ruolo IAM che le consenta di estrarre immagini da Amazon ECR e scrivere log in Logs. CloudWatch Un ruolo IAM è necessario anche quando un'attività fa riferimento a un segreto archiviato in AWS Secrets Manager, ad esempio un image pull secret.

> **Note**
>
> Se stai estraendo immagini in qualità di utente autenticato, è meno probabile che tu subisca l'impatto delle modifiche apportate ai limiti del tasso di estrazione di Docker Hub. Per ulteriori informazioni, consulta Autenticazione di registri privati per istanze di container.

Utilizzando Amazon ECR e Amazon ECR Public, puoi evitare i limiti imposti da Docker. Se estrai immagini da Amazon ECR, ciò contribuisce anche ad abbreviare i tempi di estrazione della rete e a ridurre le modifiche al trasferimento di dati quando il traffico esce dal tuo VPC.

Important

Quando utilizzi Fargate, devi autenticarti in un registro di immagini privato utilizzando `repositoryCredentials`. Non è possibile impostare le variabili di ambiente dell'agente di container Amazon ECS `ECS_ENGINE_AUTH_TYPE` o `ECS_ENGINE_AUTH_DATA` o modificare il file `ecs.config` per le attività in hosting su Fargate. Per ulteriori informazioni, consulta Autenticazione del registro privato per le attività.

Ruolo dell'istanza di container

L'agente di container Amazon ECS è un container che viene eseguito su ogni istanza Amazon EC2 in un cluster Amazon ECS. Viene inizializzato all'esterno di Amazon ECS utilizzando il comando `init` disponibile nel sistema operativo. Di conseguenza, non può ricevere autorizzazioni attraverso un ruolo di attività. Le autorizzazioni devono invece essere assegnate alle istanze Amazon EC2 su cui gli agenti vengono eseguiti. L'elenco delle azioni nella policy `AmazonEC2ContainerServiceforEC2Role` di esempio deve essere concesso a `ecsInstanceRole`. In caso contrario, le istanze non possono unirsi al cluster.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeTags",
                "ecs:CreateCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:DiscoverPollEndpoint",
                "ecs:Poll",
                "ecs:RegisterContainerInstance",
                "ecs:StartTelemetrySession",
                "ecs:UpdateContainerInstancesState",
                "ecs:Submit*",
                "ecr:GetAuthorizationToken",
```

```
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

In questa politica, le azioni `ecr` e le `logs` API consentono ai contenitori in esecuzione sulle tue istanze di estrarre immagini da Amazon ECR e scrivere log su Amazon. CloudWatch Le operazioni `ecs` consentono all'agente di effettuare e annullare la registrazione delle istanze e di comunicare con il piano di controllo Amazon ECS. Tra queste, l'operazione `ecs:CreateCluster` è facoltativa.

Ruoli collegati ai servizi

Puoi utilizzare il ruolo collegato al servizio per Amazon ECS in modo da concedere al servizio Amazon ECS l'autorizzazione a effettuare la chiamata ad altre API di servizio 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 attraveso il ruolo collegato al servizio. Tale ruolo viene creato per tuo conto la prima volta che utilizzi il servizio.

> **Note**
>
> Se elimini inavvertitamente il ruolo collegato al servizio, puoi ricrearlo. Per ricevere istruzioni, consulta Creazione di un ruolo collegato al servizio.

Consigli sui ruoli

Consigliamo di completare le seguenti operazioni quando configuri i ruoli e le policy IAM dell'attività.

Blocco dell'accesso ai metadati di Amazon EC2

Quando esegui le attività su istanze Amazon EC2, consigliamo vivamente di bloccare l'accesso ai metadati di Amazon EC2 per evitare che i container 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 metadati di Amazon EC2, 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 192.0.2.0/32 --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`.

Dovresti impostare la variabile `ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST` su `false` nel file `ecs-agent config` per impedire ai container in esecuzione all'interno della rete `host` di accedere ai metadati di Amazon EC2.

Usa la modalità **`awsvpc`** di rete

Utilizza la modalità di rete `awsvpc` per limitare il flusso di traffico tra diverse attività o tra le attività e altri servizi eseguiti all'interno di Amazon VPC. In questo modo si aggiunge un ulteriore livello di sicurezza. La modalità di rete `awsvpc` offre un isolamento della rete a livello di attività per le attività in esecuzione su Amazon EC2. È la modalità predefinita attivata AWS Fargate. È l'unica modalità di rete che puoi usare per assegnare un gruppo di sicurezza alle attività.

Uso di IAM Access Advisor per l'affinamento dei ruoli

Consigliamo di rimuovere tutte le operazioni che non sono mai state utilizzate o che non sono state utilizzate per un certo periodo di tempo. Ciò impedisce che si verifichino accessi indesiderati. A

tale scopo, esamina i risultati prodotti da IAM Access Advisor, quindi rimuovi le operazioni che non sono mai state utilizzate o che non sono state utilizzate di recente. Per farlo, completa la procedura seguente.

Utilizza il comando seguente per generare un report indicante le informazioni di accesso più recenti per la policy di riferimento:

```
aws iam generate-service-last-accessed-details --arn arn:aws:iam::123456789012:policy/
ExamplePolicy1
```

utilizza il valore `JobId` presente nell'output per eseguire il comando seguente. Dopo aver eseguito questa operazione, puoi visualizzare i risultati del report.

```
aws iam get-service-last-accessed-details --job-id 98a765b4-3cde-2101-2345-example678f9
```

Per ulteriori informazioni, consulta IAM Access Advisor.

Monitora le attività AWS CloudTrail sospette

Puoi monitorare qualsiasi attività AWS CloudTrail sospetta. La maggior parte delle chiamate AWS API viene registrata AWS CloudTrail come eventi. Vengono analizzate da AWS CloudTrail Insights e l'utente viene avvisato di eventuali comportamenti sospetti associati `write` alle chiamate API. Ciò potrebbe includere un picco nel volume delle chiamate. Questi avvisi includono diverse informazioni, come l'ora in cui si è verificata l'attività insolita e l'ARN dell'identità principale che ha contribuito alle API.

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": "AROA36C6WWEJ2YEXAMPLE:7e9894e088ad416eb5cab92afExample",
    "arn": "arn:aws:sts::123456789012:assumed-role/s3-write-go-bucket-
role/7e9894e088ad416eb5cab92afExample",
    ...
}
```

> **Note**
>
> Quando le attività che assumono un ruolo vengono eseguite su istanze di container Amazon EC2, l'agente del container Amazon ECS registra una richiesta nel log di audit 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. Per conoscere le autorizzazioni IAM necessarie per l'esecuzione dell'applicazione, consulta Ruolo IAM dell'attività Amazon ECS.

Di seguito sono elencati i casi di utilizzo comune per un ruolo IAM di esecuzione di attività:

- La tua attività è ospitata su AWS Fargateo su un'istanza esterna e:
  - estrae un'immagine del contenitore da un repository privato Amazon ECR.
  - estrae un'immagine del contenitore da un repository privato Amazon ECR in un account diverso dall'account che esegue l'attività.
  - invia i log dei contenitori a Logs utilizzando il driver di CloudWatch registro. `awslogs` Per ulteriori informazioni, consulta Invia i log di Amazon ECS a CloudWatch .
- Le tue attività sono ospitate su una delle AWS Fargate istanze Amazon EC2 e:
  - utilizza l'autenticazione del registro privato. Per ulteriori informazioni, consulta Autorizzazioni di autenticazione del registro privato.
  - utilizza Runtime Monitoring.
  - la definizione dell'attività fa riferimento a dati sensibili utilizzando i segreti di Secrets Manager o i parametri AWS Systems Manager Parameter Store. Per ulteriori informazioni, consulta Autorizzazioni Secrets Manager o Systems Manager.

> **Note**
>
> Il ruolo di esecuzione dei processi è supportato dall'agente del container di Amazon ECS versione 1.16.0 e successiva.

Amazon ECS fornisce la policy gestita denominata TaskExecutionRolePolicy AmazonECS che contiene le autorizzazioni richieste dai casi d'uso comuni sopra descritti. Per ulteriori informazioni, consulta TaskExecutionRolePolicyAmazonecs nella AWS Managed Policy Reference Guide. Potrebbe essere necessario aggiungere politiche in linea al ruolo di esecuzione delle attività per casi d'uso speciali

La console Amazon ECS crea un ruolo di esecuzione delle attività. Puoi allegare manualmente la policy IAM gestita per le attività per consentire ad Amazon ECS di aggiungere autorizzazioni per funzionalità e miglioramenti futuri man mano che vengono introdotti. Puoi utilizzare la ricerca della console IAM per cercare `ecsTaskExecutionRole` e verificare se il tuo account ha già il ruolo di esecuzione delle attività. Per ulteriori informazioni, consulta la ricerca nella console IAM nella guida per l'utente IAM.

Se richiami le immagini come utente autenticato, è meno probabile che tu sia influenzato dalle modifiche apportate ai limiti di pull rate di Docker Hub. Per ulteriori informazioni, consulta Autenticazione di registri privati per istanze di container.

Utilizzando Amazon ECR e Amazon ECR Public, puoi evitare i limiti imposti da Docker. Se estrai immagini da Amazon ECR, ciò contribuisce anche ad abbreviare i tempi di estrazione della rete e a ridurre le modifiche al trasferimento di dati quando il traffico esce dal tuo VPC.

Quando utilizzi Fargate, devi autenticarti in un registro di immagini privato utilizzando `repositoryCredentials`. Non è possibile impostare le variabili di ambiente dell'agente di container Amazon ECS `ECS_ENGINE_AUTH_TYPE` o `ECS_ENGINE_AUTH_DATA` o modificare il file `ecs.config` per le attività in hosting su Fargate. Per ulteriori informazioni, consulta Autenticazione del registro privato per le attività.

Creazione del ruolo di esecuzione attività

Se il tuo account non ha già un ruolo di esecuzione delle attività, utilizza i seguenti passaggi per creare il ruolo.

AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Seleziona Successivo.
6. Nella sezione Aggiungi autorizzazioni, cerca TaskExecutionRolePolicyAmazonecs, quindi seleziona la politica.
7. Seleziona Successivo.
8. Per il nome del ruolo, inserisci ecs Role. TaskExecution
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

AWS CLI

Sostituisci tutti gli *`input dell'utente`* con le tue informazioni.

1. Crea un file denominato `ecs-tasks-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Crea un ruolo IAM denominato `ecsTaskExecutionRole` utilizzando la policy di attendibilità creata nel passaggio precedente.

```
aws iam create-role \
      --role-name ecsTaskExecutionRole \
      --assume-role-policy-document file://ecs-tasks-trust-policy.json
```

3. Allega la `AmazonECSTaskExecutionRolePolicy` policy AWS gestita al `ecsTaskExecutionRole` ruolo.

```
aws iam attach-role-policy \
      --role-name ecsTaskExecutionRole \
      --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSTaskExecutionRolePolicy
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Usa le credenziali di Secrets Manager per accedere all'archivio privato delle immagini del contenitore	Autorizzazioni di autenticazione del registro privato
Trasmetti dati sensibili con Systems Manager o Secrets Manager	Autorizzazioni Secrets Manager o Systems Manager
Fai in modo che le attività di Fargate recuperino le immagini di Amazon ECR dagli endpoint dell'interfaccia	Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia
Ospita i file di configurazione in un bucket Amazon S3	Autorizzazioni per lo storage di file Amazon S3

Autorizzazioni di autenticazione del registro privato

Le autorizzazioni seguenti devono essere aggiunte manualmente come policy inline al ruolo per l'esecuzione di attività, per fornire l'accesso ai segreti che crei. Per ulteriori informazioni, consulta Aggiunta e rimozione delle policy IAM.

- `secretsmanager:GetSecretValue`
- `kms:Decrypt`: obbligatorio solo se la chiave utilizza una chiave KMS personalizzata e non quella di default. Il nome della risorsa Amazon (ARN) per la chiave personalizzata deve essere aggiunto come risorsa.

Di seguito viene riportata una policy inline che aggiunge le autorizzazioni.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:secret_name",
                "arn:aws:kms:<region>:<aws_account_id>:key/key_id"
            ]
        }
    ]
}
```

Autorizzazioni Secrets Manager o Systems Manager

L'autorizzazione per consentire all'agente del contenitore di estrarre le risorse necessarie AWS Systems Manager o Secrets Manager. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.

Uso di Secrets Manager

Per fornire l'accesso ai segreti di Gestione di segreti che crei, aggiungi manualmente la seguente autorizzazione al ruolo di esecuzione dell'attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente IAM.

- `secretsmanager:GetSecretValue`: obbligatorio se si fa riferimento a un segreto di Gestione dei segreti. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

La policy dell'esempio seguente aggiunge le autorizzazioni necessarie.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name"
      ]
    }
  ]
}
```

Uso di Systems Manager

> **Important**
>
> Per i processi che utilizzano il tipo di avvio EC2, devi utilizzare la variabile di configurazione dell'agente ECS `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` per utilizzare questa funzionalità. Puoi aggiungerlo al file `./etc/ecs/ecs.config` durante la creazione dell'istanza di container oppure aggiungerlo a un'istanza esistente e quindi riavviare l'agente ECS. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Per fornire l'accesso ai parametri di Archivio dei parametri Systems Manager che hai creato, aggiungi manualmente le autorizzazioni seguenti come policy al ruolo per l'esecuzione di attività. Per avere informazioni sulla gestione delle autorizzazioni, consulta Aggiunta e rimozione di autorizzazioni per identità IAM nella Guida per l'utente IAM.

- `ssm:GetParameters`: obbligatorio quando si fa riferimento a un parametro di Archivio dei parametri Systems Manager in una definizione di attività. Aggiunge l'autorizzazione per recuperare i parametri di Systems Manager.
- `secretsmanager:GetSecretValue`: obbligatorio quando si fa riferimento a un segreto di Gestione dei segreti direttamente o se il parametro di Archivio dei parametri Systems Manager

fa riferimento a un segreto di Gestione dei segreti in una definizione di attività. Aggiunge l'autorizzazione per recuperare il segreto da Secrets Manager.

- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave gestita personalizzata e non quella predefinita. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa. Aggiunge l'autorizzazione per decrittografare la chiave gestita dal cliente.

La seguente policy di esempio aggiunge le autorizzazioni necessarie:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:region:aws_account_id:parameter/parameter_name",
        "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name",
        "arn:aws:kms:region:aws_account_id:key/key_id"
      ]
    }
  ]
}
```

Fargate esegue le attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia

Quando avvii processi che usano il tipo di avvio Fargate che estraggono immagini da Amazon ECR quando Amazon ECR è configurato per utilizzare un endpoint VPC dell'interfaccia, puoi limitare l'accesso a un VPC o endpoint VPC specifico. Per fare questo, crea un ruolo di esecuzione dei processi per i processi che utilizzano chiavi di condizione IAM.

Utilizza le chiavi di condizione globali IAM seguenti per limitare l'accesso a un VPC o endpoint VPC specifico. Per ulteriori informazioni, consulta Chiavi di contesto delle condizioni globali AWS.

- `aws:SourceVpc`: limita l'accesso a un VPC specifico.
- `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:

> **Important**
>
> All'azione `ecr:GetAuthorizationToken` API non possono essere applicate le chiavi di `aws:sourceVpce` condizione `aws:sourceVpc` or perché la chiamata GetAuthorizationToken API passa attraverso l'interfaccia di rete elastica di proprietà di AWS Fargate anziché l'interfaccia di rete elastica dell'attività.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:GetAuthorizationToken",
                "logs:CreateLogStream",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpce": "vpce-xxxxxx",
                    "aws:sourceVpc": "vpc-xxxxx"
                }
            }
        }
    ]
}
```

Autorizzazioni per lo storage di file Amazon S3

Quando specifichi un file di configurazione ospitato in Amazon S3, il ruolo di esecuzione dell'attività deve includere l'`s3:GetObject`autorizzazione per il file di configurazione e l'`s3:GetBucketLocation`autorizzazione per il bucket Amazon S3 in cui si trova il file. Per ulteriori informazioni, consulta Specifica delle autorizzazioni in una policy nella Guida per l'utente di Amazon Simple Storage Service.

La seguente policy di esempio aggiunge le autorizzazioni necessarie per il recupero di un file da Amazon S3. Specifica il nome del bucket Amazon S3 e il nome del file di configurazione.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket/folder_name/config_file_name"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetBucketLocation"
      ],
      "Resource": [
        "arn:aws:s3:::examplebucket"
      ]
    }
  ]
}
```

Ruolo IAM dell'attività Amazon ECS

Le attività Amazon ECS possono avere un ruolo IAM associato. Le autorizzazioni concesse nel ruolo IAM sono assunte dai container in esecuzione nell'attività. Questo ruolo consente al codice dell'applicazione (sul contenitore) di utilizzare altri AWS servizi. Il ruolo dell'attività è necessario quando l'applicazione accede ad altri AWS servizi, come Amazon S3. Per conoscere le autorizzazioni

IAM necessarie ad Amazon ECS per estrarre le immagini di container ed eseguire l'attività, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

Di seguito sono riportati i vantaggi dell'utilizzo dei ruoli di attività:

- Isolamento delle credenziali: un container può recuperare solo le credenziali per il ruolo IAM stabilito nella definizione di attività a cui appartiene; non ha mai accesso alle credenziali destinate a un altro container appartenente a un'altra attività.
- Autorizzazione: le credenziali del ruolo IAM definite per altre attività non sono accessibili ai container privi di autorizzazione.
- Controllo: la registrazione degli accessi e degli eventi è disponibile CloudTrail per garantire un controllo retrospettivo. Le credenziali dell'attività hanno un contesto `taskArn` che è allegato alla sessione, quindi CloudTrail i registri mostrano quale attività utilizza quale ruolo.

> **Note**
>
> Quando specifichi un ruolo IAM per un'attività, gli SDK AWS CLI o altri SDK nei contenitori per quell'attività utilizzano esclusivamente le AWS credenziali fornite dal ruolo dell'attività e non ereditano più alcuna autorizzazione IAM da Amazon EC2 o dall'istanza esterna su cui sono in esecuzione.

Creazione del ruolo IAM dell'attività

Quando crei una policy IAM per le tue attività da utilizzare, la policy deve includere le autorizzazioni che desideri vengano assunte dai contenitori delle tue attività. Puoi utilizzare una policy AWS gestita esistente oppure puoi creare una policy personalizzata partendo da zero che soddisfi le tue esigenze specifiche. Per ulteriori informazioni, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

> **Important**
>
> Per le attività di Amazon ECS (per tutti i tipi di avvio), ti consigliamo di utilizzare il ruolo e la policy IAM per le tue attività. Queste credenziali consentono all'attività di effettuare richieste AWS API senza `sts:AssumeRole` dover chiamare per assumere lo stesso ruolo già associato all'attività. Se l'attività richiede che un ruolo assuma se stesso, devi creare una policy di attendibilità che consenta esplicitamente a tale ruolo di assumere se stesso. Per

ulteriori informazioni, consulta Modifica 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. Quindi, puoi collegare la tua politica IAM specifica al ruolo che fornisce ai contenitori del tuo task le autorizzazioni desiderate. Di seguito viene descritto come procedere.

In caso di più definizioni di attività o servizi che richiedono autorizzazioni IAM, è consigliabile creare un ruolo per ogni specifica definizione di attività o per ogni specifico servizio con il numero minimo di autorizzazioni necessarie per il funzionamento dell'attività, in modo da ridurre al minimo l'accesso fornito per ciascuna attività.

Per informazioni sull'endpoint di servizio per la tua regione, consulta Service endpoints nella Guida.Riferimenti generali di Amazon Web Services

Il ruolo dell'attività IAM deve avere una policy di attendibilità che specifica il servizio `ecs-tasks.amazonaws.com`. L'autorizzazione `sts:AssumeRole` consente ai processi di assumere un ruolo IAM diverso da quello utilizzato dall'istanza Amazon EC2. In questo modo, l'attività non eredita il ruolo associato all'istanza Amazon EC2. 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.

```
{
   "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"
            }
         }
      }
   ]
}
```

> **Important**
>
> Quando si crea il ruolo IAM dell'attività, si consiglia di utilizzare le chiavi `aws:SourceAccount` o `aws:SourceArn` condition nella relazione di fiducia o nella politica IAM associata al ruolo per definire ulteriormente l'ambito delle autorizzazioni ed evitare la confusa questione di sicurezza secondaria. L'uso della chiave di condizioni `aws:SourceArn` per specificare un cluster specifico non è correntemente supportato, è necessario utilizzare il carattere jolly per specificare tutti i cluster. Per saperne di più sul problema del confuso deputato e su come proteggere il tuo AWS account, consulta Il problema del vice confuso nella Guida per l'utente IAM.

Le seguenti procedure descrivono come creare una policy per recuperare oggetti da Amazon S3 con una policy di esempio. Sostituisci tutti gli *`input dell'utente`* con i tuoi valori.

AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

   ```
   {
      "Version":"2012-10-17",
   ```

```
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::my-task-secrets-bucket/*"
         ],
         "Condition":{
            "ArnLike":{
            "aws:SourceArn":"arn:aws:ecs:region:123456789012:*"
            },
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            }
         }
      }
   ]
}
```

6. Seleziona 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 gli *input dell'utente* con i tuoi valori.

1. Crea un file denominato `s3-policy.json`, con il seguente contenuto:

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "Action":[
            "s3:GetObject"
         ],
         "Resource":[
            "arn:aws:s3:::my-task-secrets-bucket/*"
         ],
         "Condition":{
            "ArnLike":{
            "aws:SourceArn":"arn:aws:ecs:region:123456789012:*"
            },
            "StringEquals":{
               "aws:SourceAccount":"123456789012"
            }
         }
      }
   ]
}
```

2. Usa il comando seguente per creare la policy IAM utilizzando il file di documento della policy JSON.

```
aws iam create-policy \
      --policy-name taskRolePolicy \
      --policy-document file://s3-policy.json
```

Le seguenti procedure descrivono come creare un ruolo IAM di task allegando una policy IAM creata dall'utente.

AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.

3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Service o use case, scegli Elastic Container Service, quindi scegli lo use case Elastic Container Service Task.
5. Seleziona Successivo.
6. Per Aggiungi autorizzazioni, cerca e scegli la policy che hai creato.
7. Seleziona 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

Sostituisci tutti gli *`input dell'utente`* con i tuoi valori.

1. Crea un file denominato `ecs-tasks-trust-policy.json` che contenga la policy di fiducia da utilizzare per il ruolo IAM dell'attività. Il file deve contenere quanto segue. Sostituisci l'identificatore della regione e specifica il numero di AWS account da utilizzare per l'avvio delle attività.

```
{
   "Version":"2012-10-17",
   "Statement":[
      {
         "Effect":"Allow",
         "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 utilizzando il seguente comando. Sostituisci *`task RolePolicy`* con il nome della policy che hai creato.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`taskRolePolicy`].Arn'
```

4. Allega la policy IAM che hai creato al `ecsTaskRole` ruolo. Sostituisci `policy-arn` con l'ARN della policy che hai creato.

```
aws iam attach-role-policy \
      --role-name ecsTaskRole \
      --policy-arn arn:aws:iam:111122223333:aws:policy/taskRolePolicy
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Usa ECS Exec	Autorizzazioni ECS Exec
Usa istanze EC2 (Windows e Linux)	Configurazione aggiuntiva delle istanze Amazon EC2
Usa istanze esterne	Configurazione aggiuntiva dell'istanza esterna
Usa istanze Windows EC2	Configurazione aggiuntiva dell'istanza Windows di Amazon EC2

Autorizzazioni ECS Exec

La funzionalità ECS Exec richiede un ruolo Task IAM per concedere ai contenitori le autorizzazioni necessarie per la comunicazione tra l'agente SSM gestito (agente) `execute-command` e il servizio SSM. È necessario aggiungere le seguenti autorizzazioni a un ruolo IAM del processo e includere il ruolo IAM del processo nella definizione di attività. Per ulteriori informazioni, consulta Aggiungere e rimuovere le politiche IAM nella Guida per l'utente IAM.

Utilizza la seguente policy per il ruolo IAM del processo e aggiungere le autorizzazioni di SSM richieste.

```
{
   "Version": "2012-10-17",
   "Statement": [
       {
       "Effect": "Allow",
       "Action": [
            "ssmmessages:CreateControlChannel",
            "ssmmessages:CreateDataChannel",
            "ssmmessages:OpenControlChannel",
            "ssmmessages:OpenDataChannel"
       ],
      "Resource": "*"
      }
   ]
}
```

Configurazione aggiuntiva delle istanze Amazon EC2

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 istanze Amazon EC2 richiedono almeno la versione `1.11.0` dell'agente contenitore per utilizzare il ruolo del task; tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente contenitore. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta Aggiornamento dell'agente del container Amazon ECS. Se utilizzi un'AMI ottimizzata per Amazon ECS, l'istanza richiede almeno una `1.11.0-1` parte del `ecs-init` pacchetto. Se le istanze di container sono avviate dall'AMI ottimizzata per Amazon ECS più recente, contengono le versioni richieste dell'agente del container e di `ecs-init`. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.

Se non utilizzi l'AMI ottimizzata per Amazon ECS per le tue istanze di container, aggiungi l'`--net=host`opzione al docker run comando che avvia l'agente e le seguenti variabili di configurazione dell'agente per la configurazione desiderata (per ulteriori informazioni, consulta): Configurazione dell'agente del container Amazon ECS

`ECS_ENABLE_TASK_IAM_ROLE=true`

Utilizza i ruoli IAM per le attività per i container con le modalità di rete `bridge` e `default`.

`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST=true`

Utilizza i ruoli IAM per le attività per i container con la modalità di rete `host`. Questa variabile è supportata sull'agente solo a partire dalla versione 1.12.0.

Per un esempio di comando di esecuzione, vedi Aggiornamento manuale dell'agente del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS). Dovrai inoltre impostare i seguenti comandi di rete sull'istanza del contenitore in modo che i contenitori utilizzati nelle tue attività possano recuperare le proprie credenziali: AWS

```
sudo sysctl -w net.ipv4.conf.all.route_localnet=1
sudo iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-
destination 127.0.0.1:51679
sudo iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --
to-ports 51679
```

Per fare in modo che le regole iptables vengano conservate dopo un riavvio, devi salvarle sull'istanza di container. Puoi salvare e ripristinare le regole iptables all'avvio tramite i comandi iptables-save e iptables-restore. Per ulteriori informazioni, consulta la documentazione specifica per il tuo sistema operativo.

Per impedire ai container in esecuzione come parte di un'attività che usano la modalità di rete `awsvpc` di accedere alle informazioni sulle credenziali fornite al profilo dell'istanza Amazon EC2 (continuando a concedere le autorizzazioni fornite dal ruolo dell'attività), imposta la variabile di configurazione dell'agente `ECS_AWSVPC_BLOCK_IMDS` su `true` nel file di configurazione dell'agente e quindi riavvia l'agente. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS.

Per impedire ai container in esecuzione come parte di un'attività che usano la modalità di rete `bridge` di accedere alle informazioni sulle credenziali fornite al profilo dell'istanza istanza Amazon EC2 (continuando a concedere le autorizzazioni fornite dal ruolo dell'attività), esegui il comando

iptables seguente nelle istanze Amazon EC2. Questo comando non influisce sui container nelle attività che usano la modalità di rete `host` o `awsvpc`. Per ulteriori informazioni, consulta Modalità di rete.

- ```
  sudo yum install -y iptables-services; sudo iptables --insert DOCKER-USER 1 --in-interface docker+ --destination 169.254.169.254/32 --jump DROP
  ```

  Per fare in modo che la regola iptables venga conservata dopo un riavvio, devi salvarla sull'istanza Amazon EC2. Quando utilizzi l'AMI ottimizzata per Amazon ECS, puoi utilizzare il comando riportato di seguito. Per gli altri sistemi operativi, consulta la relativa documentazione specifica.

  ```
  sudo iptables-save | sudo tee /etc/sysconfig/iptables && sudo systemctl enable --now iptables
  ```

Configurazione aggiuntiva dell'istanza esterna

Le istanze esterne richiedono almeno la versione `1.11.0` dell'agente contenitore per utilizzare i ruoli Task IAM; tuttavia, consigliamo di utilizzare la versione più recente dell'agente contenitore. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta Aggiornamento dell'agente del container Amazon ECS. Se utilizzi l'AMI ottimizzata per Amazon ECS, l'istanza deve disporre almeno della versione `1.11.0-1` del pacchetto `ecs-init`. Se le istanze di container sono avviate dall'AMI ottimizzata per Amazon ECS più recente, contengono le versioni richieste dell'agente del container e di `ecs-init`. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.

Se non utilizzi l'AMI ottimizzata per Amazon ECS per le tue istanze di container, aggiungi l'`--net=host`opzione al docker run comando che avvia l'agente e le seguenti variabili di configurazione dell'agente per la configurazione desiderata (per ulteriori informazioni, consulta): Configurazione dell'agente del container Amazon ECS

`ECS_ENABLE_TASK_IAM_ROLE=true`

Utilizza i ruoli IAM per le attività per i container con le modalità di rete `bridge` e `default`.

`ECS_ENABLE_TASK_IAM_ROLE_NETWORK_HOST=true`

Utilizza i ruoli IAM per le attività per i container con la modalità di rete `host`. Questa variabile è supportata sull'agente solo a partire dalla versione 1.12.0.
  ```

Per un esempio di comando di esecuzione, vedi Aggiornamento manuale dell'agente del container di Amazon ECS (per AMI non ottimizzate per Amazon ECS). Dovrai inoltre impostare i seguenti comandi di rete sull'istanza del contenitore in modo che i contenitori utilizzati nelle tue attività possano recuperare le proprie credenziali: AWS

```
sudo sysctl -w net.ipv4.conf.all.route_localnet=1
sudo iptables -t nat -A PREROUTING -p tcp -d 169.254.170.2 --dport 80 -j DNAT --to-
destination 127.0.0.1:51679
sudo iptables -t nat -A OUTPUT -d 169.254.170.2 -p tcp -m tcp --dport 80 -j REDIRECT --
to-ports 51679
```

Per fare in modo che le regole iptables vengano conservate dopo un riavvio, devi salvarle sull'istanza di container. Puoi salvare e ripristinare le regole iptables all'avvio tramite i comandi iptables-save e iptables-restore. Per ulteriori informazioni, consulta la documentazione specifica per il tuo sistema operativo.

Configurazione aggiuntiva dell'istanza Windows di Amazon EC2

> **Important**
>
> Questo vale solo per i contenitori Windows su EC2 che utilizzano ruoli di attività.

Il ruolo di attività con funzionalità Windows richiede una configurazione aggiuntiva su EC2.

- Quando avvii le istanze di container, devi configurare l'opzione `-EnableTaskIAMRole` nello script di dati utente dell'istanza di container. La `EnableTaskIAMRole` attiva la funzionalità ruoli IAM dei processi per i processi. Ad esempio:

  ```
  <powershell>
  Import-Module ECSTools
  Initialize-ECSAgent -Cluster 'windows' -EnableTaskIAMRole
  </powershell>
  ```

- Devi eseguire il bootstrap del container con i comandi di rete forniti in Script di bootstrap per contenitori Amazon ECS.
- Devi creare un ruolo IAM e una policy per i processi. Per ulteriori informazioni, consulta Creazione del ruolo IAM dell'attività.

- I ruoli IAM per il provider di credenziali dei processi utilizzano la porta 80 nell'istanza di container. Pertanto, se configuri i ruoli IAM per i processi nell'istanza di container, i container non potranno utilizzare la porta 80 come porta host in qualsiasi mappatura delle porte. Per esporre i container sulla porta 80, consigliamo di configurare un servizio che utilizzi il bilanciamento del carico. Puoi utilizzare la porta 80 sul load balancer. In questo modo, il traffico può essere instradato a un'altra porta host nelle istanze di container. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS.
- Se l'istanza di Windows viene riavviata, è necessario eliminare l'interfaccia proxy e inizializzare nuovamente l'agente del container di Amazon ECS per ripristinare il proxy delle credenziali.

Script di bootstrap per contenitori Amazon ECS

Prima che i container possano accedere al proxy di credenziali nell'istanza di container per ottenere le credenziali, devi eseguire il bootstrap del container con i comandi di rete richiesti. Il seguente script di esempio di codice deve essere eseguito nei container quando si avviano.

> **Note**
>
> Non è necessario eseguire questo script quando si utilizza la modalità di rete `awsvpc` su Windows.

Se esegui container Windows che includono Powershell, utilizza lo script seguente:

```
# Copyright Amazon.com Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may
# not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# or in the "license" file accompanying this file. This file is distributed
# on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
# express or implied. See the License for the specific language governing
# permissions and limitations under the License.

$gateway = (Get-NetRoute | Where { $_.DestinationPrefix -eq '0.0.0.0/0' } | Sort-Object
 RouteMetric | Select NextHop).NextHop
```

```
$ifIndex = (Get-NetAdapter -InterfaceDescription "Hyper-V Virtual Ethernet*" | Sort-
Object | Select ifIndex).ifIndex
New-NetRoute -DestinationPrefix 169.254.170.2/32 -InterfaceIndex $ifIndex -NextHop
 $gateway -PolicyStore ActiveStore # credentials API
New-NetRoute -DestinationPrefix 169.254.169.254/32 -InterfaceIndex $ifIndex -NextHop
 $gateway -PolicyStore ActiveStore # metadata API
```

Se esegui container Windows con solo Command shell, utilizza lo script seguente:

```
# Copyright Amazon.com Inc. or its affiliates. All Rights Reserved.
#
# Licensed under the Apache License, Version 2.0 (the "License"). You may
# not use this file except in compliance with the License. A copy of the
# License is located at
#
# http://aws.amazon.com/apache2.0/
#
# or in the "license" file accompanying this file. This file is distributed
# on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either
# express or implied. See the License for the specific language governing
# permissions and limitations under the License.

for /f "tokens=1" %i  in ('netsh interface ipv4 show interfaces ^| findstr /x /r
 ".*vEthernet.*"') do set interface=%i
for /f "tokens=3" %i  in ('netsh interface ipv4 show addresses %interface% ^| findstr /
x /r ".*Default.Gateway.*"') do set gateway=%i
netsh interface ipv4 add route prefix=169.254.170.2/32 interface="%interface%"
 nexthop="%gateway%" store=active # credentials API
netsh interface ipv4 add route prefix=169.254.169.254/32 interface="%interface%"
 nexthop="%gateway%" store=active # metadata API
```

Ruolo IAM delle istanze di container Amazon ECS

Le istanze di container Amazon ECS, incluse le istanze Amazon EC2 e le istanze esterne, eseguono l'agente del container di Amazon ECS e richiedono un ruolo IAM che consenta al servizio di rilevare 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 AWS CLI comandi.

> **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 all'agente contenitore Amazon ECS e al daemon Docker per chiamare AWS le API per tuo conto. Per ulteriori informazioni sulla policy gestita, consulta ContainerServiceforRuolo EC2 di Amazon EC2.

Amazon ECS supporta l'avvio di istanze di container con densità ENI aumentata mediante i tipi di istanze Amazon EC2 supportati. Quando utilizzi questa funzionalità, ti consigliamo di creare 2 ruoli di istanza di contenitore. Abilita l'impostazione `awsvpcTrunking` dell'account per un ruolo e utilizza quel ruolo per attività che richiedono il trunking ENI. Per informazioni sull'impostazione dell'`awsvpcTrunking`account, vedere. Accedi alle funzionalità di Amazon ECS con le impostazioni dell'account

Creare il ruolo dell'istanza del contenitore

> **Important**
>
> Se registri istanze esterne nel cluster, consulta Ruolo IAM di Amazon ECS Anywhere.

Puoi creare manualmente il ruolo e collegare la policy IAM gestita per le istanze di container in modo da consentire ad Amazon ECS di aggiungere le autorizzazioni per funzionalità e miglioramenti futuri man mano che vengono introdotti. Utilizza la seguente procedura per allegare la policy IAM gestita, se necessario.

AWS Management Console

Per creare il ruolo di servizio per Elastic Container Service (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per il servizio o il caso d'uso, scegli Elastic Container Service, quindi scegli il caso d'uso EC2 Role for Elastic Container Service.
5. Seleziona Successivo.
6. Nella sezione Politiche di autorizzazione, verifica che sia selezionata la politica EC2Role di Amazon EC2 ContainerServicefor.

   > **Important**
   >
   > La policy gestita da AmazonEC2 ContainerServicefor EC2Role deve essere allegata al ruolo IAM dell'istanza del contenitore, altrimenti riceverai un errore durante l'utilizzo del comando per creare cluster. AWS Management Console

7. Seleziona Successivo.
8. Per il nome del ruolo, inserisci ecs InstanceRole
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

AWS CLI

Sostituisci tutti gli *`input dell'utente`* con i tuoi valori.

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 comando seguente per creare il ruolo IAM dell'istanza utilizzando il documento di policy di fiducia.

```
aws iam create-role \
    --role-name ecsInstanceRole \
    --assume-role-policy-document file://instance-role-trust-policy.json
```

3. Creare un profilo dell'istanza denominato `ecsInstanceRole-profile` utilizzando il comando create-instance-profile.

```
aws iam create-instance-profile --instance-profile-name ecsInstanceRole-profile
```

Example response

```
{
    "InstanceProfile": {
        "InstanceProfileId": "AIPAJTLBPJLEGREXAMPLE",
        "Roles": [],
        "CreateDate": "2022-04-12T23:53:34.093Z",
        "InstanceProfileName": "ecsInstanceRole-profile",
        "Path": "/",
        "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole-
profile"
    }
}
```

4. Aggiungi il ruolo *ecsInstanceRole* al profilo dell'istanza *ecsInstanceRole-profile*.

```
aws iam add-role-to-instance-profile \
    --instance-profile-name ecsInstanceRole-profile \
    --role-name ecsInstanceRole
```

5. Allega la policy `AmazonEC2ContainerServiceRoleForEC2Role` gestita al ruolo utilizzando il comando seguente.

```
aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
```

```
    --role-name ecsInstanceRole
```

Dopo aver creato il ruolo, aggiungi ulteriori autorizzazioni al ruolo per le seguenti funzionalità.

Funzionalità	Autorizzazioni aggiuntive
Amazon ECR ha l'immagine del contenitore	Autorizzazioni Amazon ECR
Fai in modo che CloudWatch Logs monitori le istanze dei container	Monitoraggio delle autorizzazioni delle istanze dei container
Ospita i file di configurazione in un bucket Amazon S3	Accesso in sola lettura ad Amazon S3

Autorizzazioni Amazon ECR

Il ruolo dell'istanza di container Amazon ECS che usi con le tue istanze di container deve avere le seguenti autorizzazioni della policy IAM per Amazon ECR.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecr:BatchCheckLayerAvailability",
                "ecr:BatchGetImage",
                "ecr:GetDownloadUrlForLayer",
                "ecr:GetAuthorizationToken"
            ],
            "Resource": "*"
        }
    ]
}
```

Se utilizzi la policy gestita `AmazonEC2ContainerServiceforEC2Role` per le tue istanze di container, il tuo ruolo dispone già delle opportune autorizzazioni. Per verificare se il tuo ruolo supporta Amazon ECR, consulta Ruolo IAM per le istanze di container di Amazon ECS nella Guida per lo sviluppatore di Amazon Elastic Container.

Accesso in sola lettura ad Amazon S3

Archiviare le informazioni di configurazione in un bucket privato in Amazon S3 e concedere l'accesso in sola lettura al ruolo IAM dell'istanza di container è un modo sicuro e conveniente per permettere la configurazione delle istanze di container all'ora di avvio. Puoi archiviare una copia del tuo `ecs.config` file in un bucket privato, utilizzare i dati utente di Amazon EC2 per AWS CLI installarlo e quindi copiare le informazioni di configurazione `/etc/ecs/ecs.config` all'avvio dell'istanza.

Per ulteriori informazioni sulla creazione di un file `ecs.config`, sull'archiviazione di tale file in Amazon S3 e sull'avvio di istanze con questa configurazione, consulta Memorizzazione della configurazione dell'istanza del contenitore Amazon ECS in Amazon S3.

Puoi utilizzare il seguente AWS CLI comando per consentire l'accesso in sola lettura ad Amazon S3 per il tuo ruolo di istanza di contenitore. Sostituisci *`ecs InstanceRole`* con il nome del ruolo che hai creato.

```
aws iam attach-role-policy \
      --role-name ecsInstanceRole \
      --policy-arn arn:aws::iam::aws:policy/AmazonS3ReadOnlyAccess
```

Puoi anche utilizzare la console IAM per aggiungere Amazon S3 read-only access (`AmazonS3ReadOnlyAccess`) al tuo ruolo. Per ulteriori informazioni, consulta Modifica della politica di autorizzazione di un ruolo (console) nella Guida per l'utente.AWS Identity and Access Management

Monitoraggio delle autorizzazioni delle istanze dei container

Prima che le istanze del contenitore possano inviare dati di log a CloudWatch Logs, è necessario creare una policy IAM che consenta alle istanze del contenitore di utilizzare le API CloudWatch Logs e quindi allegare tale policy a. `ecsInstanceRole`

AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.

3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": ["arn:aws:logs:*:*:*"]
        }
    ]
}
```

6. Seleziona Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al ruolo dell'istanza del contenitore. Per informazioni su come allegare la policy al ruolo, consulta Modificare una politica di autorizzazioni di ruolo (console) nella Guida per l'AWS Identity and Access Management utente.

AWS CLI

1. Crea un file denominato `instance-cw-logs.json`, con il seguente contenuto:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": ["arn:aws:logs:*:*:*"]
        }
    ]
}
```

2. Utilizza il comando seguente per creare la policy IAM utilizzando il file del documento della policy JSON.

```
aws iam create-policy \
      --policy-name cwlogspolicy \
      --policy-document file://instance-cw-logs.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando. Sostituisci *cwlogspolicy con il nome della policy* che hai creato.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cwlogsppolicy`].Arn'
```

4. Usa il comando seguente per allegare la policy al ruolo IAM dell'istanza del contenitore utilizzando la policy ARN.

```
aws iam attach-role-policy \
      --role-name ecsInstanceRole \
      --policy-arn arn:aws:iam:111122223333:aws:policy/cwlogspolicy
```

Ruolo IAM di Amazon ECS Anywhere

Quando registri un server o una macchina virtuale (VM) locale nel tuo cluster, il server o la macchina virtuale richiede un ruolo IAM per comunicare con le API. AWS Devi creare questo ruolo IAM solo una volta per ogni account. AWS Tuttavia, questo ruolo IAM deve essere associato a ogni server o macchina virtuale registrato in un cluster. Questo ruolo è il `ECSAnywhereRole`. Puoi creare questo ruolo manualmente. In alternativa, Amazon ECS può creare il ruolo per tuo conto quando registri un'istanza esterna nella AWS Management Console. Puoi utilizzare la ricerca nella console IAM per cercare `ecsAnywhereRole` e vedere se il tuo account ha già il ruolo. Per ulteriori informazioni, consulta la ricerca nella console IAM nella guida per l'utente IAM.

AWS fornisce due policy IAM gestite che possono essere utilizzate durante la creazione del ruolo IAM ECS Anywhere, `AmazonEC2ContainerServiceforEC2Role` le `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 `ECSAnywhereRole` ruolo associato alla tua istanza esterna include anche le autorizzazioni necessarie per estrarre immagini da Amazon ECR, non è necessario che il ruolo di esecuzione dell'attività le includa.

Creazione del ruolo Amazon ECS Anywhere

Sostituisci tutti gli *`input dell'utente`* con le tue informazioni.

1. Crea un file locale denominato `ssm-trust-policy.json` con la seguente politica di attendibilità.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"Service": [
      "ssm.amazonaws.com"
    ]},
    "Action": "sts:AssumeRole"
  }
}
```

2. Crea il ruolo e allega la politica di fiducia utilizzando il AWS CLI comando seguente.

```
aws iam create-role --role-name ecsAnywhereRole --assume-role-policy-document
 file://ssm-trust-policy.json
```

3. Allega le politiche AWS gestite utilizzando il comando seguente.

```
aws iam attach-role-policy --role-name ecsAnywhereRole --policy-arn
 arn:aws:iam::aws:policy/AmazonSSMManagedInstanceCore
aws iam attach-role-policy --role-name ecsAnywhereRole --policy-arn
 arn:aws:iam::aws:policy/service-role/AmazonEC2ContainerServiceforEC2Role
```

Puoi anche utilizzare il flusso di lavoro delle policy di fiducia personalizzate IAM per creare il ruolo. Per ulteriori informazioni, consulta Creazione di un ruolo utilizzando politiche di fiducia personalizzate (console) nella Guida per l'utente IAM.

Ruolo IAM dell'infrastruttura Amazon ECS

Un ruolo IAM dell'infrastruttura Amazon ECS consente ad Amazon ECS di gestire le risorse dell'infrastruttura nei cluster per tuo conto e viene utilizzato quando:

- Vuoi collegare volumi Amazon EBS alle tue attività Amazon ECS di tipo Fargate o EC2. Il ruolo di infrastruttura consente ad Amazon ECS di gestire i volumi Amazon EBS per le tue attività.
- Desideri utilizzare Transport Layer Security (TLS) per crittografare il traffico tra i tuoi servizi Amazon ECS Service Connect.

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 `roleSessionNamefolefole` sarà. `ECSTaskVolumesForEBS` Se il ruolo viene utilizzato per crittografare il traffico tra i servizi Amazon ECS Service Connect, il CloudTrail registro `roleSessionName` sarà. `ECSServiceConnectForTLS` 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 gli allegati di volume e TLS. Per ulteriori informazioni, consulta AmazonECS Volumes e InfrastructureRole PolicyFor InfrastructureRole PolicyFor ServiceConnect TransportLayer AmazonECS Security nella Managed Policy Reference Guide.AWS

Creazione del ruolo dell'infrastruttura Amazon ECS

Sostituisci tutti gli *`input dell'utente`* con le tue informazioni.

1. Crea un file denominato `ecs-infrastructure-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAccessToECSForInfrastructureManagement",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Utilizzate il AWS CLI comando seguente per creare un ruolo denominato `ecsInfrastructureRole` utilizzando la politica di fiducia creata nel passaggio precedente.

```
aws iam create-role \
      --role-name ecsInfrastructureRole \
      --assume-role-policy-document file://ecs-infrastructure-trust-policy.json
```

3. A seconda del caso d'uso, collega il AWS gestore `AmazonECSInfrastructureRolePolicyForVolumes` o la

`AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity` policy al `ecsInfrastructureRole` ruolo.

```
aws iam attach-role-policy \
      --role-name ecsInfrastructureRole \
      --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSInfrastructureRolePolicyForVolumes
```

```
aws iam attach-role-policy \
      --role-name ecsInfrastructureRole \
      --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonECSInfrastructureRolePolicyForServiceConnectTransportLayerSecurity
```

Puoi anche utilizzare il flusso di lavoro Custom Trust Policy della console IAM per creare il ruolo. Per ulteriori informazioni, consulta Creazione di un ruolo utilizzando politiche di fiducia personalizzate (console) nella Guida per l'utente IAM.

Important

Se il ruolo dell'infrastruttura ECS viene utilizzato da Amazon ECS per gestire i volumi Amazon EBS collegati alle tue attività, verifica quanto segue prima di interrompere le attività che utilizzano volumi Amazon EBS.

- Il ruolo non viene eliminato.
- La policy di fiducia per il ruolo non viene modificata per rimuovere Amazon ECS access (`ecs.amazonaws.com`).
- La policy gestita `AmazonECSInfrastructureRolePolicyForVolumes` non viene rimossa. Se devi modificare le autorizzazioni del ruolo, conservale almeno `ec2:DetachVolume` e `ec2:DescribeVolumes` per l'eliminazione di un volume. `ec2:DeleteVolume`

L'eliminazione o la modifica del ruolo prima di interrompere le attività con volumi Amazon EBS collegati comporterà il blocco delle attività e la mancata eliminazione `DEPROVISIONING` dei volumi Amazon EBS associati. Amazon ECS riproverà automaticamente a intervalli regolari per interrompere l'attività ed eliminare il volume fino al ripristino delle autorizzazioni

necessarie. Puoi visualizzare lo stato degli allegati al volume di un'attività e il motivo dello stato associato utilizzando l'API. DescribeTasks

Dopo aver creato il file, devi concedere all'utente l'autorizzazione a passare il ruolo ad Amazon ECS.

Autorizzazione a trasferire il ruolo di infrastruttura ad Amazon ECS

Per utilizzare un ruolo IAM dell'infrastruttura ECS, devi concedere all'utente l'autorizzazione a passare il ruolo ad Amazon ECS. Allega la seguente `iam:PassRole` autorizzazione al tuo utente. Sostituisci *`ecs InfrastructureRole`* con il nome del ruolo di infrastruttura che hai creato.

```
{
    "Version": "2012-10-17",
    "Statement": [

        {
            "Action": "iam:PassRole",
            "Effect": "Allow",
            "Resource": ["arn:aws:iam::*:role/ecsInfrastructureRole"],
            "Condition": {
                "StringEquals": {"iam:PassedToService": "ecs.amazonaws.com"}
            }
        }
    ]
}
```

Per ulteriori informazioni *`iam:Passrole`* e aggiornamenti delle autorizzazioni per il tuo utente, consulta Concessione a un utente delle autorizzazioni per passare un ruolo a un AWS servizio e Modifica delle autorizzazioni per un utente IAM nella Guida per l'utente.AWS Identity and Access Management

Ruolo CodeDeploy IAM di Amazon ECS

Prima di poter utilizzare il tipo di distribuzione CodeDeploy blu/verde con Amazon ECS, il CodeDeploy servizio necessita delle autorizzazioni per aggiornare il servizio Amazon ECS per tuo conto. Queste autorizzazioni sono fornite dal ruolo IAM (). CodeDeploy `ecsCodeDeployRole`

> **Note**
>
> Gli utenti richiedono inoltre le autorizzazioni per l'uso CodeDeploy; queste autorizzazioni sono descritte in. Autorizzazioni IAM richieste

Sono disponibili due policy gestite. Per ulteriori informazioni, consulta una delle seguenti voci nella AWS Managed Policy Reference Guide:

- AWSCodeDeployRoleForECS- CodeDeploy autorizza ad aggiornare qualsiasi risorsa utilizzando l'azione associata.
- AWSCodeDeployRoleForECSLimited- concede autorizzazioni CodeDeploy più limitate.

Creare il ruolo CodeDeploy

Puoi utilizzare le seguenti procedure per creare un CodeDeploy ruolo per Amazon ECS.

AWS Management Console

Per creare il ruolo di servizio per CodeDeploy (console IAM)

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione della console IAM, scegliere Ruoli e quindi Crea ruolo.
3. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
4. Per Servizio o caso d'uso, scegli CodeDeploy, quindi scegli il caso d'uso CodeDeploy - ECS.
5. Seleziona Successivo.
6. Nella sezione Allega criteri di autorizzazione, assicurati che il AWSCodeDeployRoleForECScriterio sia selezionato.
7. Seleziona Successivo.
8. Per il nome del ruolo, inserisci ecs CodeDeploy Role.
9. Verificare il ruolo e quindi scegliere Create role (Crea ruolo).

AWS CLI

Sostituisci tutti gli *input dell'utente* con le tue informazioni.

1. Crea un file denominato `codedeploy-trust-policy.json` che contenga la policy di fiducia da utilizzare per il ruolo CodeDeploy IAM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": ["codedeploy.amazonaws.com"]
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Crea un ruolo IAM denominato `ecsCodedeployRole` utilizzando la policy di attendibilità creata nel passaggio precedente.

```
aws iam create-role \
      --role-name ecsCodedeployRole \
      --assume-role-policy-document file://codedeploy-trust-policy.json
```

3. Allega la policy `AWSCodeDeployRoleForECS` o la policy `AWSCodeDeployRoleForECSLimited` gestita al `ecsTaskRole` ruolo.

```
aws iam attach-role-policy \
      --role-name ecsCodedeployRole \
      --policy-arn arn:aws::iam::aws:policy/AWSCodeDeployRoleForECS
```

```
aws iam attach-role-policy \
      --role-name ecsCodedeployRole \
      --policy-arn arn:aws::iam::aws:policy/AWSCodeDeployRoleForECSLimited
```

Quando le attività del tuo servizio richiedono un ruolo di esecuzione 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 politica.

Autorizzazioni relative al ruolo di esecuzione delle 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 ruolo come politica. CodeDeploy Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS e Ruolo IAM dell'attività Amazon ECS. Quindi, si allega tale politica al ruolo CodeDeploy

Creare la policy

AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsCodeDeployRole>"]
        }
    ]
}
```

6. Seleziona Successivo.

> **Note**
>
> È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.
8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al CodeDeploy ruolo. Per informazioni su come allegare la politica al ruolo, consulta Modifica della politica di autorizzazione di un ruolo (console) nella Guida per l'AWS Identity and Access Management utente.

AWS CLI

Sostituisci tutti gli *`input dell'utente`* con le tue informazioni.

1. Crea un file denominato `blue-green-iam-passrole.json`, con il seguente contenuto:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsCodeDeployRole>"]
        }
    ]
}
```

2. Usa il comando seguente per creare la policy IAM utilizzando il file di documento della policy JSON.

```
aws iam create-policy \
```

```
      --policy-name cdTaskExecutionPolicy \
      --policy-document file://blue-green-iam-passrole.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`cdTaskExecutionPolicy`].Arn'
```

4. Usa il comando seguente per allegare la policy al ruolo CodeDeploy IAM.

```
aws iam attach-role-policy \
      --role-name ecsCodedeployRole \
      --policy-arn arn:aws:iam:111122223333:aws:policy/cdTaskExecutionPolicy
```

Ruolo EventBridge IAM di Amazon ECS

Prima di poter utilizzare le attività pianificate di Amazon ECS con EventBridge regole e obiettivi, il EventBridge servizio necessita delle autorizzazioni per eseguire le attività di Amazon ECS per tuo conto. Queste autorizzazioni sono fornite dal ruolo EventBridge IAM (). `ecsEventsRole`

Di seguito viene mostrata la policy `AmazonEC2ContainerServiceEventsRole`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["ecs:RunTask"],
            "Resource": ["*"]
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["*"],
            "Condition": {
                "StringLike": {"iam:PassedToService": "ecs-tasks.amazonaws.com"}
            }
        },
        {
            "Effect": "Allow",
            "Action": "ecs:TagResource",
            "Resource": "*",
```

```
            "Condition": {
                "StringEquals": {
                    "ecs:CreateAction": ["RunTask"]
                }
            }
        }
    ]
}
```

Se le attività pianificate richiedono l'uso del ruolo di esecuzione dell'attività, di un ruolo di attività o di un'eccezione di ruolo dell'attività, è necessario aggiungere `iam:PassRole` le autorizzazioni per ogni ruolo di esecuzione dell'attività, ruolo dell'attività o sovrascrittura del ruolo dell'attività al ruolo IAM. EventBridge Per ulteriori informazioni sul ruolo di esecuzione delle attività, consulta Ruolo IAM di esecuzione di attività Amazon ECS.

> **Note**
>
> Specifica l'ARN completo del ruolo di esecuzione delle attività o della sostituzione del ruolo attività.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
        }
    ]
}
```

Puoi scegliere di lasciare che AWS Management Console crei il EventBridge ruolo automaticamente quando configuri un'attività pianificata. Per ulteriori informazioni, consulta Utilizzo di Amazon EventBridge Scheduler per pianificare le attività di Amazon ECS .

Creazione del EventBridge ruolo

Sostituisci tutti gli *input dell'utente* con le tue informazioni.

1. Crea un file denominato `eventbridge-trust-policy.json` contenente la policy di attendibilità da utilizzare per il ruolo IAM. Il file deve contenere il testo seguente:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Utilizza il comando seguente per creare un ruolo IAM denominato `ecsEventsRole` utilizzando la policy di fiducia creata nel passaggio precedente.

```
aws iam create-role \
      --role-name ecsEventsRole \
      --assume-role-policy-document file://eventbridge-policy.json
```

3. Associa il AWS gestore `AmazonEC2ContainerServiceEventsRole` al `ecsEventsRole` ruolo utilizzando il comando seguente.

```
aws iam attach-role-policy \
      --role-name ecsEventsRole \
      --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceEventsRole
```

Puoi anche utilizzare il flusso di lavoro Custom Trust Policy della console IAM (https://console.aws.amazon.com/iam/) per creare il ruolo. Per ulteriori informazioni, consulta Creazione di un ruolo utilizzando policy di fiducia personalizzate (console) nella Guida per l'utente IAM.

Collegamento di una policy al ruolo **`ecsEventsRole`**

È possibile utilizzare le seguenti procedure per aggiungere le autorizzazioni per il ruolo di esecuzione dell'attività al ruolo EventBridge IAM.

AWS Management Console

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
        }
    ]
}
```

6. Seleziona Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.
7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.

8. Seleziona Crea policy per salvare la nuova policy.

Dopo aver creato la policy, associa la policy al EventBridge ruolo. Per informazioni su come allegare la politica al ruolo, consulta Modifica della politica di autorizzazione di un ruolo (console) nella Guida per l'AWS Identity and Access Management utente.

AWS CLI

Sostituisci tutti gli *input dell'utente* con le tue informazioni.

1. Crea un file denominato `ev-iam-passrole.json`, con il seguente contenuto:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": ["arn:aws:iam::<aws_account_id>:role/
<ecsTaskExecutionRole_or_TaskRole_name>"]
        }
    ]
}
```

2. Usa il AWS CLI comando seguente per creare la policy IAM utilizzando il file di documento della policy JSON.

```
aws iam create-policy \
      --policy-name eventsTaskExecutionPolicy \
      --policy-document file://ev-iam-passrole.json
```

3. Recupera l'ARN della policy IAM che hai creato utilizzando il seguente comando.

```
aws iam list-policies --scope Local --query 'Policies[?
PolicyName==`eventsTaskExecutionPolicy`].Arn'
```

4. Usa il comando seguente per collegare la policy al ruolo EventBridge IAM utilizzando la policy ARN.

```
aws iam attach-role-policy \
      --role-name ecsEventsRole \
```

```
    --policy-arn arn:aws:iam:111122223333:aws:policy/eventsTaskExecutionPolicy
```

Autorizzazioni necessarie per la console Amazon ECS

Seguendo la best practice per concedere il privilegio minimo, è possibile utilizzare la policy gestita di `AmazonECS_FullAccess` come modello per la creazione di policy personalizzate. In questo modo, è possibile rimuovere o aggiungere autorizzazioni da e verso le policy gestite in base ai requisiti specifici. Per ulteriori informazioni, consulta Dettagli dell'autorizzazione.

La console Amazon ECS è alimentata da AWS CloudFormation e richiede autorizzazioni IAM aggiuntive nei seguenti casi:

- Creazione di un cluster
- Creazione di un servizio
- Creazione di un provider di capacità

Puoi creare una policy per le autorizzazioni aggiuntive e quindi collegarle al ruolo IAM che utilizzi per accedere alla console. Per ulteriori informazioni, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Autorizzazioni necessarie per creare un cluster

Quando crei un cluster nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. AWS CloudFormation

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack AWS CloudFormation . Ciò è necessario quando si creano cluster Amazon ECS tramite la AWS Management Console e la successiva gestione di tali cluster.

La seguente policy contiene le AWS CloudFormation autorizzazioni richieste e limita le azioni alle risorse create nella console Amazon ECS.

```
{
  "Statement": [
      {
          "Effect": "Allow",
```

```
          "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack"
             ],
            "Resource": [
                "arn:*:cloudformation:*:*:stack/Infra-ECS-Cluster-*"
            ]
      }
   ]
}
```

Se non hai creato il ruolo dell'istanza di container Amazon ECS (`ecsInstanceRole`) e stai creando un cluster che utilizza istanze Amazon EC2, la console creerà il ruolo per tuo conto.

Inoltre, se si utilizzano i gruppi Auto Scaling, sono necessarie autorizzazioni aggiuntive in modo che la console possa aggiungere tag ai gruppi di auto scaling quando si utilizza la funzionalità di auto scaling del cluster.

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `autoscaling`: consente alla console di assegnare tag al gruppo con dimensionamento automatico Amazon EC2. Questa funzionalità è necessaria per la gestione dei gruppi con dimensionamento automatico Amazon EC2 quando si utilizza la funzione di dimensionamento automatico 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 principali possono anche elencare i profili dell'istanza disponibili per le istanze Amazon EC2.

La policy seguente contiene le autorizzazioni IAM necessarie e limita le operazioni al ruolo `ecsInstanceRole`.

Le autorizzazioni di dimensionamento automatico non sono limitate.

```
{
  "Statement": [
      {
          "Effect": "Allow",
            "Action": [
```

```
                "iam:AttachRolePolicy",
                "iam:CreateRole",
                "iam:CreateInstanceProfile",
                "iam:AddRoleToInstanceProfile",
                "iam:ListInstanceProfilesForRole",
                "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/ecsInstanceRole"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:CreateOrUpdateTags",
            "Resource": "*"
        }
    ]
}
```

Autorizzazioni necessarie per creare un provider di capacità

Quando crei un servizio nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. AWS CloudFormation Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack AWS CloudFormation . Ciò è necessario quando si creano provider di capacità Amazon ECS utilizzando la AWS Management Console e la successiva gestione di tali provider di capacità.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni alle risorse create nella console Amazon ECS.

```
{
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack"
            ],
            "Resource": [
                "arn:*:cloudformation:*:*:stack/Infra-ECS-CapacityProvider-*"
```

```
            ]
        }
    ]
}
```

Autorizzazioni necessarie per creare un servizio

Quando crei un servizio nella console, hai bisogno di autorizzazioni aggiuntive che ti concedano le autorizzazioni per gestire gli stack. AWS CloudFormation Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `cloudformation`: consente ai principali di creare e gestire stack AWS CloudFormation . Ciò è necessario quando si creano servizi Amazon ECS utilizzando la AWS Management Console e la successiva gestione di tali cluster.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni alle risorse create nella console Amazon ECS.

```
{
  "Statement": [
      {
          "Effect": "Allow",
          "Action": [
                "cloudformation:CreateStack",
                "cloudformation:DeleteStack",
                "cloudformation:DescribeStack*",
                "cloudformation:UpdateStack"
            ],
            "Resource": [
                "arn:*:cloudformation:*:*:stack/ECS-Console-V2-Service-*"
            ]
      }
   ]
}
```

Autorizzazioni per la creazione di ruoli IAM

Le operazioni seguenti richiedono autorizzazioni aggiuntive per completare l'operazione:

- Registrazione di un'istanza esterna: per ulteriori informazioni, consulta Ruolo IAM di Amazon ECS Anywhere

- Registrazione di una definizione di attività: per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS
- Creazione di una EventBridge regola da utilizzare per la pianificazione delle attività: per ulteriori informazioni, consulta Ruolo EventBridge IAM di Amazon ECS

Puoi aggiungere queste autorizzazioni creando un ruolo in IAM prima di utilizzarle nella console Amazon ECS. Se non crei i ruoli, la console Amazon ECS li crea per tuo conto.

Autorizzazioni necessarie per registrare un'istanza esterna in un cluster

Hai bisogno di autorizzazioni aggiuntive quando registri un'istanza esterna in un cluster e desideri creare un nuovo ruolo di istanza esterna (`escExternalInstanceRole`).

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `iam`: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate.
- `ssm`: consente ai principali di registrare l'istanza esterna con Systems Manager.

> **Note**
>
> Per scegliere un ruolo `escExternalInstanceRole` esistente, devi disporre delle autorizzazioni `iam:GetRole` e `iam:PassRole`.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `escExternalInstanceRole`.

```
{
  "Statement": [
      {
          "Effect": "Allow",
            "Action": [
              "iam:AttachRolePolicy",
              "iam:CreateRole",
              "iam:CreateInstanceProfile",
              "iam:AddRoleToInstanceProfile",
              "iam:ListInstanceProfilesForRole",
              "iam:GetRole"
            ],
```

```
            "Resource": "arn:aws:iam::*:role/escExternalInstanceRole"
        },
        {
            "Effect": "Allow",
            "Action": ["iam:PassRole","ssm:CreateActivation"],
            "Resource": "arn:aws:iam::*:role/escExternalInstanceRole"
        }
    ]
}
```

Autorizzazioni necessarie per la registrazione di una definizione di attività

Hai bisogno di autorizzazioni aggiuntive quando registri una definizione di attività e desideri creare un nuovo ruolo di esecuzione di attività (`ecsTaskExecutionRole`).

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `iam`: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate.

> **Note**
>
> Per scegliere un ruolo `ecsTaskExecutionRole` esistente, devi disporre dell'autorizzazione `iam:GetRole`.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `ecsTaskExecutionRole`.

```
{
  "Statement": [
      {
          "Effect": "Allow",
            "Action": [
              "iam:AttachRolePolicy",
              "iam:CreateRole",
              "iam:GetRole"
            ],
            "Resource": "arn:aws:iam::*:role/ecsTaskExecutionRole"
        }
    ]
}
```

Autorizzazioni necessarie per creare una EventBridge regola per le attività pianificate

Hai bisogno di autorizzazioni aggiuntive quando pianifichi un'attività e desideri creare un nuovo ruolo CloudWatch Events role (). `ecsEventsRole`

Le autorizzazioni aggiuntive elencate di seguito sono obbligatorie:

- `iam`: consente ai principali di creare ed elencare i ruoli IAM e le relative policy associate e di permettere ad Amazon ECS di trasmettere il ruolo ad altri servizi affinché lo assumano.

> **Note**
>
> Per scegliere un ruolo `ecsEventsRole` esistente, devi disporre delle autorizzazioni `iam:GetRole` e `iam:PassRole`.

La policy seguente contiene le autorizzazioni necessarie e limita le operazioni al ruolo `ecsEventsRole`.

```
{
  "Statement": [
      {
          "Effect": "Allow",
            "Action": [
              "iam:AttachRolePolicy",
              "iam:CreateRole",
              "iam:GetRole",
              "iam: PassRole"
            ],
            "Resource": "arn:aws:iam::*:role/ecsEventsRole"
        }
    ]
}
```

Autorizzazioni IAM richieste per la scalabilità automatica del servizio Amazon ECS

Service Auto Scaling è reso possibile da una combinazione delle API Amazon ECS e Application Auto Scaling. CloudWatch I servizi vengono creati e aggiornati con Amazon ECS, gli allarmi vengono creati e le politiche di scalabilità vengono create con CloudWatch Application Auto Scaling.

Oltre alle autorizzazioni IAM standard per la creazione e l'aggiornamento dei servizi, sono necessarie le seguenti autorizzazioni per interagire con le impostazioni di Service Auto Scaling, come illustrato nella seguente politica di esempio.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "application-autoscaling:*",
                "ecs:DescribeServices",
                "ecs:UpdateService",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory",
                "cloudwatch:DescribeAlarmsForMetric",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:ListMetrics",
                "cloudwatch:DisableAlarmActions",
                "cloudwatch:EnableAlarmActions",
                "iam:CreateServiceLinkedRole",
                "sns:CreateTopic",
                "sns:Subscribe",
                "sns:Get*",
                "sns:List*"
            ],
            "Resource": ["*"]
        }
    ]
}
```

Gli esempi di policy IAM Crea un esempio di servizio Amazon ECS e Esempio di aggiornamento del servizio Amazon ECS riportano le autorizzazioni necessarie per utilizzare Dimensionamento automatico nella AWS Management Console.

Il servizio Application Auto Scaling necessita inoltre dell'autorizzazione per descrivere i servizi e gli CloudWatch allarmi Amazon ECS e delle autorizzazioni per modificare il conteggio desiderato del servizio per tuo conto. Le `sns:` autorizzazioni sono per le notifiche CloudWatch inviate a un argomento di Amazon SNS quando viene superata una soglia. Se usi il dimensionamento automatico per i servizi Amazon ECS, viene creato un ruolo collegato ai servizi denominato

`AWSServiceRoleForApplicationAutoScaling_ECSService`. Questo ruolo collegato al servizio concede ad Application Auto Scaling l'autorizzazione per descrivere gli allarmi per le policy, per monitorare il conteggio corrente dei processi in esecuzione del servizio e per modificare il conteggio desiderato del servizio. Il ruolo originale di Amazon ECS gestito per Application Auto Scaling era `ecsAutoscaleRole`, ma non è più richiesto. Il ruolo collegato al servizio è il ruolo predefinito per Application Auto Scaling. Per ulteriori informazioni, consulta Ruoli collegati ai servizi per Application Auto Scaling nella Guida per l'utente di Application Auto Scaling.

Se hai creato il ruolo dell'istanza del contenitore Amazon ECS prima che i CloudWatch parametri fossero disponibili per Amazon ECS, potresti dover aggiungere l'autorizzazione. `ecs:StartTelemetrySession` Per ulteriori informazioni, consulta Considerazioni.

Concessione dell'autorizzazione all'assegnazione di tag alle risorse al momento della creazione

Le seguenti operazioni API di Amazon ECS per l'assegnazione di tag al momento della creazione consentono di specificare tag quando crei la risorsa. Se i tag sono specificati nell'azione di creazione della risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vengano assegnate le autorizzazioni corrette per creare tag.

- `CreateCapacityProvider`
- `CreateCluster`
- `CreateService`
- `CreateTaskSet`
- `RegisterContainerInstance`
- `RegisterTaskDefinition`
- `RunTask`
- `StartTask`

È possibile utilizzare i tag delle risorse per implementare il controllo basato sugli attributi (ABAC). Per ulteriori informazioni, consulta the section called “Controllo dell'accesso alle risorse Amazon ECS mediante i tag delle risorse” e Assegnazione di tag alle risorse.

Per consentire l'assegnazione di tag al momento della creazione, crea o modifica una policy che includa sia le autorizzazioni per utilizzare l'operazione che crea la risorsa, ad esempio `ecs:CreateCluster` o `ecs:RunTask`, sia l'operazione `ecs:TagResource`.

L'esempio seguente illustra una politica che consente agli utenti di creare cluster e aggiungere tag durante la creazione del cluster. Gli utenti non sono autorizzati ad applicare tag alle risorse esistenti (non possono chiamare l'operazione `ecs:TagResource` direttamente).

```
{
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
         "ecs:CreateCluster"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
         "ecs:TagResource"
      ],
      "Resource": "*",
      "Condition": {
         "StringEquals": {
                 "ecs:CreateAction": [
                     "CreateCluster",
                     "CreateCapacityProvider",
                     "CreateService",
                     "CreateTaskSet",
                     "RegisterContainerInstance",
                     "RegisterTaskDefinition",
                     "RunTask",
                     "StartTask"
                 ]
          }
       }
    }
  ]
}
```

L'operazione `ecs:TagResource` viene valutata solo se i tag vengono applicati durante l'operazione di creazione di risorse. Pertanto, un utente con le autorizzazioni per la creazione di una risorsa (presupponendo che non siano presenti condizioni di assegnazione di tag) non necessità delle autorizzazioni per utilizzare l'operazione `ecs:TagResource` se nella richiesta non viene specificato

alcun tag. Tuttavia, se l'utente tenta di creare una risorsa con tag, la richiesta ha esito negativo se non dispone delle autorizzazioni per utilizzare l'operazione `ecs:TagResource`.

Amazon ECS controlla l'accesso a tag specifici

È possibile utilizzare condizioni aggiuntive nell'elemento `Condition` delle policy IAM per controllare le chiavi dei tag e i valori che possono essere applicati alle risorse.

Le seguenti chiavi di condizione possono essere utilizzate con gli esempi nella sezione precedente:

- `aws:RequestTag`: indica che una chiave di tag o una chiave e un valore di tag sono presenti in una richiesta. Anche gli altri tag devono essere specificati nella richiesta.
  - Da utilizzare assieme all'operatore di condizione `StringEquals` per applicare una combinazione specifica di chiave e valore di tag, ad esempio per applicare il tag `cost-center=cc123`:

    ```
    "StringEquals": { "aws:RequestTag/cost-center": "cc123" }
    ```

  - Da utilizzare assieme all'operatore di condizione `StringLike` per applicare una chiave di tag specifica nella richiesta, ad esempio per applicare la chiave di tag `purpose`:

    ```
    "StringLike": { "aws:RequestTag/purpose": "*" }
    ```

- `aws:TagKeys`: applica le chiavi di tag utilizzate nella richiesta.
  - Da utilizzare assieme al modificatore `ForAllValues` per applicare chiavi di tag specifiche se vengono fornite nella richiesta (se i tag vengono specificati nella richiesta, solo le chiavi di tag specifiche sono consentite; non sono consentiti altri tag). Ad esempio, la chiave di tag `environment` o `cost-center` è consentita:

    ```
    "ForAllValues:StringEquals": { "aws:TagKeys": ["environment","cost-center"] }
    ```

  - Da utilizzare assieme al modificatore `ForAnyValue` per implementare la presenza di almeno una delle chiavi di tag specificate nella richiesta. Ad esempio, nella richiesta deve essere presente almeno una delle chiavi di tag `environment` o `webserver`:

    ```
    "ForAnyValue:StringEquals": { "aws:TagKeys": ["environment","webserver"] }
    ```

Queste chiavi di condizione possono essere applicate alle operazioni di creazione di risorse che supportano l'assegnazione di tag, nonché alle operazioni `ecs:TagResource`. Per sapere se un'operazione API di Amazon ECS supporta l'assegnazione di tag, consulta Operazioni, risorse e chiavi di condizione per Amazon ECS.

Per obbligare gli utenti a specificare i tag quando creano una risorsa, devi utilizzare la chiave di condizione `aws:RequestTag` o `aws:TagKeys` con il modificatore `ForAnyValue` nell'operazione di creazione delle risorse. L'operazione `ecs:TagResource` non viene valutata se un utente non specifica i tag per l'operazione di creazione delle risorse.

Per le condizioni, la chiave di condizione non fa distinzione tra maiuscole e minuscole, mentre il valore della condizione fa distinzione tra maiuscole e minuscole. Pertanto, per applicare la distinzione tra maiuscole e minuscole per una chiave di tag, utilizza la chiave di condizione `aws:TagKeys`, specificando la chiave di tag come valore nella condizione.

Per ulteriori informazioni sulle condizioni con più valori, consulta la sezione relativa alla creazione di una condizione per il test di valori di chiave multipli nella Guida per l'utente di 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 politiche di Amazon ECS .

Se consenti o neghi a un utente l'accesso a risorse in base ai tag, devi considerare esplicitamente di negare agli utenti la possibilità di aggiungere o rimuovere tali tag dalle stesse risorse. In caso contrario, un utente può eludere le restrizioni e ottenere l'accesso a una risorsa modificandone i tag.

Esempi di politiche di Amazon ECS

Puoi utilizzare le policy IAM per concedere agli utenti le autorizzazioni per visualizzare e utilizzare risorse specifiche nella console Amazon ECS. È possibile utilizzare le politiche di esempio nella sezione precedente; tuttavia, sono progettate per le richieste effettuate con AWS CLI o un AWS SDK.

Esempio: consenti agli utenti di eliminare un cluster Amazon ECS in base ai tag

La seguente policy consente agli utenti di eliminare i cluster quando il tag presenta una coppia chiave-valore "Scopo/Test".

```
{
  "Version": "2012-10-17",
  "Statement": [
       {
      "Action": [
        "ecs:DeleteCluster"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:ecs:region:account-id:cluster/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Testing"
        }
      }
    }
  ]
}
```

Risoluzione dei problemi di identità e accesso ad Amazon Elastic Container Service

Utilizza le informazioni seguenti per eseguire la diagnosi e risolvere i problemi comuni che possono verificarsi durante l'utilizzo di Amazon ECS e IAM.

Argomenti

- Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon ECS

Non dispongo dell'autorizzazione per eseguire un'operazione in Amazon ECS

Se ricevi un errore che indica che non sei autorizzato a eseguire un'operazione, le tue policy devono essere aggiornate per poter eseguire l'operazione.

L'errore di esempio seguente si verifica quando l'utente IAM `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia ma non dispone di autorizzazioni `ecs:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 ecs:GetWidget on resource: my-example-widget
```

In questo caso, la policy per l'utente `mateojackson` deve essere aggiornata per consentire l'accesso alla risorsa *`my-example-widget`* utilizzando l'azione `ecs:`*`GetWidget`*.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Non sono autorizzato a eseguire iam: PassRole

Se si riceve un errore che indica che non si è autorizzati a eseguire l'operazione `iam:PassRole`, è necessario aggiornare le policy per poter passare un ruolo ad Amazon ECS.

Alcuni Servizi AWS consentono di passare un ruolo esistente a quel servizio invece di creare un nuovo ruolo di servizio o un ruolo collegato al servizio. Per eseguire questa operazione, è necessario disporre delle autorizzazioni per trasmettere il ruolo al servizio.

L'errore di esempio seguente si verifica quando un utente IAM denominato `marymajor` cerca di utilizzare la console per eseguire un'operazione in Amazon ECS. Tuttavia, l'azione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne a me di accedere Account AWS alle mie risorse Amazon ECS

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se Amazon ECS supporta queste funzionalità, consulta Come funziona Amazon Elastic Container Service con IAM.
- Per scoprire come fornire l'accesso alle risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide. Account AWS
- 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 tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

Altre risorse per la risoluzione dei problemi

Le pagine seguenti forniscono informazioni sui codici di errore:

- Messaggi di errore delle attività interrotte da Amazon ECS
- Visualizzazione dei messaggi relativi agli eventi del servizio Amazon ECS

Best practice IAM per Amazon ECS

È possibile utilizzare AWS Identity and Access Management (IAM) per gestire e controllare l'accesso ai AWS servizi e alle risorse tramite politiche basate su regole per scopi di autenticazione e autorizzazione. Più specificamente, tramite questo servizio, puoi controllare l'accesso alle tue AWS risorse utilizzando policy applicate a utenti, gruppi o ruoli. Tra questi tre, gli utenti sono account che possono avere accesso alle tue risorse. Inoltre, un ruolo IAM è un insieme di autorizzazioni che possono essere assunte da un'identità autenticata, che non è associata a una particolare identità esterna a IAM. Per ulteriori informazioni, consulta la panoramica di Amazon ECS sulla gestione degli accessi: autorizzazioni e politiche.

Rispetto della policy di accesso con privilegio minimo

Crea policy mirate a consentire agli utenti di eseguire i processi prescritti. Ad esempio, se uno sviluppatore deve interrompere periodicamente un'attività, crea una policy che consenta solo quella particolare azione. L'esempio seguente consente solo a un utente di interrompere un'attività che appartiene a una particolare `task_family` in un cluster con un nome della risorsa Amazon (ARN) specifico. Il riferimento a un ARN in una condizione è anche un esempio di uso delle autorizzazioni a livello di risorsa. Puoi utilizzare le autorizzazioni a livello di risorsa per specificare la risorsa a cui desideri applicare un'azione.

> **Note**
>
> Quando fai riferimento a un ARN in una policy, utilizza il nuovo formato di ARN più lungo. Per ulteriori informazioni, consulta Nomi delle risorse Amazon (ARN) e ID nella Guida per gli sviluppatori di Amazon Elastic Container Service.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:StopTask"
      ],
      "Condition": {
        "ArnEquals": {
          "ecs:cluster": "arn:aws:ecs:region:account_id:cluster/cluster_name"
        }
```

```
      },
      "Resource": [
        "arn:aws:ecs:region:account_id:task-definition/task_family:*"
      ]
    }
  ]
}
```

Fai in modo che le risorse del cluster fungano da limite amministrativo

Le policy con un ambito troppo ristretto possono causare una proliferazione di ruoli e aumentare il sovraccarico amministrativo. Anziché creare ruoli limitati solo ad attività o servizi particolari, crea ruoli limitati ai cluster e utilizza il cluster come limite amministrativo principale.

Crea pipeline automatizzate per isolare gli utenti finali dall'API

Puoi limitare le azioni che gli utenti possono utilizzare creando pipeline che impacchettano e implementano in automatico le applicazioni sui cluster Amazon ECS. Ciò delega in modo efficace il processo di creazione, aggiornamento ed eliminazione delle attività alla pipeline. Per ulteriori informazioni, consulta Tutorial: distribuzione standard di Amazon ECS con CodePipeline nella Guida per l'AWS CodePipeline utente.

Uso di condizioni di policy per un ulteriore livello di sicurezza

Quando necessiti di un ulteriore livello di sicurezza, aggiungi una condizione alla tua policy. Ciò può essere utile se stai eseguendo un'operazione privilegiata o quando devi limitare l'insieme di azioni che possono essere eseguite su particolari risorse. La policy di esempio seguente richiede l'autorizzazione a più fattori quando si elimina un cluster.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecs:DeleteCluster"
      ],
      "Condition": {
        "Bool": {
          "aws:MultiFactorAuthPresent": "true"
        }
      },
```

```
    "Resource": ["*"]
    }
  ]
}
```

I tag applicati a un servizio vengono propagati a tutte le attività che fanno parte di tale servizio. Per questo motivo, puoi creare ruoli mirati alle risorse Amazon ECS con tag specifici. Nella 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`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:RunTask"
            ],
            "Resource": "arn:aws:ecs:*",
            "Condition": {
                "StringEquals": {"ecs:ResourceTag/Department": "Accounting"}
            }
        }
    ]
}
```

Verifica periodicamente l'accesso alle API

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 alle API di Amazon ECS 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 Amazon ECS in Riferimenti generali di Amazon Web Services.

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 policy di Amazon EC2 Auto Scaling. CloudWatch gli allarmi non richiamano azioni semplicemente perché si trovano in uno stato particolare; lo stato deve essere cambiato e mantenuto per un determinato numero di periodi. Per ulteriori informazioni, consulta Monitora Amazon ECS utilizzando CloudWatch .

Per i servizi con attività che utilizzano il tipo di avvio Fargate, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le attività del servizio in base a CloudWatch metriche, come l'utilizzo della CPU e della memoria. Per ulteriori informazioni, consulta Ridimensiona automaticamente il tuo servizio Amazon ECS.

Per i cluster con attività o servizi che utilizzano il tipo di lancio EC2, puoi utilizzare gli CloudWatch allarmi per ampliare e ridimensionare le istanze del contenitore in base CloudWatch a 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 container che utilizzano il tipo di avvio 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, consulta Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge questa guida e What Is Amazon CloudWatch Events? nella Guida per l'utente di Amazon CloudWatch Events.

AWS CloudTrail Registri

CloudTrail fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio in Amazon ECS. Utilizzando le informazioni raccolte da CloudTrail, puoi determinare la richiesta che è stata effettuata ad Amazon ECS, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi. Per ulteriori informazioni, consulta Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail.

AWS Trusted Advisor

Trusted Advisor attinge alle best practice apprese servendo centinaia di migliaia di AWS clienti. Trusted Advisor ispeziona l' AWS ambiente e quindi formula raccomandazioni quando esistono opportunità per risparmiare denaro, migliorare la disponibilità e le prestazioni del sistema o contribuire a colmare le lacune di sicurezza. Tutti i AWS clienti hanno accesso a cinque Trusted Advisor controlli. I clienti con un piano di supporto Business o Enterprise possono visualizzare tutti i Trusted Advisor controlli.

Per ulteriori informazioni, consulta AWS Trusted Advisor nella Guida per l'utente di AWS Support .

AWS Compute Optimizer

AWS Compute Optimizer è un servizio che analizza le metriche di configurazione e utilizzo delle risorse. AWS Il servizio segnala se le risorse sono ideali e genera suggerimenti di ottimizzazione per ridurre i costi e migliorare le prestazioni dei carichi di lavoro.

Per ulteriori informazioni, consulta AWS Compute Optimizer consigli per Amazon ECS.

Un'altra parte importante del monitoraggio di Amazon ECS consiste nel monitorare manualmente gli elementi che gli CloudWatch allarmi non coprono. Le dashboard della AWS console CloudWatch Trusted Advisor, e altre, forniscono una at-a-glance panoramica dello stato del tuo ambiente. AWS Ti consigliamo inoltre di controllare i file di log sulle tue istanze di container e i container nei tuoi processi.

Convalida della conformità per Amazon Elastic Container Service

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla AWS sicurezza e la conformità.
- Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services: questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee all'HIPAA.

  > **Note**
  >
  > Non Servizi AWS tutte sono idonee all'HIPAA. Per ulteriori informazioni, consulta la sezione Riferimenti sui servizi conformi ai requisiti HIPAA.

- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Evaluating Resources with Rules nella AWS Config Developer Guide: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty

può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.

- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

Best practice di conformità e sicurezza per Amazon ECS

La tua responsabilità di conformità durante l'utilizzo di Amazon ECS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e normative vigenti.

AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla sicurezza e la conformità. AWS
- Whitepaper sull'architettura per la sicurezza e la conformità HIPAA: questo white paper descrive in che modo le aziende possono utilizzare per creare applicazioni conformi allo standard HIPAA. AWS
- Servizi AWS coperti dal programma di conformità: questo elenco contiene i servizi AWS che rientrano nell'ambito di programmi per la conformità specifici. Per ulteriori informazioni, consulta Programmi per la conformità di AWS.

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
- Progettazione per l'ambito e la segmentazione PCI DSS su AWS

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 KMSÈ possibile implementare una rete di sovrapposizione (come VNS3 o Weave Net) per garantire la crittografia completa delle PHI trasferite tra container o per fornire un livello di crittografia ridondante. È inoltre necessario abilitare la registrazione completa e tutti i log dei container devono essere indirizzati ad Amazon. CloudWatch Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, consulta Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

AWS Security Hub

Utilizzalo AWS Security Hub per monitorare l'utilizzo di Amazon ECS in relazione alle best practice di sicurezza. Security Hub utilizza controlli per valutare le configurazioni delle risorse e gli standard di sicurezza per aiutarti a rispettare vari framework di conformità. Per ulteriori informazioni sull'uso della Centrale di sicurezza per valutare le risorse Amazon ECS, consulta Controlli di Amazon ECS nella Guida per l'utente di AWS 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 ed EC2 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 Runtime Monitoring nella Guida per l'utente. GuardDuty GuardDuty

Consigli sulla conformità

È consigliabile coinvolgere tempestivamente i responsabili dei programmi di conformità all'interno dell'azienda e utilizzare il modello di responsabilità condivisa di AWS per identificare la titolarità del controllo della conformità e garantire l'efficacia dei programmi di conformità pertinenti.

AWS Fargate Standard federale per l'elaborazione delle informazioni (FIPS-140)

Federal Information Processing Standard (FIPS). FIPS-140 è uno standard di sicurezza del governo degli Stati Uniti e del Canada che specifica i requisiti di sicurezza previsti per i moduli crittografici che proteggono informazioni sensibili. FIPS-140 definisce una serie di funzioni di crittografia convalidate che si possono utilizzare per la crittografia dei dati in transito e dei dati a riposo.

Quando attivi la conformità a FIPS-140, puoi eseguire carichi di lavoro su Fargate in maniera conforme a FIPS-140. Per ulteriori informazioni sulla conformità a FIPS-140, consulta Federal Information Processing Standard (FIPS) 140-2.

AWS Fargate Considerazioni sulla 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) .
- La conformità a FIPS-140 è disattivata per impostazione predefinita. Devi attivarla.
- Per la conformità a FIPS-140, le tue attività devono utilizzare la configurazione seguente:
  - `operatingSystemFamily` deve essere `LINUX`.
  - `cpuArchitecture` deve essere `X86_64`.

- La versione della piattaforma Fargate deve essere `1.4.0` o successiva.

Uso di FIPS su Fargate

Utilizza la procedura seguente per adottare la conformità a FIPS-140 su Fargate.

1. Attiva la conformità a FIPS-140. Per ulteriori informazioni, consulta the section called “AWS Fargate Conformità al Federal Information Processing Standard (FIPS-140)”.
2. Facoltativamente, puoi utilizzare ECS Exec per eseguire il comando seguente allo scopo di verificare lo stato di conformità a FIPS-140 di un cluster.

   Sostituisci *`my-cluster`* con il nome del tuo cluster.

   Un valore restituito corrispondente a "1" indica che stai utilizzando FIPS.

   ```
   aws ecs execute-command --cluster cluster-name \
       --interactive \
       --command "cat /proc/sys/crypto/fips_enabled"
   ```

Uso CloudTrail per il controllo FIPS-140 di Fargate

CloudTrail è attivata nel tuo AWS account quando crei l'account. Quando si verifica un'attività di API e console in Amazon ECS, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare eventi recenti nel tuo AWS account. Per ulteriori informazioni, consulta Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nel tuo AWS account, inclusi gli eventi per Amazon ECS, crea un percorso da CloudTrail utilizzare per distribuire i file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un trail nella console, il trail sarà valido in tutte le regioni. Il trail registra gli eventi di tutte le regioni della AWS partizione e consegna i file di log al bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei log. CloudTrail Per ulteriori informazioni, consulta the section called “Registra le chiamate API di Amazon ECS utilizzando AWS CloudTrail”.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'azione dell'`PutAccountSettingDefault`API:

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDAIV5AJI5LXF5EXAMPLE",
         "arn": "arn:aws:iam::123456789012:user/jdoe",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIPWIOFC3EXAMPLE",
    },
    "eventTime": "2023-03-01T21:45:18Z",
    "eventSource": "ecs.amazonaws.com",
    "eventName": "PutAccountSettingDefault",
    "awsRegion": "us-gov-east-1",
    "sourceIPAddress": "52.94.133.131",
    "userAgent": "aws-cli/2.9.8 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
ecs.put-account-setting",
    "requestParameters": {
        "name": "fargateFIPSMode",
        "value": "enabled"
    },
    "responseElements": {
        "setting": {
            "name": "fargateFIPSMode",
            "value": "enabled",
            "principalArn": "arn:aws:iam::123456789012:user/jdoe"
        }
    },
    "requestID": "acdc731e-e506-447c-965d-f5f75EXAMPLE",
    "eventID": "6afced68-75cd-4d44-8076-0beEXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "ecs-fips.us-gov-east-1.amazonaws.com"
    }
}
```

Sicurezza dell'infrastruttura in Amazon Elastic Container Service

In quanto servizio gestito, Amazon Elastic Container Service è protetto dalla sicurezza di rete AWS globale. Per informazioni sui servizi AWS di sicurezza e su come AWS protegge l'infrastruttura, consulta AWS Cloud Security. Per progettare il tuo AWS ambiente utilizzando le migliori pratiche per la sicurezza dell'infrastruttura, vedi Infrastructure Protection in Security Pillar AWS Well-Architected Framework.

Utilizzi chiamate API AWS pubblicate per accedere ad Amazon ECS attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi chiamare queste operazioni API da una qualsiasi posizione di rete. Amazon ECS supporta le policy di accesso basate sulle risorse, che possono includere limitazioni in base all'indirizzo IP di origine, quindi assicurati che le policy contengano l'indirizzo IP della posizione di rete. È inoltre possibile utilizzare le policy di Amazon ECS per controllare l'accesso da endpoint Amazon Virtual Private Cloud o VPC specifici. 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 alle API di Amazon ECS 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 Endpoints VPC nella Amazon VPC User Guide.

Considerazioni

Le considerazioni relative agli endpoint nelle regioni sono state introdotte a partire dal 23 dicembre 2023

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- È necessario disporre dei seguenti endpoint VPC specifici della regione:
  - `com.amazonaws.region.ecs-agent`
  - `com.amazonaws.region.ecs-telemetry`
  - `com.amazonaws.region.ecs`

  Ad esempio, la regione Canada occidentale (Calgary) (ca-west-1) necessita dei seguenti endpoint VPC:
  - `com.amazonaws.ca-west-1.ecs-agent`
  - `com.amazonaws.ca-west-1.ecs-telemetry`
  - `com.amazonaws.ca-west-1.ecs`
- Se utilizzi un modello per creare AWS risorse nella nuova regione e il modello è stato copiato da una regione introdotta prima del 23 dicembre 2023, a seconda della regione di origine, esegui una delle seguenti operazioni.

  Ad esempio, la regione del mittente è Stati Uniti orientali (Virginia settentrionale) (us-east-1). La regione da copiare è Canada occidentale (Calgary) (ca-west-1).

Configurazione	Azione
La regione da cui è stata copiata non è presente alcun endpoint VPC.	Crea tutti e tre gli endpoint VPC per la nuova regione (ad esempio,). `com.amazonaws.ca-west-1.ecs-agent`
La regione copiata contiene endpoint VPC specifici della regione.	a. Crea tutti e tre gli endpoint VPC per la nuova regione (ad esempio,). `com.amazo`

Configurazione	Azione	
	`naws.ca-west-1.ecs-agent` b. Elimina tutti e tre gli endpoint VPC per la regione di origine della copia (ad esempio,). `com.amazonaws.us-east-1.ecs-agent`	

Considerazioni sugli endpoint VPC di Amazon ECS per il tipo di avvio Fargate

Quando è presente un endpoint VPC per `ecr.dkr` e nello stesso VPC `ecr.api` in cui viene distribuita un'attività Fargate, utilizzerà l'endpoint VPC. Se non è presente un endpoint VPC, utilizzerà l'interfaccia Fargate.

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- Le attività che utilizzano il tipo di avvio Fargate non richiedono l'interfaccia VPC endpoint per Amazon ECS, ma potrebbero essere necessari endpoint VPC di interfaccia per Amazon ECR, Secrets Manager o Amazon Logs descritti nei punti seguenti. CloudWatch
  - Per consentire ai processi di estrarre immagini private da Amazon ECR, è necessario creare gli endpoint VPC di interfaccia per Amazon ECR. Per ulteriori informazioni, consulta Endpoint VPC di interfaccia (AWS PrivateLink) nella Guida per l'utente di Amazon Elastic Container Registry.

    Se il tuo VPC non dispone di un gateway Internet, devi creare l'endpoint gateway per Amazon S3. Per ulteriori informazioni, consulta Create the Amazon S3 gateway endpoint (Creazione dell'endpoint gateway per Amazon S3 nella Guida per l'utente di Amazon Elastic Container Registry. Gli endpoint dell'interfaccia per Amazon S3 non possono essere utilizzati con Amazon ECR.

    > **Important**
    >
    > Se configuri Amazon ECR per utilizzare un endpoint VPC di interfaccia, puoi creare un ruolo di esecuzione delle attività che include chiavi di condizione per limitare l'accesso a un VPC o endpoint VPC specifico. Per ulteriori informazioni, consulta Fargate esegue le

attività di estrazione delle immagini Amazon ECR tramite le autorizzazioni degli endpoint dell'interfaccia.

- Per consentire ai processi di estrarre dati sensibili da Secrets Manager, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di Secrets Manager con endpoint VPC nella Guida per l'utente di AWS Secrets Manager .
- Se il tuo VPC non dispone di un gateway Internet e le tue attività utilizzano il driver di `awslogs` registro per inviare le informazioni di registro ai CloudWatch registri, devi creare un endpoint VPC di interfaccia per i registri. CloudWatch Per ulteriori informazioni, consulta Using CloudWatch Logs with Interface VPC Endpoints nella CloudWatch Amazon Logs User Guide.

- Gli endpoint VPC attualmente non supportano le richieste inter-Regionali. Assicurati di creare l'endpoint nella stessa regione in cui prevedi di inviare le chiamate API ad Amazon ECS. Supponiamo ad esempio di voler eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale). Dovrai ovviamente creare l'endpoint VPC di Amazon ECS nella Regione Stati Uniti orientali (Virginia settentrionale). Un endpoint VPC di Amazon ECS creato in qualsiasi altra Regione non può eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale).
- Gli endpoint VPC supportano solo il DNS fornito da Amazon tramite Amazon Route 53. Se si desidera utilizzare il proprio DNS, è possibile usare l'inoltro condizionale sul DNS. Per ulteriori informazioni, consulta Set opzioni DHCP nella Guida per l'utente di Amazon VPC.
- Il gruppo di sicurezza collegato all'endpoint VPC deve consentire le connessioni in entrata sulla porta TCP 443 dalla sottorete privata del VPC.
- La gestione Service Connect del proxy Envoy utilizza l'endpoint VPC `com.amazonaws.region.ecs-agent`. Quando non utilizzi gli endpoint VPC, la gestione Service Connect del proxy Envoy utilizza l'endpoint `ecs-sc` in quella determinata Regione. Per un elenco degli endpoint Amazon ECS in ciascuna Regione, consulta Endpoint e quote Amazon ECS.

Considerazioni sugli endpoint VPC di Amazon ECS per il tipo di avvio EC2

Prima di configurare gli endpoint VPC di interfaccia per Amazon ECS, tieni presente le considerazioni riportate di seguito:

- Le attività che utilizzano il tipo di avvio EC2 richiedono che le istanze di container su cui sono avviate eseguano la versione `1.25.1` o successiva dell'agente di container Amazon ECS. Per ulteriori informazioni, consulta Gestione delle istanze di container Amazon ECS Linux.

- Per consentire ai processi di estrarre dati sensibili da Secrets Manager, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di Secrets Manager con endpoint VPC nella Guida per l'utente di AWS Secrets Manager .
- Se il tuo VPC non dispone di un gateway Internet e le tue attività utilizzano il driver di `awslogs` registro per inviare le informazioni di registro ai CloudWatch registri, devi creare un endpoint VPC di interfaccia per i registri. CloudWatch Per ulteriori informazioni, consulta Using CloudWatch Logs with Interface VPC Endpoints nella CloudWatch Amazon Logs User Guide.
- Gli endpoint VPC attualmente non supportano le richieste inter-Regionali. Assicurati di creare l'endpoint nella stessa regione in cui prevedi di inviare le chiamate API ad Amazon ECS. Supponiamo ad esempio di voler eseguire attività nella Regione Stati Uniti orientali (Virginia settentrionale). Dovrai ovviamente creare l'endpoint VPC di Amazon ECS nella Regione Stati Uniti orientali (Virginia settentrionale). Un endpoint VPC Amazon ECS creato in qualsiasi altra regione non può eseguire attività negli Stati Uniti orientali (Virginia settentrionale).
- Gli endpoint VPC supportano solo il DNS fornito da Amazon tramite Amazon Route 53. Se si desidera utilizzare il proprio DNS, è possibile usare l'inoltro condizionale sul DNS. Per ulteriori informazioni, consulta Set opzioni DHCP nella Guida per l'utente di Amazon VPC.
- Il gruppo di sicurezza collegato all'endpoint VPC deve consentire le connessioni in entrata sulla porta TCP 443 dalla sottorete privata del VPC.

Creazione di endpoint VPC per Amazon ECS

Per creare gli endpoint VPC per il servizio Amazon ECS, utilizza la procedura Creazione di un endpoint di interfaccia descritta nella Guida per l'utente di Amazon VPC. 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.

- `com.amazonaws.`*`region`*`.ecs-agent`
- `com.amazonaws.`*`region`*`.ecs-telemetry`
- `com.amazonaws.`*`region`*`.ecs`

> **Note**
>
> *`region`* rappresenta l'identificatore di regione per una regione AWS supportata da Amazon ECS, ad esempio `us-east-2` per la regione Stati Uniti orientali (Ohio).

L'`ecs-agent`endpoint utilizza l'`ecs:poll`API e l'`ecs-telemetry`endpoint utilizza l'API and. `ecs:poll` `ecs:StartTelemetrySession`

Se sono presenti processi che utilizzano il tipo di Avvio EC2, dopo aver creato gli endpoint VPC, ogni istanza di container deve selezionare 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":[
    {
      "Effect": "Allow",
      "Action": [
        "ecs:CreateCluster",
        "ecs:ListClusters"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

Best practice per la sicurezza di attività e container di Amazon ECS

L'immagine di container deve essere considerata la prima linea di difesa contro un attacco. Un'immagine non sicura e mal costruita può consentire a un utente malintenzionato di sfuggire ai limiti del container e accedere all'host. Per ridurre questo rischio, effettua le seguenti operazioni.

Quando configuri le attività e i container, tieni in considerazione i suggerimenti elencati di seguito.

Creazione di immagini di base o uso di immagini distroless

Inizia rimuovendo tutti i file binari estranei dall'immagine di container. Se stai utilizzando un'immagine sconosciuta tratta da Amazon ECR Public Gallery, ispezionala per fare riferimento al contenuto di ciascuno dei livelli del container. A tale scopo, puoi utilizzare un'applicazione come Dive.

In alternativa, puoi utilizzare immagini distroless che includono solo l'applicazione e le relative dipendenze di runtime. Non contengono gestori di pacchetti o shell (interprete di comandi). Le immagini distroless migliorano il "rapporto segnale/rumore degli scanner e riducono l'onere di stabilire la provenienza in base alle proprie esigenze". Per ulteriori informazioni, consulta la GitHub documentazione su distroless.

Docker dispone di un meccanismo per creare immagini a partire da un'immagine di base e riservata, nota come scratch. Per ulteriori informazioni, consulta Creazione di una semplice immagine genitore

con scratch nella documentazione di Docker. Con linguaggi come Go, puoi creare un file binario collegato statico e farvi riferimento nel Dockerfile. L'esempio seguente mostra come si può ottenere questo risultato.

```
############################
# STEP 1 build executable binary
############################
FROM golang:alpine AS builder
# Install git.
# Git is required for fetching the dependencies.
RUN apk update && apk add --no-cache git
WORKDIR $GOPATH/src/mypackage/myapp/
COPY . .
# Fetch dependencies.
# Using go get.
RUN go get -d -v
# Build the binary.
RUN go build -o /go/bin/hello
############################
# STEP 2 build a small image
############################
FROM scratch
# Copy our static executable.
COPY --from=builder /go/bin/hello /go/bin/hello
# Run the hello binary.
ENTRYPOINT ["/go/bin/hello"]
This creates a container image that consists of your application and nothing else,
 making it extremely secure.
```

L'esempio precedente illustra inoltre una build a più fasi. Questi tipi di build sono interessanti dal punto di vista della sicurezza, perché puoi utilizzarli per ridurre al minimo le dimensioni dell'immagine finale inviata al registro di container. Le immagini di container prive di strumenti di compilazione e altri file binari estranei migliorano il livello di sicurezza, riducendo la superficie di attacco dell'immagine. Per ulteriori informazioni sulle build a più fasi, consulta Creazione di build a più fasi.

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.

Le immagini archiviate in Amazon ECR possono essere analizzate mediante invio oppure on demand (una volta ogni 24 ore). La scansione di base di Amazon ECR utilizza Clair, una soluzione di scansione immagini open source. La scansione avanzata di Amazon ECR utilizza Amazon Inspector. Dopo la scansione di un'immagine, i risultati vengono registrati nel flusso di eventi Amazon ECR in Amazon. EventBridge Puoi anche visualizzare i risultati di una scansione dalla console Amazon ECR o chiamando l'DescribeImageScanFindingsAPI. Le immagini con una vulnerabilità `HIGH` o `CRITICAL` devono essere eliminate o ricreate. Un'immagine implementata che sviluppa una vulnerabilità deve essere sostituita il prima possibile.

Docker Desktop Edge versione 2.3.6.0 o successiva può effettuare la scansione delle immagini locali. Le scansioni sono gestite da Snyk, un servizio di sicurezza per applicazioni. Quando vengono individuate vulnerabilità, Snyk identifica i livelli e le dipendenze con la vulnerabilità nel Dockerfile. Suggerisce inoltre alternative sicure, come l'uso di un'immagine di base più snella con meno vulnerabilità o l'aggiornamento di un particolare pacchetto a una versione più recente. Utilizzando la scansione di Docker, gli sviluppatori possono risolvere potenziali problemi di sicurezza prima di inviare le immagini al registro.

- L'articolo Automatizzazione della conformità delle immagini con Amazon ECR e AWS Security Hub spiega come far emergere le informazioni sulle vulnerabilità da Amazon ECR in AWS Security Hub e automatizzare la correzione bloccando l'accesso alle immagini vulnerabili.

Rimozione delle autorizzazioni speciali dalle immagini

I flag dei diritti di accesso `setuid` e `setgid` consentono l'esecuzione di un file eseguibile con le autorizzazioni del proprietario o del gruppo del file eseguibile. Rimuovi tutti i file binari con tali diritti di accesso dall'immagine, in quanto possono essere utilizzati per eseguire l'escalation dei privilegi. Valuta la possibilità di rimuovere tutti gli shell (interpreti di comandi) e le utilità come `nc` e `curl` che possono essere utilizzati per scopi nocivi. Puoi trovare i file con diritti di accesso `setuid` e `setgid` utilizzando il comando seguente.

```
find / -perm /6000 -type f -exec ls -ld {} \;
```

Per rimuovere le autorizzazioni speciali da tali file, aggiungi la direttiva seguente all'immagine di container.

```
RUN find / -xdev -perm /6000 -type f -exec chmod a-s {} \; || true
```

Creazione di un set di immagini curate

Invece di consentire agli sviluppatori di creare le proprie immagini, crea un set di immagini controllate per i diversi stack di applicazioni della tua organizzazione. In tal modo, gli sviluppatori possono fare a meno di imparare a comporre Dockerfile e concentrarsi piuttosto sulla scrittura di codice. Man mano che le modifiche vengono incorporate nella base di codice, una pipeline CI/CD può compilare in automatico l'asset e quindi archiviarlo in un repository di artefatti. Infine, copia l'artefatto nell'immagine appropriata prima di inviarlo a un registro Docker come Amazon ECR. Come minimo, dovresti creare un set di immagini di base da cui gli sviluppatori possano creare i propri Dockerfile. È sconsigliabile estrarre immagini da Docker Hub. Non sempre sai che cosa c'è nell'immagine e circa un quinto delle 1.000 immagini principali presenta vulnerabilità. Un elenco di tali immagini e delle relative vulnerabilità è disponibile all'indirizzo https://vulnerablecontainers.org/.

Scansione dei pacchetti e delle librerie di applicazioni per individuare eventuali vulnerabilità

L'uso di librerie open source è ormai comune. Come per i sistemi operativi e i pacchetti OS, le librerie possono presentare vulnerabilità. Nell'ambito del ciclo di vita dello sviluppo, è necessario analizzare e aggiornare queste librerie quando vengono rilevate vulnerabilità critiche.

Docker Desktop esegue scansioni locali utilizzando Snyk. Può anche essere utilizzato per trovare vulnerabilità e potenziali problemi di licenza nelle librerie open source. Grazie alla possibilità di integrarlo direttamente nei flussi di lavoro degli sviluppatori, permette di mitigare i rischi posti dalle librerie open source. Per ulteriori informazioni, consulta i seguenti argomenti:

- Strumenti di sicurezza delle applicazioni open source include un elenco di strumenti per rilevare le vulnerabilità nelle applicazioni.

Esecuzione dell'analisi statica del codice

È consigliabile eseguire l'analisi statica del codice prima di creare un'immagine di container. Viene eseguita sul codice di origine e consente di identificare errori di codifica e codice che potrebbero essere sfruttati da un malintenzionato, come le iniezioni di errori. SonarQubeè un'opzione popolare per i test statici di sicurezza delle applicazioni (SAST), con supporto per una varietà di linguaggi di programmazione diversi.

Esecuzione di container come utente non root

È consigliabile eseguire i container come utente non root. Per impostazione predefinita, i container vengono eseguiti come utente `root` a meno che nel Dockerfile sia inclusa la direttiva `USER`. Le funzionalità Linux predefinite assegnate da Docker limitano le operazioni che possono essere eseguite come `root`, ma solo in misura marginale. Ad esempio, un container in esecuzione come `root` non può comunque accedere ai dispositivi.

Nell'ambito della pipeline CI/CD, è consigliabile eseguire il comando lint sui Dockerfile per cercare la direttiva `USER` e interrompere la build qualora mancasse. Per ulteriori informazioni, consulta i seguenti argomenti:

- DockerFile-LINT è uno strumento open source RedHat che può essere utilizzato per verificare se il file è conforme alle migliori pratiche.
- Hadolint è un altro strumento per creare immagini Docker conformi alle best practice.

Uso di un file system root di sola lettura

È consigliabile utilizzare un file system root di sola lettura. Il file system root di un container è scrivibile per impostazione predefinita. Quando configuri un container con un file system root `RO` (di sola lettura), ti viene imposto di definire in modo esplicito dove possono essere conservati i dati. Ciò riduce la superficie di attacco, perché non è possibile scrivere sul file system del container a meno che non vengano concesse autorizzazioni specifiche.

> **Note**
>
> Disporre di un file system root di sola lettura può causare problemi con alcuni pacchetti OS che si aspettano la possibilità di scrivere sul file system. Se hai intenzione di utilizzare file system root di sola lettura, esegui prima un test approfondito.

Configurazione delle 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 l'attività non prevede alcun limite di CPU, l'agente di container Amazon ECS le assegna un minimo di 2 CPU.

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.

Evitare l'esecuzione di container con privilegi (Amazon EC2)

È consigliabile evitare di eseguire i container come privilegiati. In background, i container vengono eseguiti come `privileged` vengono eseguiti con privilegi estesi sull'host. Ciò significa che il container eredita tutte le funzionalità di Linux assegnate a `root` sull'host. Il suo uso dovrebbe essere severamente limitato o vietato. Consigliamo di impostare la variabile di ambiente dell'agente di container Amazon ECS `ECS_DISABLE_PRIVILEGED` su `true` per evitare che i container vengano eseguiti come `privileged` su determinati host se l'opzione `privileged` non è necessaria.
In alternativa è possibile utilizzare AWS Lambda per scansionare le definizioni delle attività per verificare l'utilizzo del `privileged` parametro.

> **Note**
>
> L'esecuzione di un container come `privileged` non è supportata in Amazon ECS su AWS Fargate.

Rimozione delle funzionalità di Linux non necessarie dal container

Di seguito è riportato un elenco delle funzionalità di Linux predefinite assegnate ai container Docker. Per ulteriori informazioni su ciascuna funzionalità, consulta Panoramica delle funzionalità di Linux.

```
CAP_CHOWN, CAP_DAC_OVERRIDE, CAP_FOWNER, CAP_FSETID, CAP_KILL,
CAP_SETGID, CAP_SETUID, CAP_SETPCAP, CAP_NET_BIND_SERVICE,
CAP_NET_RAW, CAP_SYS_CHROOT, CAP_MKNOD, CAP_AUDIT_WRITE,
CAP_SETFCAP
```

Se un container non richiede tutte le funzionalità del kernel Docker elencate sopra, valuta la possibilità di eliminarle dal container. Per ulteriori informazioni su ciascuna funzionalità del kernel Docker, consulta. KernelCapabilities Puoi scoprire quali funzionalità sono in uso completando la procedura seguente:

- Installa il pacchetto OS libcap-ng ed esegui l'utilità `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.

Puoi utilizzare uno qualsiasi dei seguenti tutorial per distribuire attività su Amazon ECS utilizzando AWS CLI

Panoramica del tutorial	Ulteriori informazioni
Crea un'attività Linux per il tipo di lancio Fargate.	Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI
Crea un'attività Windows per il tipo di avvio Fargate.	Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI
Crea un'attività Linux per il tipo di avvio EC2.	Creazione di un'attività Amazon ECS per il tipo di lancio EC2 con AWS CLI

Puoi utilizzare uno dei seguenti tutorial per saperne di più sul monitoraggio e la registrazione.

Panoramica del tutorial	Ulteriori informazioni
Imposta una semplice funzione Lambda che ascolti gli eventi delle attività e li scriva in un flusso di log di CloudWatch Logs.	Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events
Configura una regola di EventBridge evento Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita	Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS

Panoramica del tutorial	Ulteriori informazioni	
perché uno dei suoi contenitori essenziali è terminato.		
Concatena i messaggi di log che originariamente appartenevano a un contesto ma erano suddivisi su più record o righe di registro.	Concatenazione di messaggi di log Amazon ECS multilinea o stack-trace	
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	Distribuzione di Fluent Bit su contenitori Amazon ECS Windows	

Puoi utilizzare uno dei seguenti tutorial per saperne di più su come utilizzare l'autenticazione Active Directory con un account di servizio gestito di gruppo su Amazon ECS.

Panoramica del tutorial	Ulteriori informazioni	
Usa l'account di servizio gestito di gruppo con contenitori Linux su EC2.	Utilizzo gMSA per Linux contenitori EC2 su Amazon ECS	
Usa un account di servizio gestito di gruppo con contenitori Windows su EC2.	Scopri come usare GMSAS per contenitori EC2 Windows per Amazon ECS	
Usa l'account di servizio gestito di gruppo con contenitori Linux su Fargate.	Utilizzo gMSA per i Linux container su Fargate	
Crea un'attività che esegua un contenitore Windows con	Utilizzo di contenitori Amazon ECS Windows con modalità	

Panoramica del tutorial	Ulteriori informazioni
credenziali per accedere ad Active Directory con un account di servizio gestito di gruppo senza dominio.	domainless gMSA utilizzando AWS CLI

Creazione di un'attività Amazon ECS Linux per il tipo di lancio Fargate con AWS CLI

La procedura seguente illustra come configurare un cluster, registrare una definizione dei processi, eseguire un processo Linux e come eseguire altri scenari comuni in Amazon ECS con la AWS CLI. Usa la versione più recente di. AWS CLI Per ulteriori informazioni sull'aggiornamento all'ultima versione, consulta Installazione dell' AWS Command Line Interface.

Argomenti

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- 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
- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.

- Il tuo AWS utente dispone delle autorizzazioni richieste specificate nell'esempio di AmazonECS_FullAccess policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine di container ospitata su Amazon ECR Public, quindi il processo deve avere accesso a Internet. Per assegnare alla tua attività un percorso verso Internet, scegli una delle seguenti opzioni.
  - Utilizza una sottorete privata con un gateway NAT con un indirizzo IP elastico.
  - Utilizza una sottorete pubblica e assegna un indirizzo IP pubblico all'attività.

  Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.

  Per informazioni sui gruppi e sulle regole di sicurezza, consulta la sezione Gruppi di sicurezza predefiniti per i VPC e Regole di esempio nella Guida per l'utente di Amazon Virtual Private Cloud.
- Se segui questo tutorial utilizzando una sottorete privata, puoi usare Amazon ECS Exec per interagire direttamente con il container e testare l'implementazione. Dovrai creare un ruolo IAM dell'attività per usare ECS Exec. Per ulteriori informazioni sul ruolo IAM dell'attività e su altri prerequisiti, consulta Utilizzo di Amazon ECS Exec per il debug.
- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza la necessità di creare la propria istanza EC2. Per ulteriori informazioni, consulta Cos'è? AWS CloudShell nella Guida AWS CloudShell per l'utente.

Fase 1: creare un cluster

Di default, l'account riceve il cluster `default`.

> **Note**
>
> Utilizzare il cluster `default` offre il vantaggio di non dover specificare l'opzione `--cluster` *`cluster_name`* nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi specificare ì`--cluster` *`cluster_name`* per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name fargate-cluster
```

Output:

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

Fase 2: Registra una definizione dei processi Linux

Prima di eseguire un'attività nel cluster ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente è una semplice definizione di attività che crea un'app Web PHP utilizzando l'immagine di container httpd ospitata su Docker Hub. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta Definizioni dei processi di Amazon ECS. Per questo tutorial, è necessario utilizzare `taskRoleArn` solo se stai distribuendo l'attività in una sottorete privata e desideri testare l'implementazione. Sostituisci `taskRoleArn` con il ruolo dell'attività IAM creato per utilizzare ECS Exec, come indicato in Prerequisiti.

```
 {
        "family": "sample-fargate",
        "networkMode": "awsvpc",
        "taskRoleArn": "arn:aws:iam::aws_account_id:role/execCommandRole",
        "containerDefinitions": [
            {
```

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

Salva il JSON della definizione dell'attività come file e trasmettilo con l'opzione `--cli-input-json file://`*`path_to_file.json`*.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/fargate-task.json
```

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

Fase 3: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando list-task-definitions. Il risultato restituito dal comando mostra i valori `family` e `revision` che puoi utilizzare insieme nelle chiamate run-task o start-task.

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

Output:

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate:1"
    ]
}
```

Fase 4: Crea un servizio

Dopo aver registrato un'attività per il tuo account, puoi creare un servizio per le attività registrate nel tuo cluster. Per questo esempio, crei un servizio con un'istanza della definizione delle attività `sample-fargate:1` in esecuzione nel cluster. L'attività richiede un percorso verso Internet, quindi ci sono due modi per ottenere questo risultato. Un modo consiste nell'utilizzare una sottorete privata configurata con un gateway NAT con un indirizzo IP elastico in una sottorete pubblica. Un altro modo consiste nell'utilizzare una sottorete pubblica e assegnare un indirizzo IP pubblico all'attività. Forniamo di seguito entrambi gli esempi.

Esempio di utilizzo di una sottorete privata. L'opzione `enable-execute-command` è necessaria per utilizzare Amazon ECS Exec.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service --
task-definition sample-fargate:1 --desired-count 1 --launch-type "FARGATE" --network-
configuration "awsvpcConfiguration={subnets=[subnet-abcd1234],securityGroups=[sg-
abcd1234]}" --enable-execute-command
```

Esempio di utilizzo di una sottorete pubblica.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service --
task-definition sample-fargate:1 --desired-count 1 --launch-type "FARGATE" --network-
```

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

Il comando create-service restituisce una descrizione della definizione di attività una volta completata la registrazione.

Fase 5: Elenca i servizi

Ottieni un elenco dei servizi per il tuo cluster. Verrà visualizzato il servizio creato nella sezione precedente. Potrai utilizzare più avanti il nome del servizio o l'ARN completo restituito da questo comando per la descrizione del servizio.

```
aws ecs list-services --cluster fargate-cluster
```

Output:

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

Fase 6: Descrivi il servizio in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi il servizio utilizzando il nome del servizio recuperato in precedenza.

```
aws ecs describe-services --cluster fargate-cluster --services fargate-service
```

In caso di esito positivo, verrà restituita una descrizione degli errori del servizio e dei servizi. Ad esempio, nella sezione `services`, sono disponibili informazioni sulle implementazioni, ad esempio lo stato delle attività in esecuzione o in sospeso. È inoltre possibile trovare informazioni sulla definizione delle attività, la configurazione della rete e gli eventi con indicazione del timestamp. Nella sezione Errori, sono disponibili informazioni sugli eventuali errori associati alla chiamata. Per la risoluzione dei problemi, vedere Messaggi degli eventi di servizio. Per ulteriori informazioni sulla descrizione del servizio, vedere Descrivere i servizi.

```
{
    "services": [
        {
```

```
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-abcd1234"
                    ],
                    "securityGroups": [
                        "sg-abcd1234"
                    ],
                    "assignPublicIp": "ENABLED"
                }
            },
            "launchType": "FARGATE",
            "enableECSManagedTags": false,
            "loadBalancers": [],
            "deploymentController": {
                "type": "ECS"
            },
            "desiredCount": 1,
            "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
            "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
            "deploymentConfiguration": {
                "maximumPercent": 200,
                "minimumHealthyPercent": 100
            },
            "createdAt": 1692283199.771,
            "schedulingStrategy": "REPLICA",
            "placementConstraints": [],
            "deployments": [
                {
                    "status": "PRIMARY",
                    "networkConfiguration": {
                        "awsvpcConfiguration": {
                            "subnets": [
                                "subnet-abcd1234"
                            ],
                            "securityGroups": [
                                "sg-abcd1234"
                            ],
                            "assignPublicIp": "ENABLED"
                        }
                    },
                    "pendingCount": 0,
                    "launchType": "FARGATE",
                    "createdAt": 1692283199.771,
```

```
                    "desiredCount": 1,
                    "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-
definition/sample-fargate:1",
                    "updatedAt": 1692283199.771,
                    "platformVersion": "1.4.0",
                    "id": "ecs-svc/9223370526043414679",
                    "runningCount": 0
                }
            ],
            "serviceName": "fargate-service",
            "events": [
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 53c0de40-ea3b-489f-a352-623bf1235f08) (task d0aec985-901b-488f-9fb4-61b991b332a3).",
                    "id": "92b8443e-67fb-4886-880c-07e73383ea83",
                    "createdAt": 1510811841.408
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 b4911bee-7203-4113-99d4-e89ba457c626) (task cc5853e3-6e2d-4678-8312-74f8a7d76474).",
                    "id": "d85c6ec6-a693-43b3-904a-a997e1fc844d",
                    "createdAt": 1510811601.938
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 cba86182-52bf-42d7-9df8-b744699e6cfc) (task f4c1ad74-a5c6-4620-90cf-2aff118df5fc).",
                    "id": "095703e1-0ca3-4379-a7c8-c0f1b8b95ace",
                    "createdAt": 1510811364.691
                }
            ],
            "runningCount": 0,
            "status": "ACTIVE",
            "serviceRegistries": [],
            "pendingCount": 0,
            "createdBy": "arn:aws:iam::aws_account_id:user/user_name",
            "platformVersion": "LATEST",
            "placementStrategy": [],
            "propagateTags": "NONE",
            "roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
            "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
sample-fargate:1"
        }
    ],
```

```
    "failures": []
}
```

Fase 7: Test

Test di un'attività implementata utilizzando una sottorete pubblica

Descrivi il processo nel servizio in modo da poter ottenere l'interfaccia di rete elastica (ENI) per il processo.

Per prima cosa, ottieni l'attività ARN.

```
aws ecs list-tasks --cluster fargate-cluster --service fargate-service
```

L'output contiene l'attività ARN.

```
{
    "taskArns": [
        "arn:aws:ecs:us-east-1:123456789012:task/fargate-service/EXAMPLE
    ]
}
```

Descrivi il processo e individua l'ID ENI. Usa l'attività ARN per il parametro `tasks`.

```
aws ecs describe-tasks --cluster fargate-cluster --tasks arn:aws:ecs:us-
east-1:123456789012:task/service/EXAMPLE
```

Le informazioni relative all'allegato sono elencate nell'output.

```
{
    "tasks": [
        {
            "attachments": [
                {
                    "id": "d9e7735a-16aa-4128-bc7a-b2d5115029e9",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnetabcd1234"
                        },
```

```
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-0fa40520aeEXAMPLE"
                        },
                    ]
                }
…
}
```

Descrivi l'ENI per ottenere l'indirizzo IP pubblico.

```
aws ec2 describe-network-interfaces --network-interface-id  eni-0fa40520aeEXAMPLE
```

L'indirizzo IP pubblico è indicato nell'output.

```
{
    "NetworkInterfaces": [
        {
            "Association": {
                "IpOwnerId": "amazon",
                "PublicDnsName": "ec2-34-229-42-222.compute-1.amazonaws.com",
                "PublicIp": "198.51.100.2"
            },
…
}
```

Inserisci l'indirizzo IP pubblico nel browser Web; dovresti visualizzare una pagina Web che mostra l'applicazione Amazon ECS di esempio.

Test di un'attività implementata utilizzando una sottorete privata

Descrivi l'attività e individua `managedAgents` per verificare che `ExecuteCommandAgent` sia in esecuzione. Annota `privateIPv4Address` per utilizzarlo in un secondo momento.

```
aws ecs describe-tasks --cluster fargate-cluster --tasks arn:aws:ecs:us-
east-1:123456789012:task/fargate-service/EXAMPLE
```

Le informazioni relative all'agente gestito sono elencate nell'output.

```
{
     "tasks": [
        {
```

```
            "attachments": [
                {
                    "id": "d9e7735a-16aa-4128-bc7a-b2d5115029e9",
                    "type": "ElasticNetworkInterface",
                    "status": "ATTACHED",
                    "details": [
                        {
                            "name": "subnetId",
                            "value": "subnetabcd1234"
                        },
                        {
                            "name": "networkInterfaceId",
                            "value": "eni-0fa40520aeEXAMPLE"
                        },
                        {
                            "name": "privateIPv4Address",
                            "value": "10.0.143.156"
                        }
                    ]
                }
            ],
        ...
        "containers": [
            {
            ...
          "managedAgents": [
                        {
                            "lastStartedAt": "2023-08-01T16:10:13.002000+00:00",
                            "name": "ExecuteCommandAgent",
                            "lastStatus": "RUNNING"
                        }
                ],
            ...
    }
```

Dopo aver verificato che `ExecuteCommandAgent` sia in esecuzione, puoi eseguire il comando seguente per utilizzare una shell (interprete di comandi) interattiva nel container dell'attività.

```
aws ecs execute-command --cluster fargate-cluster \
    --task  arn:aws:ecs:us-east-1:123456789012:task/fargate-service/EXAMPLE  \
    --container  fargate-app \
    --interactive \
    --command "/bin/sh"
```

Dopo l'esecuzione della shell (interprete di comandi) interattiva, esegui i comandi seguenti per installare cURL.

```
apt update
```

```
apt install curl
```

Dopo aver installato cURL, esegui il comando seguente utilizzando l'indirizzo IP privato ottenuto in precedenza.

```
 curl 10.0.143.156
```

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

```
<html>
    <head>
     <title>Amazon ECS Sample App</title>
     <style>body {margin-top: 40px; background-color: #333;} </style>
    </head>
      <body>
      <div style=color:white;text-align:center>
      <h1>Amazon ECS Sample App</h1>
      <h2>Congratulations!</h2> <p>Your application is now running on a container in
 Amazon ECS.</p>
      </div>
      </body>
</html>
```

Fase 8: elimina

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Elimina il servizio.

```
aws ecs delete-service --cluster fargate-cluster --service fargate-service --force
```

Elimina il cluster.

```
aws ecs delete-cluster --cluster fargate-cluster
```

Creazione di un'attività Amazon ECS Windows per il tipo di avvio Fargate con AWS CLI

La procedura seguente illustra come configurare un cluster, registrare una definizione dei processi, eseguire un processo Windows e come eseguire altri scenari comuni in Amazon ECS con la AWS CLI. Assicurati di usare la versione più recente di AWS CLI. Per ulteriori informazioni sull'aggiornamento all'ultima versione, consulta Installazione dell' AWS Command Line Interface.

Argomenti

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- L'ultima versione 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
- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.
- Il tuo AWS utente dispone delle autorizzazioni richieste specificate nell'esempio di AmazonECS_FullAccess policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Questo tutorial utilizza un'immagine di container ospitata su Docker Hub, quindi l'attività deve avere accesso a Internet. Per assegnare alla tua attività un percorso verso Internet, scegli una delle seguenti opzioni.
  - Utilizza una sottorete privata con un gateway NAT con un indirizzo IP elastico.

- Utilizza una sottorete pubblica e assegna un indirizzo IP pubblico all'attività.

  Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.

  Per informazioni sui gruppi e sulle regole di sicurezza, consulta la sezione Gruppi di sicurezza predefiniti per i VPC 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 la necessità di creare la propria istanza EC2. Per ulteriori informazioni, consulta Cos'è? AWS CloudShell nella Guida AWS CloudShell per l'utente.

Fase 1: creare un cluster

Di default, l'account riceve il cluster `default`.

> **Note**
>
> Utilizzare il cluster `default` offre il vantaggio di non dover specificare l'opzione `--cluster` *`cluster_name`* nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi specificare ì`--cluster` *`cluster_name`* per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name fargate-cluster
```

Output:

```
{
    "cluster": {
        "status": "ACTIVE",
        "statistics": [],
        "clusterName": "fargate-cluster",
        "registeredContainerInstancesCount": 0,
        "pendingTasksCount": 0,
        "runningTasksCount": 0,
        "activeServicesCount": 0,
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster"
    }
```

```
}
```

Fase 2: Registrazione di una definizione di attività di Windows

Prima di eseguire un processo Windows nel cluster Amazon ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente contiene una semplice definizione di attività che crea un'app Web. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta Definizioni dei processi di Amazon ECS.

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

```
    "requiresCompatibilities": ["FARGATE"]
}
```

L'esempio precedente JSON può essere passato a AWS CLI in due modi: è possibile salvare la definizione dell'attività JSON come file e passarla con l'`--cli-input-json file://path_to_file.json`opzione.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/fargate-task.json
```

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

Fase 3: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando list-task-definitions. Il risultato restituito dal comando mostra i valori `family` e `revision` che puoi utilizzare insieme nelle chiamate run-task o start-task.

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

Output:

```
{
    "taskDefinitionArns": [
        "arn:aws:ecs:region:aws_account_id:task-definition/sample-fargate-windows:1"
    ]
}
```

Fase 4: Crea un servizio

Dopo aver registrato un'attività per il tuo account, puoi creare un servizio per le attività registrate nel tuo cluster. Per questo esempio, crei un servizio con un'istanza della definizione delle attività `sample-fargate:1` in esecuzione nel cluster. L'attività richiede un percorso verso Internet, quindi ci sono due modi per ottenere questo risultato. Un modo consiste nell'utilizzare una sottorete privata configurata con un gateway NAT con un indirizzo IP elastico in una sottorete pubblica. Un altro modo consiste nell'utilizzare una sottorete pubblica e assegnare un indirizzo IP pubblico all'attività. Forniamo di seguito entrambi gli esempi.

Esempio di utilizzo di una sottorete privata.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service
 --task-definition sample-fargate-windows:1 --desired-count 1 --launch-type
 "FARGATE" --network-configuration "awsvpcConfiguration={subnets=[subnet-
abcd1234],securityGroups=[sg-abcd1234]}"
```

Esempio di utilizzo di una sottorete pubblica.

```
aws ecs create-service --cluster fargate-cluster --service-name fargate-service
 --task-definition sample-fargate-windows:1 --desired-count 1 --launch-type
 "FARGATE" --network-configuration "awsvpcConfiguration={subnets=[subnet-
abcd1234],securityGroups=[sg-abcd1234],assignPublicIp=ENABLED}"
```

Il comando create-service restituisce una descrizione della definizione di attività una volta completata la registrazione.

Fase 5: Elenca i servizi

Ottieni un elenco dei servizi per il tuo cluster. Verrà visualizzato il servizio creato nella sezione precedente. Potrai utilizzare più avanti il nome del servizio o l'ARN completo restituito da questo comando per la descrizione del servizio.

```
aws ecs list-services --cluster fargate-cluster
```

Output:

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

Fase 6: Descrivi il servizio in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi il servizio utilizzando il nome del servizio recuperato in precedenza.

```
aws ecs describe-services --cluster fargate-cluster --services fargate-service
```

In caso di esito positivo, verrà restituita una descrizione degli errori del servizio e dei servizi. Ad esempio, nella sezione Servizi, sono disponibili informazioni sulle distribuzioni, ad esempio lo stato delle attività in esecuzione o in sospeso. È inoltre possibile trovare informazioni sulla definizione delle attività, la configurazione della rete e gli eventi con indicazione del timestamp. Nella sezione Errori, sono disponibili informazioni sugli eventuali errori associati alla chiamata. Per la risoluzione dei problemi, vedere Messaggi degli eventi di servizio. Per ulteriori informazioni sulla descrizione del servizio, vedere Descrivere i servizi.

```
{
    "services": [
        {
            "status": "ACTIVE",
            "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-definition/
sample-fargate-windows:1",
            "pendingCount": 2,
            "launchType": "FARGATE",
            "loadBalancers": [],
            "roleArn": "arn:aws:iam::aws_account_id:role/aws-service-role/
ecs.amazonaws.com/AWSServiceRoleForECS",
            "placementConstraints": [],
            "createdAt": 1510811361.128,
            "desiredCount": 2,
            "networkConfiguration": {
                "awsvpcConfiguration": {
                    "subnets": [
                        "subnet-abcd1234"
                    ],
                    "securityGroups": [
                        "sg-abcd1234"
                    ],
                    "assignPublicIp": "DISABLED"
                }
            },
            "platformVersion": "LATEST",
            "serviceName": "fargate-service",
            "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/fargate-cluster",
            "serviceArn": "arn:aws:ecs:region:aws_account_id:service/fargate-service",
            "deploymentConfiguration": {
                "maximumPercent": 200,
                "minimumHealthyPercent": 100
            },
            "deployments": [
```

```
                {
                    "status": "PRIMARY",
                    "networkConfiguration": {
                        "awsvpcConfiguration": {
                            "subnets": [
                                "subnet-abcd1234"
                            ],
                            "securityGroups": [
                                "sg-abcd1234"
                            ],
                            "assignPublicIp": "DISABLED"
                        }
                    },
                    "pendingCount": 2,
                    "launchType": "FARGATE",
                    "createdAt": 1510811361.128,
                    "desiredCount": 2,
                    "taskDefinition": "arn:aws:ecs:region:aws_account_id:task-
definition/sample-fargate-windows:1",
                    "updatedAt": 1510811361.128,
                    "platformVersion": "0.0.1",
                    "id": "ecs-svc/9223370526043414679",
                    "runningCount": 0
                }
            ],
            "events": [
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 53c0de40-ea3b-489f-a352-623bf1235f08) (task d0aec985-901b-488f-9fb4-61b991b332a3).",
                    "id": "92b8443e-67fb-4886-880c-07e73383ea83",
                    "createdAt": 1510811841.408
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 b4911bee-7203-4113-99d4-e89ba457c626) (task cc5853e3-6e2d-4678-8312-74f8a7d76474).",
                    "id": "d85c6ec6-a693-43b3-904a-a997e1fc844d",
                    "createdAt": 1510811601.938
                },
                {
                    "message": "(service fargate-service) has started 2 tasks: (task
 cba86182-52bf-42d7-9df8-b744699e6cfc) (task f4c1ad74-a5c6-4620-90cf-2aff118df5fc).",
                    "id": "095703e1-0ca3-4379-a7c8-c0f1b8b95ace",
                    "createdAt": 1510811364.691
                }
```

```
            ],
            "runningCount": 0,
            "placementStrategy": []
        }
    ],
    "failures": []
}
```

Fase 7: eliminare

Una volta terminato questo tutorial, è necessario eliminare le risorse associate per evitare costi aggiuntivi per le risorse non utilizzate.

Elimina il servizio.

```
aws ecs delete-service --cluster fargate-cluster --service fargate-service --force
```

Elimina il cluster.

```
aws ecs delete-cluster --cluster fargate-cluster
```

Creazione di un'attività Amazon ECS per il tipo di lancio 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 di. AWS CLI Per ulteriori informazioni sull'aggiornamento all'ultima versione, consulta Installazione dell' AWS Command Line Interface.

Argomenti

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- 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
- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.
- Il tuo AWS utente dispone delle autorizzazioni richieste specificate nell'esempio di AmazonECS_FullAccess policy IAM.
- Sono disponibili un VPC e un gruppo di sicurezza creati per l'uso. Per ulteriori informazioni, consulta the section called “Crea un cloud privato virtuale”.
- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza la necessità di creare la propria istanza EC2. Per ulteriori informazioni, consulta Cos'è? AWS CloudShell nella Guida AWS CloudShell per l'utente.

Fase 1: creare un cluster

All'avvio della prima istanza di container, di default l'account riceve il cluster `default`.

> **Note**
>
> Utilizzare il cluster `default` offre il vantaggio di non dover specificare l'opzione `--cluster` *`cluster_name`* nei comandi successivi. Se crei un cluster diverso da quello predefinito, devi specificare ì`--cluster` *`cluster_name`* per ogni comando che prevedi di usare con tale cluster.

Crea il tuo cluster con un nome univoco con il comando seguente:

```
aws ecs create-cluster --cluster-name MyCluster
```

Output:

```
{
    "cluster": {
        "clusterName": "MyCluster",
        "status": "ACTIVE",
        "clusterArn": "arn:aws:ecs:region:aws_account_id:cluster/MyCluster"
    }
}
```

Fase 2: Avvia un'istanza con l'AMI Amazon ECS

Devi disporre di un'istanza di container Amazon ECS nel cluster prima di potervi eseguire operazioni. Se non disponi di istanze di container nel cluster, consulta Avvio di un'istanza di container Linux di Amazon ECS per ulteriori informazioni.

Fase 3: Elenca le istanze di container

Entro pochi minuti dall'avvio dell'istanza di container, l'agente Amazon ECS registra quest'ultima nel cluster predefinito. Per visualizzare l'elenco delle istanze di container in un cluster, esegui questo comando:

```
aws ecs list-container-instances --cluster default
```

Output:

```
{
    "containerInstanceArns": [
        "arn:aws:ecs:us-east-1:aws_account_id:container-instance/container_instance_ID"
    ]
}
```

Fase 4: Descrivi la tua istanza di container

Dopo aver ottenuto l'ARN o l'ID di un'istanza di container, puoi utilizzare il comando describe-container-instances per ottenere informazioni importanti sull'istanza, ad esempio le risorse di CPU e memoria registrate e quelle ancora disponibili.

```
aws ecs describe-container-instances --cluster default --container-
instances container_instance_ID
```

Output:


```
{
    "failures": [],
    "containerInstances": [
        {
            "status": "ACTIVE",
            "registeredResources": [
                {
                    "integerValue": 1024,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
                {
                    "integerValue": 995,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                },
                {
                    "name": "PORTS_UDP",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
```

```
            "ec2InstanceId": "instance_id",
            "agentConnected": true,
            "containerInstanceArn": "arn:aws:ecs:us-west-2:aws_account_id:container-
instance/container_instance_ID",
            "pendingTasksCount": 0,
            "remainingResources": [
                {
                    "integerValue": 1024,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "CPU",
                    "doubleValue": 0.0
                },
                {
                    "integerValue": 995,
                    "longValue": 0,
                    "type": "INTEGER",
                    "name": "MEMORY",
                    "doubleValue": 0.0
                },
                {
                    "name": "PORTS",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [
                        "22",
                        "2376",
                        "2375",
                        "51678"
                    ],
                    "type": "STRINGSET",
                    "integerValue": 0
                },
                {
                    "name": "PORTS_UDP",
                    "longValue": 0,
                    "doubleValue": 0.0,
                    "stringSetValue": [],
                    "type": "STRINGSET",
                    "integerValue": 0
                }
            ],
            "runningTasksCount": 0,
            "attributes": [
```

```
                {
                    "name": "com.amazonaws.ecs.capability.privileged-container"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.17"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.18"
                },
                {
                    "name": "com.amazonaws.ecs.capability.docker-remote-api.1.19"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.json-file"
                },
                {
                    "name": "com.amazonaws.ecs.capability.logging-driver.syslog"
                }
            ],
            "versionInfo": {
                "agentVersion": "1.5.0",
                "agentHash": "b197edd",
                "dockerVersion": "DockerVersion: 1.7.1"
            }
        }
    ]
}
```

È inoltre possibile trovare l'ID istanza Amazon EC2 che è possibile utilizzare per monitorare l'istanza nella console Amazon EC2 o con il comando aws ec2 describe-instances --instance-id ***instance_id***.

Fase 5: Registra una definizione di attività

Prima di eseguire un'attività nel cluster ECS, devi registrare una definizione di attività. Le definizioni di attività sono elenchi di container raggruppati. L'esempio seguente illustra una semplice definizione di attività che utilizza un'immagine `busybox` da Docker Hub ed entra semplicemente in sospensione per 360 secondi. Per ulteriori informazioni sui parametri disponibili per la definizione di attività, consulta Definizioni dei processi di Amazon ECS.

```
{
    "containerDefinitions": [
```

```
        {
            "name": "sleep",
            "image": "busybox",
            "cpu": 10,
            "command": [
                "sleep",
                "360"
            ],
            "memory": 10,
            "essential": true
        }
    ],
    "family": "sleep360"
}
```

L'esempio precedente JSON può essere passato a AWS CLI in due modi: è possibile salvare la definizione dell'attività JSON come file e passarla con l'`--cli-input-json file://`*`path_to_file.json`*opzione. Oppure puoi inserire il carattere di escape prima delle virgolette nel JSON e trasmettere le definizioni del container JSON nella riga di comando, come nell'esempio riportato di seguito. Se scegli di trasmettere le definizioni del container nella riga di comando, è necessario aggiungere al comando il parametro `--family` utilizzato per mantenere la reciproca associazione tra più versioni della definizione di attività.

Per utilizzare un file JSON per le definizioni dei container:

```
aws ecs register-task-definition --cli-input-json file://$HOME/tasks/sleep360.json
```

Per utilizzare una stringa JSON per le definizioni dei container:

```
aws ecs register-task-definition --family sleep360 --container-definitions "[{\"name
\":\"sleep\",\"image\":\"busybox\",\"cpu\":10,\"command\":[\"sleep\",\"360\"],\"memory
\":10,\"essential\":true}]"
```

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

```
{
    "taskDefinition": {
        "volumes": [],
        "taskDefinitionArn": "arn:aws:ec2:us-east-1:aws_account_id:task-definition/
sleep360:1",
```

```
        "containerDefinitions": [
            {
                "environment": [],
                "name": "sleep",
                "mountPoints": [],
                "image": "busybox",
                "cpu": 10,
                "portMappings": [],
                "command": [
                    "sleep",
                    "360"
                ],
                "memory": 10,
                "essential": true,
                "volumesFrom": []
            }
        ],
        "family": "sleep360",
        "revision": 1
    }
}
```

Fase 6: Elenca le definizioni di attività

Puoi sempre ottenere un elenco delle definizioni di attività per il tuo account tramite il comando list-task-definitions. Il risultato restituito dal comando mostra i valori `family` e `revision` che puoi utilizzare insieme nelle chiamate run-task o start-task.

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

Output:

```
{
    "taskDefinitionArns": [
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep300:1",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep300:2",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/sleep360:1",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:3",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:4",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:5",
        "arn:aws:ec2:us-east-1:aws_account_id:task-definition/wordpress:6"
    ]
```

```
}
```

Fase 7: Esegui un'attività

Dopo aver registrato un'attività per il tuo account e aver avviato un'istanza di container registrata nel cluster, puoi eseguire l'attività registrata in tale cluster. In questo esempio, una singola istanza della definizione di attività `sleep360:1` viene posizionata nel tuo cluster predefinito.

```
aws ecs run-task --cluster default --task-definition sleep360:1 --count 1
```

Output:

```
{
    "tasks": [
        {
            "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ]
            },
            "lastStatus": "PENDING",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:aws_account_id:container-
instance/container_instance_ID",
            "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/default",
            "desiredStatus": "RUNNING",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/
sleep360:1",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-
east-1:aws_account_id:container/container_ID",
                    "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
                    "lastStatus": "PENDING",
                    "name": "sleep"
                }
            ]
        }
    ]
}
```

Fase 8: Elenca le attività

Ottieni un elenco delle attività per il tuo cluster. Verrà visualizzata l'attività eseguita nella sezione precedente. Potrai utilizzare più avanti l'ID attività o l'ARN completo restituito da questo comando per la descrizione dell'attività.

```
aws ecs list-tasks --cluster default
```

Output:

```
{
    "taskArns": [
        "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID"
    ]
}
```

Fase 9: Descrivi l'attività in esecuzione

Per ottenere ulteriori informazioni sulle attività, descrivi l'attività utilizzando l'ID recuperato in precedenza.

```
aws ecs describe-tasks --cluster default --task task_ID
```

Output:

```
{
    "failures": [],
    "tasks": [
        {
            "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
            "overrides": {
                "containerOverrides": [
                    {
                        "name": "sleep"
                    }
                ]
            },
            "lastStatus": "RUNNING",
            "containerInstanceArn": "arn:aws:ecs:us-east-1:aws_account_id:container-
instance/container_instance_ID",
```

```
            "clusterArn": "arn:aws:ecs:us-east-1:aws_account_id:cluster/default",
            "desiredStatus": "RUNNING",
            "taskDefinitionArn": "arn:aws:ecs:us-east-1:aws_account_id:task-definition/
sleep360:1",
            "containers": [
                {
                    "containerArn": "arn:aws:ecs:us-
east-1:aws_account_id:container/container_ID",
                    "taskArn": "arn:aws:ecs:us-east-1:aws_account_id:task/task_ID",
                    "lastStatus": "RUNNING",
                    "name": "sleep",
                    "networkBindings": []
                }
            ]
        }
    ]
}
```

Configurazione di Amazon ECS per CloudWatch ascoltare gli eventi Events

Scopri come configurare una semplice funzione Lambda che ascolta gli eventi delle attività e li scrive in un flusso di log di CloudWatch Logs.

Prerequisito: configurazione di un cluster di verifica

Se non disponi di un cluster in esecuzione da cui acquisire eventi, segui la procedura descritta in the section called “Creazione di un cluster per il tipo di lancio Fargate” per crearne uno. Al termine del tutorial, potrai eseguire un processo su questo cluster per verificare la corretta configurazione della funzione Lambda.

Fase 1: Creazione della funzione Lambda

In questa procedura, viene creata una funzione Lambda semplice che funge da destinazione per i messaggi del flusso di eventi di Amazon ECS.

1. Apri la AWS Lambda console all'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. Selezionare Salva.

Fase 2: Registrazione di una regola di evento

Successivamente, crei una regola CloudWatch degli eventi che acquisisce gli eventi delle attività provenienti dai tuoi cluster Amazon ECS. Questa regola acquisisce tutti gli eventi provenienti da tutti i cluster all'interno dell'account in cui è definita. Gli stessi messaggi di attività contengono informazioni sull'origine dell'evento, tra cui il cluster in cui esso risiede, utilizzabile per filtrare e ordinare gli eventi in modo programmatico.

> **Note**
>
> Quando usi AWS Management Console per creare una regola di evento, la console aggiunge automaticamente le autorizzazioni IAM necessarie per concedere a CloudWatch Events l'autorizzazione a chiamare la tua funzione Lambda. Se stai creando una regola di evento

utilizzando il AWS CLI, devi concedere questa autorizzazione in modo esplicito. Per ulteriori informazioni, consulta Events and Event Patterns nella Amazon CloudWatch Events 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
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": [
        {
            "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,
            "memory": 300,
            "image": "httpd:2.4",
            "name": "simple-app"
        }
    ],
    "family": "console-sample-app-static"
}
```

5. Scegli 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 all'indirizzo https://console.aws.amazon.com/ecs/v2.
2. Scegli Task definitions (Definizioni di attività).

3. Scegli console-sample-app-static, quindi scegli Deploy (Implementa), seguito da Run new task (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 Log, quindi seleziona il gruppo di log della funzione Lambda (ad esempio, /aws/lambda/*my-function*).
7. Seleziona un flusso di log per visualizzare i dati di evento.

Invio di avvisi di Amazon Simple Notification Service per eventi di interruzione delle attività di Amazon ECS

Configura una regola di EventBridge evento Amazon che acquisisca solo gli eventi delle attività in cui l'attività ha smesso di essere eseguita perché uno dei suoi contenitori essenziali è terminato. L'evento invia solo eventi di processo con una specifica proprietà `stoppedReason` all'argomento Amazon SNS designato.

Prerequisito: configurazione di un cluster di verifica

Se non disponi di un cluster in esecuzione da cui acquisire eventi, segui la procedura descritta in Nozioni di base sull'utilizzo della console con i container Linux su AWS Fargate per crearne uno. Alla fine di questo tutorial, esegui un'attività su questo cluster per verificare di aver configurato correttamente l'argomento e la EventBridge regola di Amazon SNS.

Prerequisito: configurazione delle autorizzazioni per Amazon SNS

EventBridge Per consentire la pubblicazione su un argomento Amazon SNS, usa i comandi aws sns get-topic-attributes e aws sns. set-topic-attributes

Per ulteriori informazioni su come aggiungere l'autorizzazione, consulta la sezione Amazon SNS permissions (Autorizzazioni di Amazon SNS) nella Guida per gli sviluppatori di Servizio di notifica semplice Amazon.

Aggiungi le autorizzazioni seguenti:

```
{
  "Sid": "PublishEventsToMyTopic",
  "Effect": "Allow",
  "Principal": {
```

```
      "Service": "events.amazonaws.com"
    },
    "Action": "sns: Publish",
    "Resource": "arn:aws:sns:region:account-id:TaskStoppedAlert",
}
```

Fase 1: Creazione e sottoscrizione a un argomento Amazon SNS

In questo tutorial, configuri un argomento Amazon SNS che funga da destinazione evento per la nuova regola di evento.

Per informazioni sulla creazione e l'abbonamento a un argomento Amazon SNS, consulta Nozioni di base su Amazon SNS nella Guida per gli sviluppatori di Servizio di notifica semplice Amazon e utilizza la tabella seguente per determinare quali opzioni selezionare.

Opzione	Valore
Type	Standard
Nome	TaskStoppedAvviso
Protocollo	E-mail
Endpoint	Un indirizzo e-mail a cui hai attualmente accesso

Fase 2: Registrazione di una regola di evento

Successivamente, registra una regola dell'evento che acquisisca solo eventi di arresto dell'attività per attività con container interrotti.

Per informazioni su come creare e sottoscrivere un argomento di Amazon SNS, consulta Create a rule EventBridge in Amazon nella Amazon EventBridge User Guide e utilizza la tabella seguente per determinare quali opzioni selezionare.

Opzione	Valore
Tipo di regola	Regola con un modello di evento

Opzione	Valore
Origine eventi	AWS eventi o eventi per i EventBridge partner
Modello di evento	Modello personalizzato (editor JSON)

Modello di evento	{
	"source":[
	"aws.ecs"
	],
	"detail-type":[
	"ECS Task State Change"
	],
	"detail":{
	"lastStatus":[
	"STOPPED"
	],
	"stoppedReason":[
	"Essential container in task exited"
	]
	}
	}

Target type (Tipo di destinazione)	AWS servizio
Target	Argomento SNS
Argomento	TaskStoppedAlert (L'argomento che hai creato nel passaggio 1)

Fase 3: Test del tuo articolo

Verificare che la regola funzioni eseguendo un'attività che termine poco dopo l'avvio. Se la regola dell'evento è configurata correttamente, riceverai una e-mail contenente il testo dell'evento entro pochi minuti. Se si dispone di una definizione di attività esistente in grado di soddisfare i requisiti della regola, eseguire un'attività utilizzando tale definizione. In caso contrario, le fasi seguenti descrivono la procedura di registrazione di una definizione di attività Fargate e la relativa esecuzione.

1. Apri la console all'indirizzo 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, modifica il file JSON e copia quanto segue nell'editor.

```
{
   "containerDefinitions":[
      {
         "command":[
            "sh",
            "-c",
            "sleep 5"
         ],
         "essential":true,
         "image":"amazonlinux:2",
         "name":"test-sleep"
      }
   ],
   "cpu":"256",
   "executionRoleArn":"arn:aws:iam::012345678910:role/ecsTaskExecutionRole",
   "family":"fargate-task-definition",
   "memory":"512",
   "networkMode":"awsvpc",
   "requiresCompatibilities":[
      "FARGATE"
   ]
}
```

5. Scegli Crea.

Per eseguire un'attività dalla console

1. Apri la console all'indirizzo 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, seleziona fargate-task-definition.
6. Per Desired tasks (Attività desiderate), specifica il numero di attività da avviare.
7. Scegli Crea.

Concatenazione di messaggi di log Amazon ECS multilinea o stack-trace

A partire dalla versione 2.22.0 AWS di Fluent Bit, è incluso un filtro multilinea. Il filtro multilinea aiuta a concatenare i messaggi di log originariamente appartenenti a un contesto ma suddivisi su più record o righe di registro. Per ulteriori informazioni sul filtro multilinea, consulta la documentazione di Fluent Bit.

Esempi comuni di messaggi di log divisi sono:

- Tracce dello stack.
- Applicazioni che stampano i log su più righe.
- Registra i messaggi che sono stati divisi perché erano più lunghi della dimensione massima del buffer di runtime specificata. È possibile concatenare i messaggi di registro suddivisi in base al runtime del contenitore seguendo l'esempio su GitHub: Esempio: FireLens Concatenate log di container parziali/divisi.

Autorizzazioni IAM richieste

Disponi delle autorizzazioni IAM necessarie affinché l'agente container estragga le immagini del contenitore da Amazon ECR e che il container instrada i log verso Logs. CloudWatch

Per tali autorizzazioni devi disporre anche dei seguenti ruoli:

- Un ruolo IAM del processo.

- Un ruolo IAM per l'esecuzione di attività.

Come utilizzare l'editor di policy JSON per creare una policy

1. Accedi AWS Management Console e apri la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione a sinistra, seleziona Policies (Policy).

   Se è la prima volta che selezioni Policy, verrà visualizzata la pagina Benvenuto nelle policy gestite. Seleziona Inizia.
3. Nella parte superiore della pagina, scegli Crea policy.
4. Nella sezione Editor di policy, scegli l'opzione JSON.
5. Inserisci il documento di policy JSON seguente:

```
{
 "Version": "2012-10-17",
 "Statement": [{
  "Effect": "Allow",
  "Action": [
   "logs:CreateLogStream",
   "logs:CreateLogGroup",
   "logs:PutLogEvents"
  ],
  "Resource": "*"
 }]
}
```

6. Seleziona Successivo.

   > **Note**
   >
   > È possibile alternare le opzioni dell'editor Visivo e JSON in qualsiasi momento. Se tuttavia si apportano modifiche o si seleziona Successivo nell'editor Visivo, IAM potrebbe ristrutturare la policy in modo da ottimizzarla per l'editor visivo. Per ulteriori informazioni, consulta Modifica della struttura delle policy nella Guida per l'utente di IAM.

7. Nella pagina Rivedi e crea, inserisci un valore in Nome policy e Descrizione (facoltativo) per la policy in fase di creazione. Rivedi Autorizzazioni definite in questa policy per visualizzare le autorizzazioni concesse dalla policy.

8. Seleziona Crea policy per salvare la nuova policy.

Determinazione dei casi in cui utilizzare l'impostazione del log multilinea

Di seguito sono riportati alcuni esempi di frammenti di log che vedi nella console CloudWatch Logs con l'impostazione di registro predefinita. Puoi guardare la riga che inizia con `log` per determinare se è necessario il filtro multilinea. Quando il contesto è lo stesso, puoi utilizzare l'impostazione di registro multilinea. In questo esempio, il contesto è «com.myproject.model». MyProject».

```
2022-09-20T15:47:56:595-05-00                          {"container_id":
 "82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-
app", "source=": "stdout", "log": ": "     at com.myproject.modele.
(MyProject.badMethod.java:22)",
    {
      "container_id":  "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
      "container_name: ": "example-app",
      "source": "stdout",
      "log": ": "     at com.myproject.model.MyProject.badMethod(MyProject.java:22)",
      "ecs_cluster": "default",
      "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE",
      "ecs_task_definition": "firelense-example-multiline:3"
     }
```

```
2022-09-20T15:47:56:595-05-00                          {"container_id":
 "82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-app", "stdout",
 "log": ": "     at com.myproject.modele.(MyProject.oneMoreMethod.java:18)",
    {
      "container_id":  "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
      "container_name: ": "example-app",
      "source": "stdout",
      "log": ": "     at
 com.myproject.model.MyProject.oneMoreMethod(MyProject.java:18)",
      "ecs_cluster": "default",
      "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE,
      "ecs_task_definition": "firelense-example-multiline:3"
     }
```

Dopo aver utilizzato l'impostazione del log multilinea, l'output sarà simile all'esempio seguente.

```
2022-09-20T15:47:56:595-05-00                          {"container_id":
 "82ba37cada1d44d389b03e78caf74faa-EXAMPLE", "container_name": "example-app",
 "stdout",...
    {
      "container_id":  "82ba37cada1d44d389b03e78caf74faa-EXAMPLE",
      "container_name: ": "example-app",
      "source": "stdout",
      "log:    "September 20, 2022 06:41:48 Exception in thread \"main\"
 java.lang.RuntimeException: Something has gone wrong, aborting!\n
    at com.myproject.module.MyProject.badMethod(MyProject.java:22)\n    at
    at com.myproject.model.MyProject.oneMoreMethod(MyProject.java:18)
 com.myproject.module.MyProject.main(MyProject.java:6)",
      "ecs_cluster": "default",
      "ecs_task_arn": "arn:aws:region:123456789012:task/default/
b23c940d29ed4714971cba72cEXAMPLE",
      "ecs_task_definition": "firelense-example-multiline:2"
     }
```

Opzioni di analisi e concatenazione

Per analizzare i log e concatenare righe divise a causa di nuove righe, puoi utilizzare una di queste due opzioni.

- Utilizza il tuo file parser contenente le regole per analizzare e concatenare le righe che appartengono allo stesso messaggio.
- Utilizzare un parser integrato Fluent Bit. Per l'elenco dei linguaggi supportati dai parser integrati Fluent Bit, consulta la documentazione di Fluent Bit.

Il seguente tutorial illustra i passaggi per ogni caso d'uso. I passaggi mostrano come concatenare più righe e inviare i log ad Amazon. CloudWatch Puoi specificare una destinazione diversa per i log.

Esempio: utilizzare un parser creato

In questo esempio eseguirai i passaggi seguenti:

1. Crea e carica l'immagine per un container Fluent Bit.

2. Crea e carica l'immagine per un'applicazione demo multilinea che esegue, fallisce e genera una traccia dello stack multilinea.
3. Crea la definizione di un processo e avviare il processo.
4. Visualizza i log per verificare che i messaggi che si estendono su più righe appaiano concatenati.

Creare e caricare l'immagine per un container Fluent Bit

Questa immagine includerà il file parser in cui si specifica l'espressione regolare e un file di configurazione che fa riferimento al file parser.

1. Crea una cartella con il nome `FluentBitDockerImage`.
2. Nella cartella, crea il file parser contenente le regole per analizzare il log e concatenare le righe che appartengono allo stesso messaggio.

   a. Incolla i contenuti seguenti nel file parser:

```
[MULTILINE_PARSER]
    name          multiline-regex-test
    type          regex
    flush_timeout 1000
    #
    # Regex rules for multiline parsing
    # ---------------------------------
    #
    # configuration hints:
    #
    #  - first state always has the name: start_state
    #  - every field in the rule must be inside double quotes
    #
    # rules |   state name  | regex pattern                  | next state
    # ------|---------------|--------------------------------------------
    rule      "start_state"   "/(Dec \d+ \d+\:\d+\:\d+)(.*)/"  "cont"
    rule      "cont"          "/^\s+at.*/"                     "cont"
```

   Mentre personalizzi il modello di regex, ti consigliamo di utilizzare un editor di espressioni regolari per provare l'espressione.

   b. Salva il file con nome `parsers_multiline.conf`.
3. All'interno della cartella `FluentBitDockerImage`, crea un file di configurazione personalizzato che fa riferimento al file parser creato nel passaggio precedente.

Per ulteriori informazioni sul file di configurazione personalizzato, consulta Specifica di un file di configurazione personalizzato nella Guida per gli sviluppatori di Amazon Elastic Container Service

a. Incolla i contenuti seguenti nel file:

```
[SERVICE]
    flush                 1
    log_level             info
    parsers_file          /parsers_multiline.conf

[FILTER]
    name                  multiline
    match                 *
    multiline.key_content log
    multiline.parser      multiline-regex-test
```

> **Note**
>
> È necessario utilizzare il percorso assoluto del parser.

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`*

      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 a fluent-bit-multiline-repo . `repositoryUri`

   d. Effettua l'autenticazione su Amazon ECR eseguendo il comando `aws ecr get-login-password` e specificando l'ID del log in cui desideri eseguire l'autenticazione: `aws ecr`

`get-login-password | docker login --username AWS --password-stdin` *`registry ID`*`.dkr.ecr.`*`region`*`.amazonaws.com`

Esempio: `ecr get-login-password | docker login --username AWS --password-stdin xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com`

Viene visualizzato un messaggio di accesso riuscito.

e. Invia l'immagine ad Amazon ECR: `docker push` *`registry ID`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`repository name`*

Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-repo`

Crea e carica l'immagine per un'applicazione demo multilinea

Questa immagine includerà un file di script Python che esegue l'applicazione e un file di log di esempio.

Quando esegui il processo, l'applicazione simula le esecuzioni, quindi fallisce e crea una traccia dello stack.

1. Crea una cartella denominata `multiline-app`: `mkdir multiline-app`
2. Crea un file di script Python.
   a. Nella cartella `multiline-app`, crea un file con il nome `main.py`.
   b. Incolla i contenuti seguenti nel file:

```
import os
import time
file1 = open('/test.log', 'r')
Lines = file1.readlines()

count = 0

for i in range(10):
    print("app running normally...")
    time.sleep(1)

# Strips the newline character
for line in Lines:
```

```
    count += 1
    print(line.rstrip())
print(count)
print("app terminated.")
```

c. Salvare il file `main.py`.

3. Crea un file di log di esempio.

a. Nella cartella `multiline-app`, crea un file con il nome `test.log`.

b. Incolla i contenuti seguenti nel file:

```
single line...
Dec 14 06:41:08 Exception in thread "main" java.lang.RuntimeException:
 Something has gone wrong, aborting!
    at com.myproject.module.MyProject.badMethod(MyProject.java:22)
    at com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)
    at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)
    at com.myproject.module.MyProject.someMethod(MyProject.java:10)
    at com.myproject.module.MyProject.main(MyProject.java:6)
another line...
```

c. Salvare il file `test.log`.

4. All'interno della cartella `multiline-app`, crea il Dockerfile.

a. Incolla i contenuti seguenti nel file:

```
FROM public.ecr.aws/amazonlinux/amazonlinux:latest
ADD test.log /test.log

RUN yum upgrade -y && yum install -y python3

WORKDIR /usr/local/bin

COPY main.py .

CMD ["python3", "main.py"]
```

b. Salvare il file `Dockerfile`.

5. Usando il Dockerfile, crea un'immagine.

a. Crea l'immagine: `docker build -t multiline-app-image .`

Dove: `multiline-app-image` è il nome dell'immagine in questo esempio.

b. Verifica che l'immagine sia stata creata correttamente: `docker images –filter reference=multiline-app-image`

In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

6. Carica l'immagine nel registro del container Amazon Elastic.

a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name multiline-app-repo --region us-east-1`

Dove: `multiline-app-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

L'output fornisce i dettagli del nuovo repository. Annota il valore `repositoryUri` poiché sarà necessario nei passaggi successivi.

b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag` *`multiline-app-image repositoryUri`*

Esempio: `docker tag multiline-app-image xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images –filter reference=`*`repositoryUri`*

Nell'output, il nome del repository cambia da `multiline-app-repo` al valore `repositoryUri`.

d. Invia l'immagine ad Amazon ECR: `docker push` *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`repository name`*

Esempio: `docker push` *`xxxxxxxxxxxx`*`.dkr.ecr.`*`us-east-1`*`.amazonaws.com/multiline-app-repo`

Crea la definizione di un processo e avviare il processo

1. Crea un file di definizione del processo con il nome del file `multiline-task-definition.json`.
2. Incolla i contenuti seguenti nel file `multiline-task-definition.json`:

```
{
    "family": "firelens-example-multiline",
    "taskRoleArn": "task role ARN,
    "executionRoleArn": "execution role ARN",
    "containerDefinitions": [
        {
            "essential": true,
            "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-
multiline-image:latest",
            "name": "log_router",
            "firelensConfiguration": {
                "type": "fluentbit",
                "options": {
                    "config-file-type": "file",
                    "config-file-value": "/extra.conf"
                }
            },
            "memoryReservation": 50
        },
        {
            "essential": true,
            "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/multiline-app-
image:latest",
            "name": "app",
            "logConfiguration": {
                "logDriver": "awsfirelens",
                "options": {
                    "Name": "cloudwatch_logs",
                    "region": "us-east-1",
                    "log_group_name": "multiline-test/application",
                    "auto_create_group": "true",
                    "log_stream_prefix": "multiline-"
                }
            },
            "memoryReservation": 100
        }
    ],
    "requiresCompatibilities": ["FARGATE"],
    "networkMode": "awsvpc",
    "cpu": "256",
    "memory": "512"
}
```

Sostituisci quanto segue nella definizione del processo `multiline-task-definition.json`:

a. *`task role ARN`*

   Per trovare l'ARN del ruolo del processo, vai alla Console IAM. Scegli Roles (Roli) e trova il ruolo del processo `ecs-task-role-for-firelens` che hai creato. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

b. *`execution role ARN`*

   Per trovare l'ARN del ruolo di esecuzione, vai alla Console IAM. Scegli Roles (Ruoli) e trova il ruolo `ecsTaskExecutionRole`. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

c. *`aws_account_id`*

   Per trovare l'`aws_account_id`, accedi alla AWS Management Console. Scegli il tuo nome utente in alto a destra e copia il tuo ID account.

d. *`us-east-1`*

   Sostituisci la regione se necessario.

3. Registra il file di definizione del processo: `aws ecs register-task-definition --cli-input-json file://multiline-task-definition.json --region` *`region`*
4. Apri la console all'indirizzo 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 Crea.

Verifica che i messaggi di log multilinea in Amazon CloudWatch appaiano concatenati

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ CloudWatch .
2. Nel pannello di navigazione, espandi Logs (Log) e scegli Log groups (Gruppi di log).

3. Scegli il gruppo di log `multiline-test/applicatio`.
4. Scegli il log. Visualizza i messaggi. Le righe che corrispondono alle regole nel file parser vengono concatenate e appaiono come un singolo messaggio.

   Il seguente frammento di log mostra le righe concatenate in un singolo evento di traccia dello stack Java:

```
{
    "container_id": "xxxxxx",
    "container_name": "app",
    "source": "stdout",
    "log": "Dec 14 06:41:08 Exception in thread \"main\"
 java.lang.RuntimeException: Something has gone wrong, aborting!\n
 at com.myproject.module.MyProject.badMethod(MyProject.java:22)\n    at
 com.myproject.module.MyProject.oneMoreMethod(MyProject.java:18)\n
 at com.myproject.module.MyProject.anotherMethod(MyProject.java:14)\n
 at com.myproject.module.MyProject.someMethod(MyProject.java:10)\n    at
 com.myproject.module.MyProject.main(MyProject.java:6)",
    "ecs_cluster": "default",
    "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
    "ecs_task_definition": "firelens-example-multiline:2"
}
```

   Il seguente frammento di log mostra come viene visualizzato lo stesso messaggio con una singola riga se esegui un container Amazon ECS non configurato per concatenare messaggi di log multilinea.

```
{
    "log": "Dec 14 06:41:08 Exception in thread \"main\"
 java.lang.RuntimeException: Something has gone wrong, aborting!",
    "container_id": "xxxxxx-xxxxxx",
    "container_name": "app",
    "source": "stdout",
    "ecs_cluster": "default",
    "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
    "ecs_task_definition": "firelens-example-multiline:3"
}
```

Esempio: utilizzo di un parser integrato Fluent Bit

In questo esempio eseguirai i passaggi seguenti:

1. Crea e carica l'immagine per un container Fluent Bit.
2. Crea e carica l'immagine per un'applicazione demo multilinea che esegue, fallisce e genera una traccia dello stack multilinea.
3. Crea la definizione di un processo e avviare il processo.
4. Visualizza i log per verificare che i messaggi che si estendono su più righe appaiano concatenati.

Crea e carica l'immagine per un container Fluent Bit

Questa immagine includerà un file di configurazione che fa riferimento al parser Fluent Bit.

1. Crea una cartella con il nome `FluentBitDockerImage`.
2. All'interno della cartella `FluentBitDockerImage`, crea un file di configurazione personalizzato che fa riferimento al file parser integrato Fluent Bit.

   Per ulteriori informazioni sul file di configurazione personalizzato, consulta Specifica di un file di configurazione personalizzato nella Guida per gli sviluppatori di Amazon Elastic Container Service

   a. Incolla i contenuti seguenti nel file:

   ```
   [FILTER]
       name                  multiline
       match                 *
       multiline.key_content log
       multiline.parser      go
   ```

   b. Salva il file con nome `extra.conf`.
3. All'interno della cartella `FluentBitDockerImage`, crea il Dockerfile con l'immagine Fluent Bit e il parser e i file di configurazione creati.

   a. Incolla i contenuti seguenti nel file:

   ```
   FROM public.ecr.aws/aws-observability/aws-for-fluent-bit:latest
   ADD extra.conf /extra.conf
   ```

b. Salva il file con nome `Dockerfile`.

4. Utilizzando il Dockerfile, crea un'immagine Fluent Bit personalizzata con il file di configurazione personalizzato inclusi.

> **Note**
>
> È possibile posizionare il file di configurazione in qualsiasi punto dell'immagine Docker, a meno che `/fluent-bit/etc/fluent-bit.conf` questo percorso del file non venga utilizzato da FireLens.

a. Crea l'immagine: `docker build -t fluent-bit-multiline-image .`

   Dove: `fluent-bit-multiline-image` è il nome dell'immagine in questo esempio.

b. Verifica che l'immagine sia stata creata correttamente: `docker images –filter reference=fluent-bit-multiline-image`

   In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

5. Carica l'immagine Fluent Bit personalizzata in Amazon Elastic Container Registry.

a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name fluent-bit-multiline-repo --region us-east-1`

   Dove: `fluent-bit-multiline-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

   L'output fornisce i dettagli del nuovo repository.

b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag` *`fluent-bit-multiline-image repositoryUri`*

   Esempio: `docker tag fluent-bit-multiline-image xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-repo`

c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images –filter reference=`*`repositoryUri`*

   Nell'output, il nome del repository cambia da fluent-bit-multiline-repo a. `repositoryUri`

d. Effettua l'autenticazione su Amazon ECR eseguendo il comando `aws ecr get-login-password` e specificando l'ID del log in cui desideri eseguire l'autenticazione: `aws ecr get-login-password | docker login --username AWS --password-stdin registry ID.dkr.ecr.region.amazonaws.com`

   Esempio: `ecr get-login-password | docker login --username AWS --password-stdin xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com`

   Viene visualizzato un messaggio di accesso riuscito.

e. Invia l'immagine ad Amazon ECR: `docker push registry ID.dkr.ecr.region.amazonaws.com/repository name`

   Esempio: `docker push xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-multiline-repo`

Crea e carica l'immagine per un'applicazione demo multilinea

Questa immagine includerà un file di script Python che esegue l'applicazione e un file di log di esempio.

1. Crea una cartella denominata `multiline-app`: `mkdir multiline-app`
2. Crea un file di script Python.

   a. Nella cartella `multiline-app`, crea un file con il nome `main.py`.

   b. Incolla i contenuti seguenti nel file:

```
import os
import time
file1 = open('/test.log', 'r')
Lines = file1.readlines()

count = 0

for i in range(10):
    print("app running normally...")
    time.sleep(1)

# Strips the newline character
for line in Lines:
    count += 1
```

```
    print(line.rstrip())
print(count)
print("app terminated.")
```

c. Salvare il file `main.py`.

3. Crea un file di log di esempio.

a. Nella cartella `multiline-app`, crea un file con il nome `test.log`.

b. Incolla i contenuti seguenti nel file:

```
panic: my panic

goroutine 4 [running]:
panic(0x45cb40, 0x47ad70)
  /usr/local/go/src/runtime/panic.go:542 +0x46c fp=0xc42003f7b8 sp=0xc42003f710
 pc=0x422f7c
main.main.func1(0xc420024120)
  foo.go:6 +0x39 fp=0xc42003f7d8 sp=0xc42003f7b8 pc=0x451339
runtime.goexit()
  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003f7e0
 sp=0xc42003f7d8 pc=0x44b4d1
created by main.main
  foo.go:5 +0x58

goroutine 1 [chan receive]:
runtime.gopark(0x4739b8, 0xc420024178, 0x46fcd7, 0xc, 0xc420028e17, 0x3)
  /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc420053e30 sp=0xc420053e00
 pc=0x42503c
runtime.goparkunlock(0xc420024178, 0x46fcd7, 0xc, 0x1000f010040c217, 0x3)
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc420053e70 sp=0xc420053e30
 pc=0x42512e
runtime.chanrecv(0xc420024120, 0x0, 0xc420053f01, 0x4512d8)
  /usr/local/go/src/runtime/chan.go:506 +0x304 fp=0xc420053f20 sp=0xc420053e70
 pc=0x4046b4
runtime.chanrecv1(0xc420024120, 0x0)
  /usr/local/go/src/runtime/chan.go:388 +0x2b fp=0xc420053f50 sp=0xc420053f20
 pc=0x40439b
main.main()
  foo.go:9 +0x6f fp=0xc420053f80 sp=0xc420053f50 pc=0x4512ef
runtime.main()
  /usr/local/go/src/runtime/proc.go:185 +0x20d fp=0xc420053fe0 sp=0xc420053f80
 pc=0x424bad
```

```
runtime.goexit()
  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc420053fe8
 sp=0xc420053fe0 pc=0x44b4d1

goroutine 2 [force gc (idle)]:
runtime.gopark(0x4739b8, 0x4ad720, 0x47001e, 0xf, 0x14, 0x1)
  /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003e768 sp=0xc42003e738
 pc=0x42503c
runtime.goparkunlock(0x4ad720, 0x47001e, 0xf, 0xc420000114, 0x1)
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003e7a8 sp=0xc42003e768
 pc=0x42512e
runtime.forcegchelper()
  /usr/local/go/src/runtime/proc.go:238 +0xcc fp=0xc42003e7e0 sp=0xc42003e7a8
 pc=0x424e5c
runtime.goexit()
  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003e7e8
 sp=0xc42003e7e0 pc=0x44b4d1
created by runtime.init.4
  /usr/local/go/src/runtime/proc.go:227 +0x35

goroutine 3 [GC sweep wait]:
runtime.gopark(0x4739b8, 0x4ad7e0, 0x46fdd2, 0xd, 0x419914, 0x1)
  /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003ef60 sp=0xc42003ef30
 pc=0x42503c
runtime.goparkunlock(0x4ad7e0, 0x46fdd2, 0xd, 0x14, 0x1)
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003efa0 sp=0xc42003ef60
 pc=0x42512e
runtime.bgsweep(0xc42001e150)
  /usr/local/go/src/runtime/mgcsweep.go:52 +0xa3 fp=0xc42003efd8
 sp=0xc42003efa0 pc=0x419973
runtime.goexit()
  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003efe0
 sp=0xc42003efd8 pc=0x44b4d1
created by runtime.gcenable
  /usr/local/go/src/runtime/mgc.go:216 +0x58
one more line, no multiline
```

c. Salvare il file `test.log`.

4. All'interno della cartella `multiline-app`, crea il Dockerfile.

   a. Incolla i contenuti seguenti nel file:

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

```
ADD test.log /test.log

RUN yum upgrade -y && yum install -y python3

WORKDIR /usr/local/bin

COPY main.py .

CMD ["python3", "main.py"]
```

b. Salvare il file `Dockerfile`.

5. Usando il Dockerfile, crea un'immagine.

   a. Crea l'immagine: `docker build -t multiline-app-image .`

      Dove: `multiline-app-image` è il nome dell'immagine in questo esempio.

   b. Verifica che l'immagine sia stata creata correttamente: `docker images –filter reference=multiline-app-image`

      In caso di esito positivo, l'output mostra l'immagine e il tag `latest`.

6. Carica l'immagine nel registro del container Amazon Elastic.

   a. Crea un repository Amazon ECR per archiviare l'immagine: `aws ecr create-repository --repository-name multiline-app-repo --region us-east-1`

      Dove: `multiline-app-repo` è il nome del repository e `us-east-1` è la regione in questo esempio.

      L'output fornisce i dettagli del nuovo repository. Annota il valore `repositoryUri` poiché sarà necessario nei passaggi successivi.

   b. Applica un tag all'immagine con il valore `repositoryUri` ricavato dall'output precedente: `docker tag` *`multiline-app-image repositoryUri`*

      Esempio: `docker tag multiline-app-image xxxxxxxxxxxx.dkr.ecr.us-east-1.amazonaws.com/multiline-app-repo`

   c. Esegui l'immagine Docker per verificare che sia stata eseguita correttamente: `docker images –filter reference=`*`repositoryUri`*

      Nell'output, il nome del repository cambia da `multiline-app-repo` al valore `repositoryUri`.

d. Invia l'immagine ad Amazon ECR: `docker push` *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`repository name`*

   Esempio: `docker push` *`xxxxxxxxxxxx`*`.dkr.ecr.`*`us-east-1`*`.amazonaws.com/multiline-app-repo`

Crea la definizione di un processo e avviare il processo

1. Crea un file di definizione del processo con il nome del file `multiline-task-definition.json`.
2. Incolla i contenuti seguenti nel file `multiline-task-definition.json`:

```
{
    "family": "firelens-example-multiline",
    "taskRoleArn": "task role ARN,
    "executionRoleArn": "execution role ARN",
    "containerDefinitions": [
        {
            "essential": true,
            "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/fluent-bit-
multiline-image:latest",
            "name": "log_router",
            "firelensConfiguration": {
                "type": "fluentbit",
                "options": {
                    "config-file-type": "file",
                    "config-file-value": "/extra.conf"
                }
            },
            "memoryReservation": 50
        },
        {
            "essential": true,
            "image": "aws_account_id.dkr.ecr.us-east-1.amazonaws.com/multiline-app-
image:latest",
            "name": "app",
            "logConfiguration": {
                "logDriver": "awsfirelens",
                "options": {
                    "Name": "cloudwatch_logs",
                    "region": "us-east-1",
```

```
                    "log_group_name": "multiline-test/application",
                    "auto_create_group": "true",
                    "log_stream_prefix": "multiline-"
                }
            },
            "memoryReservation": 100
        }
    ],
    "requiresCompatibilities": ["FARGATE"],
    "networkMode": "awsvpc",
    "cpu": "256",
    "memory": "512"
}
```

Sostituisci quanto segue nella definizione del processo `multiline-task-definition.json`:

a. *`task role ARN`*

   Per trovare l'ARN del ruolo del processo, vai alla Console IAM. Scegli Roles (Roli) e trova il ruolo del processo `ecs-task-role-for-firelens` che hai creato. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

b. *`execution role ARN`*

   Per trovare l'ARN del ruolo di esecuzione, vai alla Console IAM. Scegli Roles (Ruoli) e trova il ruolo `ecsTaskExecutionRole`. Scegli il ruolo e copia l'ARN che appare nella sezione Summary (Riepilogo).

c. *`aws_account_id`*

   Per trovare l'`aws_account_id`, accedi alla AWS Management Console. Scegli il tuo nome utente in alto a destra e copia il tuo ID account.

d. *`us-east-1`*

   Sostituisci la regione se necessario.

3. Registra il file di definizione del processo: `aws ecs register-task-definition --cli-input-json file://multiline-task-definition.json --region us-east-1`
4. Apri la console all'indirizzo 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 Crea.

Verifica che i messaggi di log multilinea in Amazon CloudWatch appaiano concatenati

1. Apri la console all'indirizzo https://console.aws.amazon.com/cloudwatch/ 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\ngoroutine 4 [running]:\npanic(0x45cb40,
 0x47ad70)\n  /usr/local/go/src/runtime/panic.go:542 +0x46c fp=0xc42003f7b8
 sp=0xc42003f710 pc=0x422f7c\nmain.main.func1(0xc420024120)\n  foo.go:6
 +0x39 fp=0xc42003f7d8 sp=0xc42003f7b8 pc=0x451339\nruntime.goexit()\n  /usr/
local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003f7e0 sp=0xc42003f7d8
 pc=0x44b4d1\ncreated by main.main\n  foo.go:5 +0x58\n\ngoroutine 1 [chan receive]:
\nruntime.gopark(0x4739b8, 0xc420024178, 0x46fcd7, 0xc, 0xc420028e17, 0x3)\n  /usr/
local/go/src/runtime/proc.go:280 +0x12c fp=0xc420053e30 sp=0xc420053e00 pc=0x42503c
\nruntime.goparkunlock(0xc420024178, 0x46fcd7, 0xc, 0x1000f010040c217, 0x3)\n
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc420053e70 sp=0xc420053e30
 pc=0x42512e\nruntime.chanrecv(0xc420024120, 0x0, 0xc420053f01, 0x4512d8)\n
  /usr/local/go/src/runtime/chan.go:506 +0x304 fp=0xc420053f20 sp=0xc420053e70
 pc=0x4046b4\nruntime.chanrecv1(0xc420024120, 0x0)\n  /usr/local/go/src/runtime/
chan.go:388 +0x2b fp=0xc420053f50 sp=0xc420053f20 pc=0x40439b\nmain.main()\n
 foo.go:9 +0x6f fp=0xc420053f80 sp=0xc420053f50 pc=0x4512ef\nruntime.main()\n
  /usr/local/go/src/runtime/proc.go:185 +0x20d fp=0xc420053fe0 sp=0xc420053f80
 pc=0x424bad\nruntime.goexit()\n  /usr/local/go/src/runtime/asm_amd64.s:2337
 +0x1 fp=0xc420053fe8 sp=0xc420053fe0 pc=0x44b4d1\n\ngoroutine 2 [force gc
 (idle)]:\nruntime.gopark(0x4739b8, 0x4ad720, 0x47001e, 0xf, 0x14, 0x1)\n  /
usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc42003e768 sp=0xc42003e738
 pc=0x42503c\nruntime.goparkunlock(0x4ad720, 0x47001e, 0xf, 0xc420000114, 0x1)\n
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc42003e7a8 sp=0xc42003e768
 pc=0x42512e\nruntime.forcegchelper()\n  /usr/local/go/src/runtime/proc.go:238
```

```
 +0xcc fp=0xc42003e7e0 sp=0xc42003e7a8 pc=0x424e5c\nruntime.goexit()\n  /usr/
local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc42003e7e8 sp=0xc42003e7e0
 pc=0x44b4d1\ncreated by runtime.init.4\n  /usr/local/go/src/runtime/proc.go:227
 +0x35\n\ngoroutine 3 [GC sweep wait]:\nruntime.gopark(0x4739b8, 0x4ad7e0,
 0x46fdd2, 0xd, 0x419914, 0x1)\n  /usr/local/go/src/runtime/proc.go:280 +0x12c
 fp=0xc42003ef60 sp=0xc42003ef30 pc=0x42503c\nruntime.goparkunlock(0x4ad7e0,
 0x46fdd2, 0xd, 0x14, 0x1)\n  /usr/local/go/src/runtime/proc.go:286 +0x5e
 fp=0xc42003efa0 sp=0xc42003ef60 pc=0x42512e\nruntime.bgsweep(0xc42001e150)\n
  /usr/local/go/src/runtime/mgcsweep.go:52 +0xa3 fp=0xc42003efd8 sp=0xc42003efa0
 pc=0x419973\nruntime.goexit()\n  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1
 fp=0xc42003efe0 sp=0xc42003efd8 pc=0x44b4d1\ncreated by runtime.gcenable\n  /usr/
local/go/src/runtime/mgc.go:216 +0x58",
    "container_id": "xxxxxx-xxxxxx",
    "container_name": "app",
    "source": "stdout",
    "ecs_cluster": "default",
    "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
    "ecs_task_definition": "firelens-example-multiline:2"
}
```

Il seguente snippet di log mostra come viene visualizzato lo stesso evento se si esegue un container ECS non configurato per concatenare messaggi di log multilinea. Il campo di log contiene una singola riga.

```
{
    "log": "panic: my panic",
    "container_id": "xxxxxx-xxxxxx",
    "container_name": "app",
    "source": "stdout",
    "ecs_cluster": "default",
    "ecs_task_arn": "arn:aws:ecs:us-east-1:xxxxxxxxxxxx:task/default/xxxxxx",
    "ecs_task_definition": "firelens-example-multiline:3"
```

> **Note**
>
> Se i tuoi log passano ai file di log anziché all'output standard, ti consigliamo di specificare i parametri di configurazione `multiline.parser` e `multiline.key_content` nel plug-in Tail input anziché nel filtro.

Distribuzione di Fluent Bit su contenitori Amazon ECS Windows

Fluent Bit è un processore e router di log veloce e flessibile supportato da vari sistemi operativi. Può essere utilizzato per indirizzare i log verso varie AWS destinazioni come Amazon CloudWatch Logs, Firehose, Amazon S3 e Amazon Service. OpenSearch Fluent Bit supporta soluzioni partner comuni come Datadog, Splunk e server HTTP personalizzati. Per ulteriori informazioni su Fluent Bit, consulta il 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 dello spazio dei nomi dell'host.
  - L'agente Amazon ECS aggiunge le etichette ai container che includono il nome del cluster, l'ARN dell'attività, il nome della famiglia della definizione di attività, il numero di revisione della definizione di attività, l'ARN dell'attività e il nome del container delle attività. Le stesse informazioni vengono aggiunte al record di log utilizzando l'opzione labels del driver di registrazione fluentd docker. Per ulteriori informazioni, consulta labels, labels-regex, env ed env-regex sul sito Web contenente la documentazione di Docker.

- Poiché l'opzione `async` del driver di registrazione fluentd è impostata su `true`, quando il container Fluent Bit viene riavviato, docker memorizza i log nel buffer fino al riavvio del container Fluent Bit. Puoi aumentare il limite del buffer impostando l'opzione. fluentd-buffer-limit Per ulteriori informazioni, consulta fluentd-buffer-limit sul sito Web contenente la documentazione di Docker.

Di seguito è riportato il flusso di lavoro:

- Il container di Fluent Bit si avvia e ascolta sulla porta 24224 che è esposta all'host.
- Fluent Bit utilizza le credenziali del ruolo IAM dell'attività specificate nella definizione di attività.
- Altre attività avviate sulla stessa istanza utilizzano il driver di registrazione fluentd docker per connettersi al container di Fluent Bit sulla porta 24224.
- Quando i container dell'applicazione generano log, il runtime docker aggiunge un tag a tali record, aggiunge altri metadati specificati nelle etichette e quindi li inoltra sulla porta 24224 dello spazio dei nomi dell'host.
- Fluent Bit riceve il record di log sulla porta 24224 perché è esposto allo spazio dei nomi dell'host.
- Fluent Bit esegue l'elaborazione interna e indirizza i log come specificato.

Questo tutorial utilizza la configurazione CloudWatch Fluent Bit predefinita che esegue le seguenti operazioni:

- Crea un nuovo gruppo di log per ogni cluster e famiglia di definizione di attività.
- Crea un nuovo flusso di log per ogni container di attività nel gruppo di log sopra generato ogni volta che viene avviata una nuova attività. Ogni flusso verrà contrassegnato con l'ID dell'attività a cui appartiene il container.
- Aggiunge altri metadati tra cui il nome del cluster, l'ARN dell'attività, il nome del container di attività, la famiglia di definizioni di attività e il numero di revisione della definizione di attività in ogni voce del log.

  Ad esempio, se hai `task_1` with `container_1` `container_2` e t `ask_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`

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 i log CloudWatch
- Fase 9: Pulizia

Prerequisiti

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- La versione più recente di AWS CLI è installata e configurata. Per ulteriori informazioni, consulta l'argomento relativo all'installazione di AWS Command Line Interface.
- 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'istanza EC2. È possibile utilizzare il VPC di default. Puoi anche utilizzare una sottorete privata che consente agli CloudWatch endpoint Amazon di raggiungere la sottorete. Per ulteriori informazioni sugli CloudWatch endpoint Amazon, consulta la sezione CloudWatch Endpoints e quote Amazon nel. Riferimenti generali di AWS Per ulteriori informazioni sull'utilizzo della procedura guidata di Amazon VPC per creare un VPC, consulta the section called “Crea un cloud privato virtuale”.

Fase 1: Creazione dei ruoli di accesso IAM

Crea i ruoli IAM di Amazon ECS.

1. Crea il ruolo dell'istanza del contenitore Amazon ECS denominato «ecsInstanceRole». Per ulteriori informazioni, consulta Ruolo IAM delle istanze di container di Amazon ECS.
2. Crea un ruolo IAM per l'attività Fluent Bit denominata `fluentTaskRole`. Per ulteriori informazioni, consulta the section called “Ruolo IAM del processo”.

   Le autorizzazioni IAM concesse nel ruolo IAM sono assunte dai container delle attività. Per consentire a Fluent Bit di inviare i log a CloudWatch, devi assegnare le seguenti autorizzazioni al ruolo IAM dell'attività.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
       {
           "Effect": "Allow",
           "Action": [
               "logs:CreateLogStream",
               "logs:CreateLogGroup",
               "logs:DescribeLogStreams",
               "logs:PutLogEvents"
           ],
           "Resource": "*"
       }
       ]
   }
   ```

3. Collegare la policy al ruolo.

   a. Salva il contenuto di cui sopra in un file denominato `fluent-bit-policy.json`.

   b. Per collegare la policy in linea al ruolo IAM `fluentTaskRole`, esegui il comando riportato.

```
aws iam put-role-policy --role-name fluentTaskRole --policy-name
 fluentTaskPolicy --policy-document file://fluent-bit-policy.json
```

Fase 2: Creazione di un'istanza di container Windows di Amazon ECS

Creazione di un'istanza di container Windows di Amazon ECS

Creazione di un'istanza Amazon ECS

1. Usa il comando `aws ssm get-parameters` per recuperare l'ID AMI per la regione che ospita il tuo VPC. Per ulteriori informazioni, consulta Recupero dei metadati dell'AMI ottimizzata per Amazon ECS.
2. Utilizza la console di Amazon EC2 per avviare l'istanza.
   a. Apri la console Amazon EC2 all'indirizzo https://console.aws.amazon.com/ec2/.
   b. Seleziona la Regione da utilizzare nella barra di navigazione.
   c. Da Pannello di controllo EC2, scegli Avvia istanza.
   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 ecs. InstanceRole
   k. Configura la tua istanza di container Amazon ECS con i seguenti dati utente. In Advanced Details (Dettagli avanzati), incolla il seguente script nel campo User data (Dati utente), sostituendo *`cluster_name`* con il nome del tuo 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

Puoi 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:

- *`regione`* 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                $ecs_task_arn ^([a-z-:0-9]+)/([a-zA-Z0-9-_]+)/([a-z0-9]+)$
 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 log nel formato out.*ID ATTIVITÀ*.*NOME_CONTAINER*.
- Il nuovo tag verrà indirizzato al plug-in cloudwatch_logs di output che crea i gruppi di log e gli stream come descritto in precedenza utilizzando le opzioni e del plug-in di output. `log_group_template` `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'istanza EC2 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 il nome della risorsa Amazon (ARN) del ruolo IAM dell'attività
   - *`region`* con la regione in cui viene eseguita l'attività

```
{
  "family": "ecs-windows-fluent-bit",
  "taskRoleArn": "task-iam-role",
  "containerDefinitions": [
    {
      "name": "fluent-bit",
      "image": "public.ecr.aws/aws-observability/aws-for-fluent-
bit:windowsservercore-latest",
      "cpu": 512,
      "portMappings": [
```

```
        {
          "hostPort": 24224,
          "containerPort": 24224,
          "protocol": "tcp"
        }
      ],
      "entryPoint": [
        "Powershell",
        "-Command"
      ],
      "command": [
        "C:\\entrypoint.ps1 -ConfigFile C:\\ecs_windows_forward_daemon\
\cloudwatch.conf"
      ],
      "environment": [
        {
          "name": "AWS_REGION",
          "value": "region"
        }
      ],
      "memory": 512,
      "essential": true,
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-group": "/ecs/fluent-bit-logs",
          "awslogs-region": "region",
          "awslogs-stream-prefix": "flb",
          "awslogs-create-group": "true"
        }
      }
    }
  ],
  "memory": "512",
  "cpu": "512"
}
```

2. Utilizza il comando seguente per registrare la definizione dell'attività.

```
aws ecs register-task-definition --cli-input-json file://fluent-bit.json --
region region
```

Puoi sempre ottenere un elenco delle definizioni di attività per l'account tramite il comando `list-task-definitions`. L'output che mostra i valori della famiglia e delle revisioni che è possibile utilizzare insieme a `run-task` o `start-task`.

Fase 5: Esecuzione della definizione di attività **`ecs-windows-fluent-bit`** come servizio Amazon ECS utilizzando la strategia di pianificazione del daemon

Dopo aver registrato una definizione di attività per l'account, puoi eseguire un'attività nel cluster. Per questo tutorial, esegui una istanza della definizione di attività `ecs-windows-fluent-bit:1` nel cluster `FluentBit-cluster`. Esegui l'attività in un servizio che utilizza la strategia di pianificazione dei daemon, che garantisce che una singola istanza di Fluent Bit venga sempre eseguita su ciascuna delle istanze di container.

Per eseguire un'attività

1. Esegui il comando seguente per avviare la definizione di attività `ecs-windows-fluent-bit:1` (registrata nel passaggio precedente) come servizio.

   > **Note**
   >
   > Questa definizione di attività utilizza il driver di registrazione `awslogs`; l'istanza di container deve disporre delle autorizzazioni necessarie.

   Sostituisci le seguenti variabili:

   - *`region`* con la regione in cui viene eseguito il servizio

   ```
   aws ecs create-service \
       --cluster FluentBit-cluster \
       --service-name FluentBitForwardDaemonService \
       --task-definition ecs-windows-fluent-bit:1 \
       --launch-type EC2 \
       --scheduling-strategy DAEMON \
       --region region
   ```

2. Per elencare le attività, esegui il comando riportato.

   Sostituisci le seguenti variabili:

   - *region* con la regione in cui vengono eseguite le attività di servizio

```
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 i log 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 nella Guida di riferimento dell'AWS Command Line Interface .

```
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 Linux contenitori EC2 su Amazon ECS

Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux su EC2 tramite un tipo speciale di account di servizio chiamato Managed Service Account di 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.

Un container Linux che viene eseguito con gMSA si basa sul daemon `credentials-fetcher` che viene eseguito sull'istanza Amazon EC2 host del container. 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 container vengono eseguiti su EC2, puoi utilizzare gMSA per i container Windows e Linux. Per informazioni su come utilizzare gMSA il contenitore Linux su Fargate, vedere. Utilizzo gMSA per i Linux container 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 perWindows. 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.

  Opzioni di archiviazione di dati gMSA per sistema operativo

Posizione di archiviazione	Linux	Windows
Amazon Simple Storage Service	CredSpec	CredSpec
AWS Secrets Manager	credenziali utente senza dominio	credenziali utente senza dominio
Archivio dei parametri Systems Manager per Amazon EC2	CredSpec	CredSpec, credenziali utente senza dominio

Posizione di archiviazione	Linux	Windows
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 AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida introduttiva a AWS Managed Microsoft AD nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta AWS Direct Connect.
- Disponi di un account gMSA nell'Active Directory. Per ulteriori informazioni, consulta Utilizzo gMSA per Linux contenitori EC2 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. AWS Secrets Manager
- Se archivi la specifica delle credenziali nell'archivio dei parametri SSM, le autorizzazioni IAM per l'archivio dei parametri Systems Manager per Amazon EC2 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 fornire le credenziali iniziali e configura i dati utente EC2 in un modello di avvio EC2 riutilizzabile per installare il daemon `credentials-fetcher`.

1. Decidi se aggiungere le istanze al dominio o utilizzare gMSA senza dominio.
   - Aggiunta di istanze EC2 al dominio di Active Directory
     - Aggiunta delle istanze in base ai dati dell'utente

       Completa i passaggi per aggiungere il dominio Active Directory ai dati utente EC2 in un modello di avvio EC2. Più gruppi con dimensionamento automatico Amazon EC2 possono utilizzare lo stesso modello di avvio.

       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'istanza EC2 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 AWS Secrets Manager 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 AWS Secrets Manager, dopo aver creato l'utente in Active Directory. Per ulteriori informazioni, consulta Creare un AWS Secrets Manager segreto.

c. Inserisci il nome utente, la password e il dominio dell'utente nelle coppie chiave-valore JSON denominate rispettivamente `username`, `password` e `domainName`.

```
{"username":"username","password":"passw0rd", "domainName":"example.com"}
```

d. 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 daemon `credentials-fetcher` con uno script di dati utente nel tuo modello di avvio Amazon EC2. Gli esempi seguenti mostrano due tipi di dati utente, `cloud-config` YAML o lo script bash. Questi esempi si applicano ad 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-fetch && 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-fetch && 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`.

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": [
                   "secretsmanager:GetSecretValue"
               ],
               "Resource": [
                   "arn:aws:ssm:aws-region:111122223333:secret:MySecret"
               ]
           }
       ]
   }
   ```

   > **Note**
   >
   > Se utilizzi la tua chiave KMS per crittografare il tuo segreto, devi aggiungere le autorizzazioni necessarie a questo ruolo e aggiungere questo ruolo alla politica delle chiavi. AWS KMS

2. 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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
```

```
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/{object}"
            ]
        }
    ]
}
```

- Parametro dell'archivio parametri di SSM

  Aggiungi la specifica delle credenziali a un parametro SSM Parameter Store. Quindi, fai riferimento al nome della risorsa Amazon (ARN) del parametro SSM Parameter Store nel campo `credentialSpecs` della definizione delle attività.

```
{
    "family": "",
    "executionRoleArn": "",
    "containerDefinitions": [
        {
            "name": "",
            ...
            "credentialSpecs": [
                "credentialspecdomainless:arn:aws:ssm:aws-
region:111122223333:parameter/parameter_name"
            ],
            ...
        }
    ],
    ...
}
```

  Aggiungi le seguenti autorizzazioni come policy inline al ruolo IAM di esecuzione delle attività Amazon ECS per consentire alle attività l'accesso al parametro SSM Parameter Store.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
```

```
            "Resource": [
                "arn:aws:ssm:aws-region:111122223333:parameter/parameter_name"
            ]
        }
    ]
}
```

File di specifica delle credenziali

Amazon ECS utilizza un file di 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 gMSA per i Linux container su Fargate

Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux su Fargate tramite un tipo speciale di account di servizio chiamato Managed Service Account di 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

Considerate quanto segue prima di utilizzarli gMSA per i Linux contenitori su Fargate:

- È necessario utilizzare la versione 1.4 o successiva della piattaforma.
- 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 perWindows. Per ulteriori informazioni, consulta Installazione degli strumenti di amministrazione Active Directory.
- È necessario utilizzare la modalità domainless gMSA.

  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 CredSpec file viene generato e quindi archiviato 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 AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida introduttiva a AWS Managed Microsoft AD nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta AWS Direct Connect.
- Disponi di un gMSA account esistente in Active Directory e di un utente che dispone dell'autorizzazione per accedere all'account del gMSA servizio. Per ulteriori informazioni, consulta Creazione di un utente Active Directory per gMSA senza dominio.
- Hai un bucket Amazon S3. Per ulteriori informazioni, consulta Creare un 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 utilizzi domainlessgMSA, il contenitore non viene aggiunto al dominio. Le altre applicazioni eseguite sul contenitore non possono utilizzare le credenziali per accedere al dominio. Le attività che utilizzano un dominio diverso possono essere eseguite sullo stesso contenitore. Fornisci il nome di un segreto AWS Secrets Manager nel CredSpec file. Il segreto deve contenere un nome utente, una password e un dominio a cui accedere.

  Questa funzionalità è simile alla 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. Configura un utente nel tuo dominio Active Directory. L'utente di Active Directory deve disporre dell'autorizzazione per accedere all'account di gMSA servizio utilizzato nelle attività.

b. Disponi di un VPC e delle sottoreti in grado di risolvere il nome di dominio Active Directory. Configura il VPC con opzioni DHCP con il nome di dominio che punta al nome del servizio Active Directory. Per informazioni su come configurare le opzioni DHCP per un VPC, consulta Work with DHCP option sets nella Amazon Virtual Private Cloud User Guide.

c. Crea un account segreto in. AWS Secrets Manager

d. Crea il file con le specifiche delle credenziali.

Configurazione di autorizzazioni e segreti

Esegui i passaggi seguenti una volta per ogni applicazione e ogni definizione di attività. Consigliamo di utilizzare la best practice di concedere il privilegio minimo e limitare le autorizzazioni utilizzate nella policy. In questo modo, ogni attività può leggere solo i segreti di cui ha bisogno.

1. Crea un utente nel dominio di Active Directory. L'utente 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 AWS Secrets Manager. Per ulteriori informazioni, consulta Creazione di un segreto AWS Secrets Manager.
3. Inserisci il nome utente, la password e il dominio dell'utente nelle coppie chiave-valore JSON denominate rispettivamente `username`, `password` e `domainName`.

```
{"username":"username","password":"passw0rd", "domainName":"example.com"}
```

4. Inoltre, devi aggiungere le autorizzazioni seguenti come policy inline al ruolo di esecuzione dell'attività del ruolo IAM. In questo modo il daemon `credentials-fetcher` accede al segreto di Secrets Manager. Sostituisci l'esempio `MySecret` con il nome della risorsa Amazon (ARN) del segreto nell'elenco `Resource`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:aws-region:111122223333:secret:MySecret"
      ]
```

```
    }
  ]
}
```

> **Note**
>
> Se utilizzi la tua chiave KMS per crittografare il tuo segreto, devi aggiungere le autorizzazioni necessarie a questo ruolo e aggiungere questo ruolo alla politica delle AWS KMS chiavi.

5. Aggiungi le specifiche delle credenziali a un bucket Amazon S3. Quindi, fai riferimento al nome della risorsa Amazon (ARN) del bucket Amazon S3 nel campo `credentialSpecs` della definizione di attività.

```
{
    "family": "",
    "executionRoleArn": "",
    "containerDefinitions": [
        {
            "name": "",
            ...
            "credentialSpecs": [
                "credentialspecdomainless:arn:aws:s3:::${BucketName}/${ObjectName}"
            ],
            ...
        }
    ],
    ...
}
```

Devi aggiungere le seguenti autorizzazioni come policy inline al ruolo IAM di esecuzione delle attività Amazon ECS per consentire alle attività l'accesso al bucket Amazon S3.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
```

```
                "s3:ListObject"
            ],
            "Resource": [
                "arn:aws:s3:::{bucket_name}",
                "arn:aws:s3:::{bucket_name}/{object}"
            ]
        }
    ]
}
```

File di specifica delle credenziali

Amazon ECS utilizza un file di 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"
            }
        }
    }
}
```

Creare un file CredSpec e caricarlo su un 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 delle specifiche delle credenziali, caricalo in 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 `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.

Per PowerShell, usa il seguente comando:

```
$ Write-S3Object -BucketName "MyBucket" -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://MyBucket/ecs-domainless-gmsa-credspec
```

Utilizzo di contenitori Amazon ECS Windows con modalità domainless 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

Questo tutorial presuppone che siano stati soddisfatti i prerequisiti seguenti:

- Hai completato le fasi descritte in Configurazione per l'uso di Amazon ECS.
- AWS L'utente dispone delle autorizzazioni richieste specificate nell'esempio di policy IAM. AmazonECS_ 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

- Configura un dominio Active Directory con le risorse a cui desideri che i tuoi container accedano. Amazon ECS supporta le configurazioni seguenti:
  - Un AWS Directory Service Active Directory. AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida introduttiva a AWS Managed Microsoft AD nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta AWS Direct Connect.
- 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.

  Opzioni di archiviazione di dati gMSA per sistema operativo

Posizione di archiviazione	Linux	Windows
Amazon Simple Storage Service	CredSpec	CredSpec
AWS Secrets Manager	credenziali utente senza dominio	credenziali utente senza dominio
Archivio dei parametri Systems Manager per Amazon EC2	CredSpec	CredSpec, credenziali utente senza dominio

Posizione di archiviazione	Linux	Windows
File locale	N/D	CredSpec

- (Facoltativo) AWS CloudShell è uno strumento che offre ai clienti una riga di comando senza la necessità di creare la propria istanza EC2. 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.

1. Genera una chiave principale del servizio di distribuzione delle chiavi

   > **Note**
   >
   > Se lo stai utilizzando AWS Directory Service, puoi saltare questo passaggio.

   La chiave principale KDS e le autorizzazioni gMSA sono configurate con Microsoft AD gestito da AWS .

   Se non hai ancora creato un account del sevizio 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'gMSAaccount e consentire il `ccg.exe` recupero della gMSA password, esegui i seguenti PowerShell comandi da un server o client Windows con accesso al dominio. Sostituisci `ExampleAccount` con il nome che desideri assegnare all'account gMSA.

   a.
   ```
   PS C:\> Install-WindowsFeature RSAT-AD-PowerShell
   ```

   b.
   ```
   PS C:\> New-ADGroup -Name "ExampleAccount Authorized Hosts" -SamAccountName
    "ExampleAccountHosts" -GroupScope DomainLocal
   ```

   c.
   ```
   PS C:\> New-ADServiceAccount -Name "ExampleAccount" -DnsHostName
    "contoso" -ServicePrincipalNames "host/ExampleAccount", "host/contoso" -
   PrincipalsAllowedToRetrieveManagedPassword "ExampleAccountHosts"
   ```

   d. Crea un utente con una password permanente senza scadenza. Queste credenziali vengono archiviate AWS Secrets Manager e utilizzate da ogni attività per accedere al dominio.

   ```
   PS C:\> New-ADUser -Name "ExampleAccount" -AccountPassword (ConvertTo-
   SecureString -AsPlainText "Test123" -Force) -Enabled 1 -PasswordNeverExpires 1
   ```

   e.
   ```
   PS C:\> Add-ADGroupMember -Identity "ExampleAccountHosts" -Members
    "ExampleAccount"
   ```

   f. Installa il PowerShell modulo per la creazione 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. 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\":\"ExampleAccount\",\"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": "contoso",
       "DnsName": "contoso",
       "NetBiosName": "contoso"
     },
     "ActiveDirectoryConfig": {
       "GroupManagedServiceAccounts": [
         {
           "Name": "ExampleAccount",
           "Scope": "contoso"
         },
         {
           "Name": "ExampleAccount",
           "Scope": "contoso"
   ```

```
      }
    ],
    "HostAccountConfig": {
      "PortableCcgVersion": "1",
      "PluginGUID": "{859E1386-BDB4-49E8-85C7-3070B13920E1}",
      "PluginInput": "{\"credentialArn\": \"arn:aws:secretsmanager:aws-
region:111122223333:secret:gmsa-plugin-input\"}"
    }
  }
}
```

Fase 4: 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 negli AWS CLI SDK e AWS CloudFormation. Puoi utilizzare cluster aggiuntivi per raggruppare e organizzare le attività e l'infrastruttura e assegnare impostazioni predefinite per alcune configurazioni.

È possibile creare un cluster dagli AWS Management Console, AWS CLI, SDK o. AWS 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 di container è un computer host per l'esecuzione di container in attività ECS, ad esempio istanze Amazon EC2. Ogni istanza di container viene registrata in un cluster Amazon ECS. Prima di avviare le istanze Amazon EC2 e registrarle in un cluster, devi creare un ruolo IAM per le istanze di container da utilizzare.

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 crittografa del segreto in Secrets Manager con una chiave personalizzata, devi consentire anche `kms:Decrypt` per la chiave.

   Se al posto di Secrets Manager utilizzi l'archivio parametri SSM, devi consentire `ssm:GetParameter` per il parametro, anziché `secretsmanager:GetSecretValue`.

   ```
   {
   "Version": "2012-10-17",
     "Statement": [
       {
         "Effect": "Allow",
         "Action": [
           "s3:GetObject"
         ],
         "Resource": "arn:aws:s3:::MyBucket/ecs-domainless-gmsa-credspec/gmsa-cred-
   spec.json"
       },
       {
         "Effect": "Allow",
         "Action": [
           "secretsmanager:GetSecretValue"
         ],
         "Resource": "arn:aws:secretsmanager:aws-region:111122223333:secret:gmsa-
   plugin-input"
       }
     ]
   }
   ```

Fase 8: creazione di un ruolo dell'attività per Amazon ECS Exec

Questo tutorial utilizza Amazon ECS Exec per verificare la funzionalità eseguendo un comando all'interno di un'attività in esecuzione. Per utilizzare ECS Exec, è necessario attivare ECS Exec nel servizio o nell'attività. Inoltre, il ruolo dell'attività (non il ruolo di esecuzione dell'attività) deve disporre delle autorizzazioni `ssmmessages`. Per informazioni sulla policy IAM richiesta, consulta Autorizzazioni ECS Exec.

Questo passaggio utilizza il. AWS CLI Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero `bash`.

Per creare un ruolo di attività utilizzando il AWS CLI, procedi nel seguente modo.

1. Crea un file denominato `ecs-tasks-trust-policy.json` con i seguenti contenuti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "ecs-tasks.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Crea un ruolo IAM. Puoi sostituire il nome `ecs-exec-demo-task-role` ma mantenerlo per i passaggi successivi.

   Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da `sh` e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws iam create-role --role-name ecs-exec-demo-task-role \
--assume-role-policy-document file://ecs-tasks-trust-policy.json
```

   Puoi eliminare il file `ecs-tasks-trust-policy.json`.

3. Crea un file denominato `ecs-exec-demo-task-role-policy.json` con i seguenti contenuti:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssmmessages:CreateControlChannel",
                "ssmmessages:CreateDataChannel",
```

```
                "ssmmessages:OpenControlChannel",
                "ssmmessages:OpenDataChannel"
            ],
            "Resource": "*"
        }
    ]
}
```

4. Crea una policy IAM e collegala al ruolo della fase precedente.

   Il comando seguente utilizza i caratteri di continuazione (barra rovesciata) utilizzati da `sh` e dalle shell compatibili. Questo comando non è compatibile con PowerShell. È necessario modificare il comando con cui utilizzarlo PowerShell.

```
$ aws iam put-role-policy \
    --role-name ecs-exec-demo-task-role \
    --policy-name ecs-exec-demo-task-role-policy \
    --policy-document file://ecs-exec-demo-task-role-policy.json
```

   Puoi eliminare il file `ecs-exec-demo-task-role-policy.json`.

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 Windows di Amazon EC2 ed esegui l'agente di container ECS per registrarla come istanza di container nel cluster. ECS esegue le attività sulle istanze di container registrate nel cluster in cui vengono avviate le attività.

1. Per avviare un'istanza Windows di Amazon EC2 configurata per Amazon ECS in, consulta AWS Management Console. 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 IMDS EC2 al fine di recuperare le credenziali del ruolo, aggiungi l'argomento `-AwsvpcBlockIMDS`. Ciò si applica solo alle attività che utilizzano la modalità di rete `awsvpc`.

   ```
   <powershell>
   [Environment]::SetEnvironmentVariable("ECS_GMSA_SUPPORTED", $TRUE, "Machine")
   Import-Module ECSTools
   Initialize-ECSAgent -Cluster windows-domainless-gmsa-cluster -EnableTaskIAMRole -
   AwsvpcBlockIMDS
   </powershell>
   ```

3. Analizza un riepilogo della configurazione dell'istanza nel pannello Summary (Riepilogo) e, quando è tutto pronto, scegli Launch instance (Avvia istanza).

Fase 11: verifica dell'istanza di container

Puoi verificare la presenza di un'istanza di container nel cluster utilizzando la AWS Management Console. Tuttavia, gMSA necessita di funzionalità aggiuntive indicate come attributi. Questi attributi non sono visibili in AWS Management Console, quindi questo tutorial utilizza il. AWS CLI

Questo passaggio utilizza il AWS CLI. Puoi eseguire questi comandi in AWS CloudShell nella shell (interprete di comandi) predefinita, ovvero `bash`.

1. Elenca le istanze di container nel cluster. Le istanze di container hanno un ID diverso dall'ID dell'istanza EC2.

```
$ 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'istanza Amazon EC2. Successivamente, l'istanza di container nel cluster verrà eliminata automaticamente dopo un'ora.

   Puoi individuare e terminare l'istanza mediante la console Amazon EC2 o puoi eseguire il comando seguente. Per eseguire il comando, trova l'ID dell'istanza EC2 nell'output del comando `aws ecs describe-container-instances` della fase 1, Fase 11: verifica dell'istanza di container .i-10a64379è un esempio di ID dell'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 presenza un problema, viene interrotta e impostata nello stato `STOPPED`. Esistono due tipi comuni di problemi relativi alle attività. Innanzitutto, le attività che non possono essere avviate. In secondo luogo, le attività in cui l'applicazione si è interrotta all'interno di uno dei container. Nel campo Motivo interrotto dell'attività AWS Management Console, esamina il motivo per cui l'attività è stata interrotta. Nella AWS CLI, descrivi l'attività e osserva il campo `stoppedReason`. Per i passaggi da seguire AWS CLI, consultaVisualizzazione degli errori delle attività interrotte da Amazon ECS. AWS Management Console

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 gMSA per i container Windows nel 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 GMSAS per contenitori 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 basa sulla relativa istanza host Amazon EC2 per recuperare le credenziali gMSA dal controller di dominio Active Directory e fornirle all'istanza del container. Per ulteriori informazioni, consulta l'argomento relativo alla creazione di account gMSA per container Windows.

> **Note**
>
> Questa funzione non è supportata per i container Windows su Fargate.

Argomenti

Considerazioni

Quando si utilizzano gli account gMSA per container Windows, è opportuno 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.

Opzioni di archiviazione di dati gMSA per sistema operativo

Posizione di archiviazione	Linux	Windows
Amazon Simple Storage Service	CredSpec	CredSpec
AWS Secrets Manager	credenziali utente senza dominio	credenziali utente senza dominio
Archivio dei parametri Systems Manager per Amazon EC2	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 Active Directory. AWS Directory Service AWS Directory Service è un Active Directory AWS gestito ospitato su Amazon EC2. Per ulteriori informazioni, vedere Guida introduttiva a AWS Managed Microsoft AD nella Guida all'AWS Directory Service amministrazione.
  - Una Active Directory on-premise. Devi assicurarti che l'istanza di container Linux di Amazon ECS possa essere aggiunta al dominio. Per ulteriori informazioni, consulta AWS Direct Connect.
- Disponi di un account gMSA nell'Active Directory. Per ulteriori informazioni, consulta l'argomento relativo alla creazione di account gMSA per container Windows.
- 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, AWS Secrets Manager sono necessarie le autorizzazioni IAM per il ruolo dell'istanza Amazon EC2.
  - Se archivi la specifica delle credenziali nell'archivio dei parametri SSM, le autorizzazioni IAM per l'archivio dei parametri Systems Manager per Amazon EC2 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 con modalità domainless 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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor",
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::{bucket_name}",
                "arn:aws:s3:::{bucket_name}/{object}"
            ]
        }
    ]
}
```

Parametro dell'archivio parametri di SSM

Aggiungi le specifiche delle credenziali a un parametro dell'archivio parametri di SSM e quindi fai riferimento all'Amazon Resource Name (ARN) del parametro dell'archivio parametri di SSM nel campo `credentialSpecs` della definizione di attività.

```
{
    "family": "",
    "executionRoleArn": "",
    "containerDefinitions": [
        {
            "name": "",
            ...
            "credentialSpecs": [

 "credentialspecdomainless:arn:aws:ssm:region:111122223333:parameter/parameter_name"
            ],
            ...
        }
    ],
    ...
}
```

È inoltre necessario aggiungere le seguenti autorizzazioni come policy in linea al ruolo IAM di esecuzione del processo di Amazon ECS per consentire ai processi l'accesso al parametro dell'archivio parametri di SSM.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters"
            ],
            "Resource": [
                "arn:aws:ssm:region:111122223333:parameter/parameter_name"
            ]
        }
    ]
}
```

File locale

Con i dettagli delle specifiche delle credenziali in un file locale, fai riferimento al percorso del file nel campo `credentialSpecs` della definizione di attività. Il percorso del file a cui si fa riferimento deve essere relativo alla directory `C:\ProgramData\Docker\CredentialSpecs` e utilizzare la barra rovesciata ("\") come separatore del percorso del file.

```
{
    "family": "",
    "executionRoleArn": "",
    "containerDefinitions": [
        {
            "name": "",
            ...
            "credentialSpecs": [
                "credentialspec:file://CredentialSpecDir\CredentialSpecFile.json"
            ],
            ...
        }
    ],
    ...
}
```

Utilizzo di EC2 Image Builder per creare AMI personalizzate ottimizzate per Amazon ECS

AWS consiglia di utilizzare le AMI ottimizzate per Amazon ECS perché sono preconfigurate con i requisiti e i consigli per eseguire i carichi di lavoro dei container. In alcuni casi potrebbe essere necessario personalizzare l'AMI per aggiungere software aggiuntivo. È possibile utilizzare EC2 Image Builder per la creazione, la gestione e l'implementazione delle immagini del server. Mantieni la proprietà delle immagini personalizzate create nel tuo account. È possibile utilizzare le pipeline di EC2 Image Builder per automatizzare gli aggiornamenti e l'applicazione di patch di sistema per le immagini oppure utilizzare un comando autonomo per creare un'immagine con le risorse di configurazione definite.

Crei una ricetta per la tua immagine. La ricetta include un'immagine principale ed eventuali componenti aggiuntivi. Inoltre, crei una pipeline che distribuisce la tua AMI personalizzata.

Crei una ricetta per la tua immagine. Una ricetta di immagini di Image Builder è un documento che definisce l'immagine di base e i componenti che vengono applicati all'immagine di base per produrre la configurazione desiderata per l'immagine AMI di output. Inoltre, crei una pipeline che distribuisce la tua AMI personalizzata. Per ulteriori informazioni, consulta Come funziona EC2 Image Builder nella Guida per l'utente di EC2 Image Builder.

Ti consigliamo di utilizzare una delle seguenti AMI ottimizzate per Amazon ECS come «immagine principale» in EC2 Image Builder:

- Linux
  - AL2023 x86 ottimizzato per Amazon ECS
  - AMI Amazon Linux 2023 (arm64) ottimizzata per Amazon ECS
  - AMI Amazon Linux 2 kernel 5 ottimizzata per Amazon ECS
  - AMI Amazon Linux 2 x86 ottimizzata per Amazon ECS
- Windows
  - Windows 2022 Full x86 ottimizzato per Amazon ECS
  - Windows 2022 Core x86 ottimizzato per Amazon ECS
  - Windows 2019 Full x86 ottimizzato per Amazon ECS
  - Windows 2019 Core x86 ottimizzato per Amazon ECS
  - Windows 2016 completo x86 ottimizzato per Amazon ECS

Ti consigliamo inoltre di selezionare «Usa l'ultima versione del sistema operativo disponibile». La pipeline utilizzerà il controllo delle versioni semantiche per l'immagine principale, che aiuta a rilevare gli aggiornamenti delle dipendenze nei lavori pianificati automaticamente. Per ulteriori informazioni, consulta il controllo delle versioni semantiche nella Guida per l'utente di EC2 Image Builder.

AWS aggiorna regolarmente le immagini AMI ottimizzate per Amazon ECS con patch di sicurezza e la nuova versione di Container Agent. Quando utilizzi un ID AMI come immagine principale nella ricetta dell'immagine, devi controllare regolarmente gli aggiornamenti dell'immagine principale. Se sono disponibili aggiornamenti, è necessario creare una nuova versione della ricetta con l'AMI aggiornata. Ciò garantisce che le immagini personalizzate incorporino la versione più recente dell'immagine principale. Per informazioni su come creare un flusso di lavoro per aggiornare automaticamente le istanze EC2 nel cluster Amazon ECS con le AMI appena create, 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 EC2 Image Builder gestita. Amazon pubblica regolarmente immagini AMI ottimizzate per Amazon ECS tramite pipeline gestite. Queste immagini sono accessibili al pubblico. È necessario disporre delle autorizzazioni corrette per accedere all'immagine. Quando utilizzi un'immagine ARN anziché un'AMI nella tua ricetta Image Builder, la pipeline utilizza automaticamente la versione più recente dell'immagine principale ogni volta che viene eseguita. Questo approccio elimina la necessità di creare manualmente nuove versioni di ricette per ogni aggiornamento.

Utilizzo dell'immagine ARN con infrastruttura come codice (IaC)

Puoi configurare la ricetta utilizzando la console EC2 Image Builder o l'infrastruttura come codice (AWS CloudFormation ad esempio, AWS) o l'SDK. Quando specifichi un'immagine principale nella tua ricetta, puoi specificare un ID AMI EC2, un ARN Marketplace AWS dell'immagine Image Builder, un ID prodotto o un'immagine del contenitore. AWS pubblica pubblicamente sia gli ID AMI che gli ARN di immagini Image Builder di AMI ottimizzate per Amazon ECS. Di seguito è riportato il formato ARN per l'immagine:

```
arn:${Partition}:imagebuilder:${Region}:${Account}:image/${ImageName}/${ImageVersion}
```

Il formato `ImageVersion` è il seguente. Sostituisci *`major`*, *`minor`* e *`patch`* con i valori più recenti.

```
<major>.<minor>.<patch>
```

È possibile sostituire `major` `minor` e `patch` utilizzare valori specifici oppure utilizzare l'ARN senza versione di un'immagine, in modo che la pipeline rimanga up-to-date con la versione più recente dell'immagine principale. Un ARN senza versione utilizza il formato wildcard 'x.x.x' per rappresentare la versione dell'immagine. Questo approccio consente al servizio Image Builder di passare automaticamente alla versione più recente dell'immagine. L'utilizzo di un ARN senza versione garantisce che il riferimento punti sempre all'ultima immagine disponibile, semplificando il processo di mantenimento delle immagini aggiornate durante l'implementazione. Quando crei una ricetta con la console, EC2 Image Builder identifica automaticamente l'ARN per l'immagine principale. Quando si utilizza iAc per creare la ricetta, è necessario identificare l'ARN e selezionare la versione dell'immagine desiderata o utilizzare l'arn senza versione per indicare l'ultima immagine disponibile. Ti consigliamo di creare uno script automatico per filtrare e visualizzare solo le immagini in linea con i tuoi criteri. Il seguente script Python mostra come recuperare un elenco di AMI ottimizzate per Amazon ECS.

Lo script accetta due argomenti opzionali: `owner` e`platform`, con i valori predefiniti di «Amazon» e «Windows» rispettivamente. I valori validi per l'argomento proprietario sono: `Self`,`Shared`,`Amazon`, e`ThirdParty`. I valori validi per l'argomento platform sono `Windows` e`Linux`. Ad esempio, se esegui lo script con l'`owner`argomento impostato su `Amazon` e impostato su`Linux`, lo script genera un elenco di immagini pubblicate da Amazon, incluse immagini ottimizzate per Amazon ECS. `platform`

```
import boto3
import argparse

def list_images(owner, platform):
    # Create a Boto3 session
    session = boto3.Session()

    # Create an EC2 Image Builder client
    client = session.client('imagebuilder')

    # Define the initial request parameters
    request_params = {
        'owner': owner,
        'filters': [
            {
                'name': 'platform',
                'values': [platform]
            }
        ]
```

```
    }

    # Initialize the results list
    all_images = []

    # Get the initial response with the first page of results
    response = client.list_images(**request_params)

    # Extract images from the response
    all_images.extend(response['imageVersionList'])

    # While 'nextToken' is present, continue paginating
    while 'nextToken' in response and response['nextToken']:
        # Update the token for the next request
        request_params['nextToken'] = response['nextToken']

        # Get the next page of results
        response = client.list_images(**request_params)

        # Extract images from the response
        all_images.extend(response['imageVersionList'])

    return all_images

def main():
    # Initialize the parser
    parser = argparse.ArgumentParser(description="List AWS images based on owner and
 platform")

    # Add the parameters/arguments
    parser.add_argument("--owner", default="Amazon", help="The owner of the images.
 Default is 'Amazon'.")
    parser.add_argument("--platform", default="Windows", help="The platform type of the
 images. Default is 'Windows'.")

    # Parse the arguments
    args = parser.parse_args()

    # Retrieve all images based on the provided owner and platform
    images = list_images(args.owner, args.platform)

    # Print the details of the images
    for image in images:
```

```
        print(f"Name: {image['name']}, Version: {image['version']}, ARN:
 {image['arn']}")

if __name__ == "__main__":
    main()
```

Utilizzo dell'immagine ARN con AWS CloudFormation

Una ricetta di immagini Image Builder è un progetto che specifica l'immagine principale e i componenti necessari per ottenere la configurazione prevista dell'immagine di output. La risorsa viene utilizzata. `AWS::ImageBuilder::ImageRecipe` Imposta il `ParentImage` valore sull'ARN dell'immagine. Usa l'ARN senza versione dell'immagine desiderata per assicurarti che la pipeline utilizzi sempre la versione più recente dell'immagine. Ad esempio, `arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2023-ecs-optimized-x86/x.x.x`. La seguente definizione di `AWS::ImageBuilder::ImageRecipe` risorsa utilizza un ARN di immagine gestita da Amazon:

```
ECSRecipe:
    Type: AWS::ImageBuilder::ImageRecipe
    Properties:
      Name: MyRecipe
      Version: '1.0.0'
      Components:
        - ComponentArn: [<The component arns of the image recipe>]
      ParentImage: "arn:aws:imagebuilder:us-east-1:aws:image/amazon-linux-2023-ecs-
optimized-x86/x.x.x"
```

Per ulteriori informazioni sulla `AWS::ImageBuilder::ImageRecipe`risorsa, consulta la Guida per l'AWS CloudFormation utente.

Puoi automatizzare la creazione di nuove immagini nella tua pipeline impostando la `Schedule` proprietà della `AWS::ImageBuilder::ImagePipeline` risorsa. La pianificazione include una condizione di avvio e un'espressione cron. Per ulteriori informazioni, consulta `AWS::ImageBuilder::ImagePipeline` nella Guida per l'utente di AWS CloudFormation .

L'`AWS::ImageBuilder::ImagePipeline`esempio seguente fa in modo che la pipeline esegua una build ogni giorno alle 10:00 UTC (Coordinated Universal Time). Imposta i seguenti valori: `Schedule`

- Imposta `PipelineExecutionStartCondition` su `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE`. La build inizia solo se vengono aggiornate risorse dipendenti come l'immagine principale o i componenti, che utilizzano il carattere jolly 'x' nelle loro versioni semantiche. Ciò garantisce che la build incorpori gli ultimi aggiornamenti di tali risorse.
- Imposta ScheduleExpression sull'espressione cron. `(0 10 * * ? *)`

```
ECSPipeline:
    Type: AWS::ImageBuilder::ImagePipeline
    Properties:
      Name: my-pipeline
      ImageRecipeArn: <arn of the recipe you created in previous step>
      InfrastructureConfigurationArn: <ARN of the infrastructure configuration
 associated with this image pipeline>
      Schedule:
        PipelineExecutionStartCondition:
 EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE
        ScheduleExpression: 'cron(0 10 * * ? *)'
```

Utilizzo dell'immagine ARN con Terraform

L'approccio per specificare l'immagine e la pianificazione principali della pipeline in Terraform è in linea con quello in. AWS CloudFormation Usi la risorsa. `aws_imagebuilder_image_recipe` Imposta il `parent_image` valore sull'ARN dell'immagine. Usa l'ARN senza versione dell'immagine desiderata per assicurarti che la pipeline utilizzi sempre la versione più recente dell'immagine. Per ulteriori informazioni, consulta la documentazione di `aws_imagebuilder_image_recipe`Terraform.

Nel blocco di configurazione della schedulazione di`aws_imagebuilder_image_pipeline resource`, imposta il valore dell'`schedule_expression`argomento su un'espressione cron a tua scelta per specificare la frequenza di esecuzione della pipeline e imposta su. `pipeline_execution_start_condition` `EXPRESSION_MATCH_AND_DEPENDENCY_UPDATES_AVAILABLE` Per ulteriori informazioni, consulta la documentazione `aws_imagebuilder_image_pipeline`di Terraform.

Utilizzo di AWS Deep Learning Containers su Amazon ECS

AWS I Deep Learning Containers forniscono un set di immagini Docker per addestrare e servire modelli in TensorFlow Apache MXNet (Incubating) su Amazon ECS. I Deep Learning Containers consentono ambienti ottimizzati con TensorFlow le librerie NVIDIA CUDA (per istanze GPU) e Intel MKL (per istanze CPU). Le immagini dei contenitori per Deep Learning Container sono disponibili in Amazon ECR per fare riferimento alle definizioni di processo di Amazon ECS. Puoi utilizzare i Deep Learning Containers insieme a Amazon Elastic Inference per ridurre i costi di inferenza.

Per iniziare a usare i Deep Learning Containers senza Elastic Inference su Amazon ECS, consulta Deep Learning Containers su Amazon ECS nella Guida per gli sviluppatori di AWS Deep Learning AMI .

Deep Learning Containers con Elastic Inference su Amazon ECS

> **Note**
>
> A partire dal 15 aprile 2023, non AWS effettuerà l'onboarding di nuovi clienti in Amazon Elastic Inference (EI) e aiuterà i clienti attuali a migrare i propri carichi di lavoro verso opzioni che offrono prezzi e prestazioni migliori. Dopo il 15 aprile 2023, i nuovi clienti non saranno in grado di avviare istanze con acceleratori Amazon EI su Amazon, SageMaker Amazon ECS o Amazon EC2. Tuttavia, i clienti che hanno utilizzato Amazon EI almeno una volta negli ultimi 30 giorni sono considerati clienti attuali e potranno continuare a usufruire del servizio.

AWS I Deep Learning Containers forniscono un set di immagini Docker per servire modelli in TensorFlow Apache MXNet (Incubating) che sfruttano gli acceleratori Amazon Elastic Inference. Amazon ECS fornisce parametri di definizione dei processi per collegare acceleratori di Elastic Inference ai tuoi container. Quando specifichi un tipo di acceleratore Elastic Inference nella definizione di processo, Amazon ECS gestisce il ciclo di vita e la configurazione per l'acceleratore. Quando utilizzi questa funzionalità è necessario il ruolo collegato ai servizi Amazon ECS. Per ulteriori informazioni sugli acceleratori Elastic Inference, consulta Informazioni di base su Amazon Elastic Inference.

> **Important**
>
> Le istanze di container di Amazon ECS richiedono almeno la versione `1.30.0` dell'agente del 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 iniziare a usare i Deep Learning Containers con Elastic Inference su Amazon ECS, consulta Deep Learning Containers con Elastic Inference su Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Inference.

Service Quotas di Amazon ECS

La tabella seguente fornisce le quote di servizio di default, definite anche limiti, per Amazon ECS per un Account AWS. Per ulteriori informazioni sulle quote di servizio per altri Servizi AWS che puoi utilizzare con Amazon ECS, ad esempio Elastic Load Balancing e Dimensionamento automatico, consulta Quote di servizio di AWS 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.

Service Quotas di Amazon ECS

Di seguito sono riportate le quote di servizio di Amazon ECS.

AWS I nuovi account potrebbero avere quote iniziali inferiori che possono aumentare nel tempo. Amazon ECS monitora costantemente l'utilizzo dell'account all'interno di ciascuna regione, quindi aumenta automaticamente le quote in base all'utilizzo. Puoi inoltre richiedere un aumento della quota per i valori indicati come parametri regolabili, consulta Richiesta di aumento delle quote nella Guida per l'utente di Service Quotas.

Nome	Predefinita	Adattabile	Descrizione
Provider di capacità per cluster	Ogni regione supportata: 20	No	Il numero massimo di provider di capacità che è possibile associare a un cluster.
Classic Load Balancer per servizio	Ogni regione supportata: 1	No	Il numero massimo di Classic Load Balancer per servizio.
Cluster per account	Ogni regione supportata: 10.000	Sì	Numero di cluster per account
Istanze di container per cluster	Ogni regione supportata: 5.000	No	Numero di istanze di container per cluster

Nome	Predefinita	Adatta e	Descrizione
Numero di istanze di container per StartTask	Ogni regione supportata: 10	No	Il numero massimo di istanze del contenitore specificato in un'azione StartTask API.
Container per definizione di attività	Ogni regione supportata: 10	No	Il numero massimo di definizioni di container all'interno di una definizione di attività.
Sessioni ECS Exec	Ogni regione supportata: 1.000	No	Il numero massimo di sessioni ECS Exec per container.
Frequenza delle attività avviate da un servizio su AWS Fargate	Ogni regione supportata: 500	No	Il numero massimo di attività di cui è possibile eseguire il provisioning per servizio al minuto su Fargate dal pianificatore di servizio Amazon ECS.
Tasso di processi avviati da un servizio su un'istanza Amazon EC2 o esterna	Ogni regione supportata: 500	No	Il numero massimo di attività di cui è possibile eseguire il provisioning per servizio al minuto su un'istanza Amazon EC2 o esterna dal pianificatore di servizio Amazon ECS.

Nome	Predefinita	Adatta e	Descrizione
Revisioni per famiglia di definizioni di processo	Ogni regione supportata: 1.000.000	No	Il numero massimo di revisioni per famiglia di definizioni di processo. L'annullamento della registrazione o l'elimina zione di una revisione della definizione di attività non esclude che tale revisione venga inclusa in questo limite.
Gruppi di sicurezza per configurazione awsvpcConfiguration	Ogni Regione supportata: 5	No	Il numero massimo di gruppi di sicurezza specificati all'interno di una awsvpcConfiguration.
Servizi per cluster	Ogni regione supportata: 5.000	No	Numero massimo di servizi per cluster
Servizi per spazio dei nomi	Ogni regione supportata: 100	Sì	Il numero massimo di servizi che possono essere eseguiti in uno spazio dei nomi.
Sottoreti per awsvpcConfiguration	Ogni regione supportata: 16	No	Il numero massimo di sottoreti specifica te all'interno di una awsvpcConfiguration.
Tag per risorsa	Ogni regione supportata: 50	No	Il numero massimo di tag per risorsa. Questo vale per definizioni di processi, cluster, processi e servizi.

Nome	Predefinita	Adatta e	Descrizione
Gruppi di destinazione per servizio	Ogni Regione supportata: 5	No	Il numero massimo di gruppi di destinazione per servizio, se si utilizza un servizio di Applicati on Load Balancer o un Network Load Balancer.
Dimensione della definizione dell'attività	Ogni regione supportata: 64 KB	No	La dimensione massima, in KiB, di una definizione di attività.
Processi in stato di PROVISIONING per cluster	Ogni regione supportata: 500	No	Numero massimo di processi in attesa nello stato PROVISIONING per cluster. Questa quota si applica solo ai processi avviati mediante un provider di capacità del gruppo Auto Scaling di Amazon EC2.
Processi avviati per RunTask	Ogni regione supportata: 10	No	Il numero massimo di attività che possono essere avviate per azione API. RunTask
Attività per servizio	Ogni regione supportata: 5.000	No	Il numero massimo di attività per servizio (conteggio desiderato)

Note

I valori predefiniti sono le quote iniziali impostate da AWS, che sono separate dal valore di quota effettivamente applicato e dalla quota massima di servizio possibile. Per ulteriori

informazioni, consulta Terminologia di Service Quotas nella Guida per l'utente di Service Quotas.

> **Note**
>
> I servizi configurati per l'utilizzo dell'individuazione dei servizi Amazon ECS hanno un limite di 1.000 attività per servizio. Ciò si verifica a causa della quote di servizio AWS Cloud Map per il numero di istanze per servizio. Per ulteriori informazioni, consulta Service Quotas di AWS Cloud Map nella Riferimenti generali di Amazon Web Services.

> **Note**
>
> Nella pratica, le velocità di avvio dei processi 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 processi con un load balancer. Potresti riscontrare delle differenze nelle velocità di avvio delle attività rispetto alle quote rappresentate qui. Queste variazioni sono causate dalle funzionalità utilizzate per i servizi. Per ulteriori informazioni, consulta Best practice per i parametri del servizio Amazon ECS .

> **Note**
>
> I servizi configurati per l'utilizzo di Amazon ECS Service Connect hanno un limite di 1.000 attività per servizio. Ciò è dovuto alla quota AWS Cloud Map di servizio per il numero di istanze per servizio. Per ulteriori informazioni, consulta Service Quotas di AWS Cloud Map nella Riferimenti generali di Amazon Web Services.

AWS Fargate quote di servizio

Di seguito sono elencate le quote di AWS Fargate servizio di Amazon ECS e sono elencate sotto il AWS Fargateservizio nella console Service Quotas.

AWS I nuovi account potrebbero avere quote iniziali inferiori che possono aumentare nel tempo. Fargate monitora costantemente l'utilizzo dell'account all'interno di ciascuna regione, quindi aumenta

automaticamente le quote in base all'utilizzo. Puoi inoltre richiedere un aumento della quota per i valori indicati come parametri regolabili, consulta Richiesta di aumento delle quote nella Guida per l'utente di Service Quotas.

Nome	Predefinita	Adatta e	Descrizione
Conteggio delle risorse vCPU Fargate on demand	Ogni regione supportata: 6	Sì	Il numero di vCPU Fargate che vengono eseguite simultane amente come Fargate On-Demand in questo account nella regione corrente.
Conteggio risorse vCPU Fargate Spot	Ogni regione supportata: 6	Sì	Il numero di vCPU Fargate che vengono eseguite simultane amente come Fargate Spot in questo account nella regione corrente.

Note

I valori predefiniti sono le quote iniziali impostate da AWS, che sono separate dal valore effettivo della quota applicata e dalla quota massima di servizio possibile. Per ulteriori informazioni, consulta Terminologia di Service Quotas nella Guida per l'utente di Service Quotas.

Note

Fargate applica inoltre le attività Amazon ECS e i limiti di velocità di avvio dei pod Amazon EKS. Per ulteriori informazioni, consulta la sezione Limiti di velocità di Fargate.

Gestione delle quote di Amazon ECS e dei AWS Fargate servizi in AWS Management Console

Amazon ECS si è integrato con Service Quotas, AWS un servizio che consente di visualizzare e gestire le quote da una posizione centrale. Per ulteriori informazioni, consulta Cos'è Service Quotas? nella Guida per l'utente di Service Quotas.

Service Quotas semplifica la ricerca del valore di tutte le quote di servizio di Amazon ECS.

AWS Management Console

Come visualizzare le quote di servizio di Amazon ECS e Fargate tramite la AWS Management Console

1. Apri la console Service Quotas all'indirizzo https://console.aws.amazon.com/servicequotas/
2. Nel pannello di navigazione, scegli 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, AWS Management Console consulta la Service Quotas User Guide. Per richiedere un aumento delle quote, consultare Richiesta di aumento delle quote nella Guida dell'utente di Service Quotas.

AWS CLI

Come visualizzare le quote di servizio di Amazon ECS e Fargate tramite la AWS CLI

Esegui questo comando per visualizzare le quote di default di Amazon ECS.

```
aws service-quotas list-aws-default-service-quotas \
```

```
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code ecs \
    --output table
```

Esegui questo comando per visualizzare le quote di default di Fargate.

```
aws service-quotas list-aws-default-service-quotas \
    --query 'Quotas[*].
{Adjustable:Adjustable,Name:QuotaName,Value:Value,Code:QuotaCode}' \
    --service-code fargate \
    --output table
```

Esegui questo comando per visualizzare le quote di Fargate applicate.

```
aws service-quotas list-service-quotas \
    --service-code fargate
```

> **Note**
>
> Amazon ECS non supporta le quote applicate.

Per ulteriori informazioni sull'utilizzo delle quote di servizio utilizzando il AWS CLI, vedere Service Quotas Command AWS CLI Reference. Per richiedere un aumento delle quote, consultare il comando `request-service-quota-increase` nella Documentazione di riferimento sui comandi AWS CLI.

Gestisci le quote di servizio Amazon ECS e i limiti di limitazione delle API

Amazon ECS è integrato con diversi sistemi Servizi AWS, tra cui Elastic Load Balancing AWS Cloud Map e Amazon EC2. Grazie a questa stretta integrazione, Amazon ECS include diverse funzionalità come il bilanciamento del carico di servizio, Service Connect, il task networking e la scalabilità automatica dei cluster. Amazon ECS e l'altro Servizi AWS che integra con tutti mantengono le quote di servizio e i limiti di velocità delle API per garantire prestazioni e utilizzo costanti. Queste quote di servizio impediscono inoltre l'approvvigionamento accidentale di più risorse del necessario e proteggono da azioni dannose che potrebbero far aumentare la bolletta.

Acquisendo dimestichezza con le quote di servizio e i limiti di velocità delle AWS API, puoi pianificare la scalabilità dei carichi di lavoro senza preoccuparti di un peggioramento imprevisto delle prestazioni. Per ulteriori informazioni, consulta Request throttling for the Amazon ECS API.

Quando si ridimensionano i carichi di lavoro su Amazon ECS, si consiglia di prendere in considerazione la seguente quota di servizio.

- AWS Fargate prevede quote che limitano il numero di attività in esecuzione simultanee per ciascuna di esse. Regione AWS Esistono quote per le attività On-Demand e Fargate Spot su Amazon ECS. Ogni quota di servizio include anche tutti i pod Amazon EKS che esegui su Fargate.
- Per le attività eseguite su istanze Amazon EC2, il numero massimo di istanze Amazon EC2 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 Amazon EC2 per il tuo cluster da solo, questa quota potrebbe diventare un collo di bottiglia nell'implementazione. Se hai bisogno di maggiore capacità, puoi creare più cluster o richiedere un aumento della quota di servizio.
- Se utilizzi l'autoscaling del cluster Amazon ECS con un fornitore di capacità di gruppo Auto Scaling, quando ridimensioni i tuoi servizi considera la quota. `Tasks in the PROVISIONING state per cluster` Questa quota è il numero massimo di attività nello `PROVISIONING` stato per ogni cluster per le quali i fornitori di capacità possono aumentare la capacità. Quando avvii un gran numero di attività contemporaneamente, puoi facilmente raggiungere questa quota. Un esempio è l'implementazione simultanea di decine di servizi, ciascuno con centinaia di attività. Quando ciò accade, il provider di capacità deve avviare nuove istanze di container per effettuare le attività quando la capacità del cluster è insufficiente. Mentre il provider di capacità sta lanciando istanze Amazon EC2 aggiuntive, è probabile che il service scheduler di Amazon ECS continuerà ad avviare attività in parallelo. Tuttavia, questa attività potrebbe essere limitata a causa dell'insufficiente capacità del cluster. Lo strumento di pianificazione dei servizi Amazon ECS implementa una strategia di limitazione esponenziale e di back-off per riprovare a posizionare le attività man mano che vengono lanciate nuove istanze di container. Di conseguenza, potrebbero verificarsi tempi di implementazione o scalabilità più lenti. Per evitare questa situazione, è possibile pianificare le implementazioni dei servizi in uno dei seguenti modi. Distribuire un gran numero di attività non richiede un aumento della capacità del cluster oppure mantenere la capacità del cluster di riserva per il lancio di nuove attività.

Oltre a considerare la quota di servizio di Amazon ECS durante la scalabilità dei carichi di lavoro, considera anche la quota di servizio per gli altri Servizi AWS che sono integrati con Amazon ECS.

Sistema di bilanciamento del carico elastico

Puoi configurare i tuoi servizi Amazon ECS per utilizzare Elastic Load Balancing per distribuire il traffico in modo uniforme tra le attività. Per ulteriori informazioni e best practice consigliate su come scegliere un sistema di bilanciamento del carico, consulta. Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS

Quote del servizio Elastic Load Balancing

Quando scalate i carichi di lavoro, prendete in considerazione le seguenti quote di servizio Elastic Load Balancing. La maggior parte delle quote del servizio Elastic Load Balancing è regolabile ed è possibile richiedere un aumento nella console Service Quotas.

Application Load Balancer

Quando si utilizza un Application Load Balancer, a seconda del caso d'uso, potrebbe essere necessario richiedere un aumento della quota per:

- La `Targets per Application Load Balancer` quota, ossia il numero di obiettivi alla base dell'Application Load Balancer.
- La `Targets per Target Group per Region` quota, che è il numero di obiettivi alla base dei gruppi target.

Per ulteriori informazioni, consulta Quotas for your Application Load Balancers nella User Guide for Application Load Balancers.

Network Load Balancer

Esistono limitazioni più rigorose sul numero di target che è possibile registrare con un Network Load Balancer. Quando si utilizza un Network Load Balancer, spesso si desidera abilitare il supporto tra zone, che comporta limitazioni di scalabilità aggiuntive sul `Targets per Availability Zone Per Network Load Balancer` numero massimo di destinazioni per zona di disponibilità per ogni Network Load Balancer. Per ulteriori informazioni, consulta Quotas for your Network Load Balancer nella Guida per l'utente di Network Load Balancers.

Limitazione dell'API Elastic Load Balancing

Quando configuri un servizio Amazon ECS per utilizzare un sistema di bilanciamento del carico, i controlli di integrità del gruppo target devono essere superati prima che il servizio venga considerato

integro. Per eseguire questi controlli di integrità, Amazon ECS richiama le operazioni dell'API Elastic Load Balancing per tuo conto. Se hai un gran numero di servizi configurati con sistemi di bilanciamento del carico nel tuo account, potresti rallentare le implementazioni dei servizi a causa di potenziali limitazioni specifiche per le operazioni dell'API Elastic Load Balancing e di `RegisterTarget` Elastic `DeregisterTarget` Load Balancing. `DescribeTargetHealth` Quando si verifica una limitazione, si verificano errori di limitazione nei messaggi relativi agli eventi del servizio Amazon ECS.

Se riscontri una limitazione AWS Cloud Map delle API, puoi contattarci AWS Support per ricevere indicazioni su come aumentare i limiti di limitazione delle API. AWS Cloud Map Per ulteriori informazioni sul monitoraggio e la risoluzione di tali errori di limitazione, consulta. Gestisci i problemi di limitazione di Amazon ECS

Interfacce di rete elastiche

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

Quote di servizio di interfaccia di rete elastiche

Quando si eseguono attività che utilizzano la modalità di `awsvpc` rete, a ciascuna attività viene collegata un'interfaccia elastica di rete unica. Se tali attività devono essere raggiunte tramite Internet, assegna un indirizzo IP pubblico all'elastic network interface per tali attività. Quando ridimensioni i tuoi carichi di lavoro Amazon ECS, prendi in considerazione queste due quote importanti:

- La `Network interfaces per Region` quota, ossia il numero massimo di interfacce di rete disponibili per il tuo account Regione AWS .
- La `Elastic IP addresses per Region` quota che è il numero massimo di indirizzi IP elastici in un Regione AWS.

Entrambe queste quote di servizio sono regolabili e per queste puoi richiedere un aumento dalla tua console Service Quotas. Per ulteriori informazioni, consulta le quote dei servizi Amazon VPC nella Guida per l'utente di Amazon Virtual Private Cloud.

Per i carichi di lavoro Amazon ECS ospitati su istanze Amazon EC2, quando esegui attività che utilizzano `awsvpc` la modalità di rete, considera la quota di servizio, `Maximum network`

`interfaces` il numero massimo di istanze di rete per ogni istanza Amazon EC2. Questa quota limita il numero di attività che puoi eseguire su un'istanza. Non è possibile modificare la quota e non è disponibile nella console Service Quotas. Per ulteriori informazioni, consulta Indirizzi IP per interfaccia di rete per tipo di istanza nella Guida per l'utente di Amazon EC2.

Sebbene non sia possibile modificare il numero di interfacce di rete che possono essere collegate a un'istanza Amazon EC2, puoi utilizzare la funzionalità elastic network interface trunking per aumentare il numero di interfacce di rete disponibili. Ad esempio, per impostazione predefinita, un'`c5.large`istanza può avere fino a tre interfacce di rete. L'interfaccia di rete principale per l'istanza viene calcolata come unica. Pertanto, è possibile collegare altre due interfacce di rete all'istanza. Poiché ogni attività che utilizza la modalità di `awsvpc` rete richiede un'interfaccia di rete, in genere è possibile eseguire solo due di queste attività su questo tipo di istanza. Ciò può portare a un sottoutilizzo della capacità del cluster. Se abiliti l'elastic network interface trunking, puoi aumentare la densità dell'interfaccia di rete per inserire un numero maggiore di attività su ciascuna istanza. Con il trunking attivato, un'`c5.large`istanza può avere fino a 12 interfacce di rete. L'istanza ha l'interfaccia di rete principale e Amazon ECS crea e collega un'interfaccia di rete «trunk» all'istanza. Di conseguenza, con questa configurazione puoi eseguire 10 attività sull'istanza anziché le due attività predefinite. Per ulteriori informazioni, consulta Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.

Limitazione delle API dell'interfaccia di rete elastica

Quando esegui attività che utilizzano la modalità di `awsvpc` rete, Amazon ECS si affida alle seguenti API di Amazon EC2. Ciascuna di queste API ha diverse accelerazioni di API. Per ulteriori informazioni, consulta Request throttling for the Amazon EC2 API Reference in Amazon EC2 API Reference.

- CreateNetworkInterfaccia
- AttachNetworkInterfaccia
- DetachNetworkInterfaccia
- DeleteNetworkInterfaccia
- DescribeNetworkInterfacce
- DescribeVpcs
- DescribeSubnets
- DescribeSecurityGruppi
- DescribeInstances

Se le chiamate API di Amazon EC2 vengono limitate durante i flussi di lavoro di provisioning dell'interfaccia di rete elastica, lo scheduler dei servizi di Amazon ECS riprova automaticamente con back-off esponenziali. Questi ritiri automatici possono talvolta causare un ritardo nell'avvio delle attività, con conseguente rallentamento della velocità di implementazione. Quando si verifica la limitazione delle API, viene visualizzato il messaggio `Operations are being throttled. Will try again later.` nei messaggi relativi agli eventi di servizio. Se soddisfi costantemente i limiti delle API di Amazon EC2, puoi contattarci AWS Support per ricevere indicazioni su come aumentare i limiti di limitazione delle API. Per ulteriori informazioni sul monitoraggio e la risoluzione degli errori di limitazione, consulta Gestione dei problemi di limitazione.

AWS Cloud Map

Amazon ECS service discovery e Service Connect utilizzano le AWS Cloud Map API per gestire i namespace per i tuoi servizi Amazon ECS. Se i tuoi servizi prevedono un gran numero di attività, prendi in considerazione i seguenti consigli.

AWS Cloud Map quote di servizio

Quando i servizi Amazon ECS sono configurati per utilizzare service discovery o Service Connect, la `Tasks per service` quota, che è il numero massimo di attività per il servizio, è influenzata dalla quota di AWS Cloud Map `Instances per service` servizio, che è il numero massimo di istanze per quel servizio. In particolare, la quota AWS Cloud Map di servizio riduce la quantità di attività che è possibile eseguire fino a un massimo di 1.000 attività per il servizio. Non è possibile modificare la AWS Cloud Map quota. Per ulteriori informazioni, consulta Service Quotas di AWS Cloud Map.

AWS Cloud Map Limitazione delle API

Amazon ECS chiama le `DeregisterInstance` AWS Cloud Map API `ListInstancesGetInstancesHealthStatus`,`RegisterInstance`, e per tuo conto. Aiutano a scoprire i servizi ed eseguono controlli sullo stato di salute quando avvii un'attività. Quando vengono distribuiti contemporaneamente più servizi che utilizzano il service discovery con un gran numero di attività, ciò può comportare il superamento dei limiti di limitazione delle AWS Cloud Map API. Quando ciò accade, è probabile che venga visualizzato il seguente messaggio: `Operations are being throttled. Will try again later` nei messaggi relativi agli eventi del servizio Amazon ECS e una velocità di implementazione e avvio delle attività più lente. AWS Cloud Map non documenta i limiti di limitazione per queste API. Se riscontri una limitazione dovuta a queste, puoi contattarci AWS Support per ricevere assistenza su come aumentare i limiti di limitazione delle API. Per ulteriori consigli sul monitoraggio e la risoluzione di tali errori di limitazione, consulta. Gestisci i problemi di limitazione di Amazon ECS

Documentazione di riferimento dell'API Amazon ECS

Oltre a AWS Management Console and the AWS Command Line Interface (AWS CLI), Amazon ECS fornisce anche un'API. Puoi usare l'API per automatizzare le attività per la gestione delle risorse Amazon ECS.

- Per un elenco delle operazioni API in base alla risorsa Amazon ECS, consulta Azioni della risorsa Amazon ECS.
- Per un elenco alfabetico delle operazioni API, consulta Operazioni.
- Per un elenco alfabetico dei tipi di dati, consulta la pagina Tipi di dati.
- Per un elenco di parametri di query comuni, consulta la pagina Parametri Comuni.
- Per le descrizioni dei codici di errore, consulta la pagina Errori comuni.

Per ulteriori informazioni su AWS CLI, consulta il AWS Command Line Interface riferimento per Amazon ECS.

Cronologia dei documenti

La tabella riportata di seguito illustra i principali aggiornamenti e le nuove funzioni per la Guida per gli sviluppatori di Amazon Elastic Container Service. Inoltre, aggiorniamo frequentemente la documentazione per tener conto del feedback inviatoci.

Modifica	Descrizione	Data
Supporto gMSA per contenitori Linux su Fargate	Amazon ECS supporta l'autenticazione Active Directory per contenitori Linux su Fargate tramite un tipo speciale di account di servizio chiamato Managed Service Account (gMSA) di gruppo. Per ulteriori informazioni, consulta Utilizzo gMSA per i Linux contenitori su Fargate.	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 parametri di Amazon ECS. CloudWatch	8 febbraio 2024
Service Connect TLS	Ora puoi usare TLS con Service Connect.	22 gennaio 2024
Politica gestita da Service Connect TLS	Aggiunta una nuova politica AmazonECS. InfrastructureRolePolicyForServiceConnectTransportLayerSecurity	22 gennaio 2024
Aggiornamento della configurazione del timeout di Service Connect	La configurazione del timeout di Service Connect può ora essere aggiornata e include due parametri opzionali: `idleTimeout` `eperRequestTimeout` .	22 gennaio 2024
Drenaggio delle istanze gestite da Amazon ECS	Puoi utilizzare il drenaggio gestito delle istanze di Amazon ECS per facilitare la chiusura ordinata delle istanze Amazon ECS.	19 gennaio 2024

Modifica	Descrizione	Data
È stato aggiunto il supporto per Ubuntu 2.2 per ECS Anywhere	Il supporto per il sistema operativo Ubuntu 22 è stato aggiunto a ECS Anywhere. Per ulteriori informazioni, consulta Sistemi operativi e architetture di sistema supportati.	16 gennaio 2024
Aggiungi la politica `AmazonECSInfrastructureRolePolicyForVolumes` IAM	È stato aggiunto `AmazonECSInfrastructureolePolicyForVolumes`. 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 Amazon ECS	Puoi configurare 1 volume di dati Amazon EBS per attività durante la distribuzione per il collegamento a attività Amazon ECS autonome o attività gestite da un servizio ECS. La configurazione di un volume al momento della distribuzione consente di creare definizioni di attività riutilizzabili non limitate a tipi o impostazioni di volume specifici. I volumi Amazon EBS forniscono uno storage a blocchi ad alta disponibilità, conveniente, durevole e ad alte prestazioni per carichi di lavoro containerizzati a uso intensivo di dati.	11 gennaio 2024
La console classica Amazon ECS ha raggiunto la fine del ciclo di vita	La console Amazon ECS ha raggiunto la fine del ciclo di vita.	4 dicembre 2023
Policy aggiornata	La policy IAM gestita da ServiceRolePolicy AmazonECS è stata aggiornata con `events` nuove autorizzazioni e autorizzazioni e aggiuntive. `autoscaling autoscaling-plans`	4 dicembre 2023

Modifica	Descrizione	Data
Supporto per il monitoraggio del runtime	Puoi utilizzare Runtime Monitoring per monitorare i carichi di lavoro Amazon ECS e identificare comportamenti dannosi o non autorizzati. Per ulteriori informazioni, consulta Runtime Monitoring.	26 novembre 2023
Policy aggiornata	La policy IAM `AmazonECSServiceRolePolicy` gestita è stata aggiornata per consentire l'accesso all' AWS Cloud Map `DiscoverInstancesRevision` API.	4 ottobre 2023
AWS Fargate configurazione del ritiro delle attività	Puoi configurare il periodo di attesa prima che le attività di Fargate vengano ritirate. Per ulteriori informazioni, Consulta Manutenzione delle attività di AWS Fargate.	5 settembre 2023
Parametri aggiuntivi di definizione delle attività in AWS Fargate	AWS Fargate aggiunge il supporto per `pidMode` e `systemControls` nella versione della piattaforma Linux`1.4.0`. Per ulteriori informazioni, consulta Definizioni delle attività.	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 Creazione di una definizione delle attività utilizzando la console.	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 la sezione Lazy loading delle immagini dei container utilizzando Seekable OCI (SOCI) nella Amazon ECS User Guide for Fargate. AWS	17 luglio 2023

Modifica	Descrizione	Data
Supporto migliorato per gMSA su Linux e Windows	La definizione delle attività ha un nuovo campo `credentialSpecs` per gMSA per Linux e Windows. È stato aggiunto un nuovo tutorial completo per gMSA senza dominio su Windows, consulta Tutorial: utilizzo di container Windows con gMSA senza dominio utilizzando l' AWS CLI. Per ulteriori informazioni, consulta Utilizzo di gMSA per i container Linux e Utilizzo di gMSA per i container Windows.	14 luglio 2023
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 Versioni dell'agente del container Amazon ECS Linux su GitHub.	20 giugno 2023
Disponibilità regionale aggiornata per il supporto Fargate ARM64	La disponibilità regionale per il supporto Fargate ARM64 è stata aggiornata. Per ulteriori informazioni, consulta Considerazioni.	19 giugno 2023
Miglioramento della documentazione relativa al dimensionamento automatico del cluster	La documentazione per il dimensionamento di Amazon ECS di Dimensionamento automatico Amazon EC2 presenta miglioramenti significativi in termini di precisione e leggibilità. Per maggiori informazioni, consulta Dimensionamento automatico del cluster Amazon ECS.	4 maggio 2023

Modifica	Descrizione	Data
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 si crea una risorsa e si specificano i tag per quella risorsa, AWS esegue un'autorizzazione aggiuntiva per verificare che vi siano le autorizzazioni per creare tag. Per ulteriori informazioni, consulta Autorizzazione all'assegnazione di tag e Concessione dell'autorizzazione per assegnare tag alle risorse al momento della creazione.	18 aprile 2023
Supporto per gMSA per container Linux su EC2	Puoi utilizzare gMSA per autenticarti nei container Active Directory per Linux su EC2. Per ulteriori informazioni, consulta Utilizzo di gMSA per container Linux.	14 aprile 2023
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 Archiviazione delle attività di Fargate.	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 Fargate ed EC2. Per ulteriori informazioni, consulta Report di costi e utilizzo a livello di attività.	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 AMI Linux ottimizzate per Amazon ECS.	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 AWS Fargate Standard federale per l'elaborazione delle informazioni (FIPS-140).	10 aprile 2023

Modifica	Descrizione	Data
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 Eliminazione di una revisione della definizione delle attività utilizzando la console e Definizioni delle attività.	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 Visualizzazione dei suggerimenti per i servizi Amazon ECS su Fargate.	27 gennaio 2023
Console Amazon ECS	La nuova console Amazon ECS è ora la console predefinita. Per ulteriori informazioni, consulta Nuova console Amazon ECS.	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 AmazonECS_ FullAccess.	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 the section called “Rilevamento degli errori ”.	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 the section called “Mappature di porte”.	15 dicembre 2022

Modifica	Descrizione	Data
Disponibilità generale di Amazon ECS Service Connect	Questa funzionalità aggiunge la funzione di individua zione dei servizi e la mesh di servizi controllati dalle implementazioni dei servizi Amazon ECS. Per ulteriori informazioni, consulta the section called “ Service Connect”.	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 the section called “Creazione di una definizione di attività attraverso la nuova console”.	27 ottobre 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 the section called “Creazione di una definizione di attività attraverso la nuova console”.	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, consulta the section called “Creazione di un servizio” e the section called “Esecuzione di un'applicazione come attività”.	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 the section called “Endpoint metadati delle attività versione 4”.	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 the section called “Dimensioni processo”.	16 settembre 2022

Modifica	Descrizione	Data
Pagine della CLI ECS archiviate	La documentazione della CLI ECS è stata archiviat a. Ti consigliamo di utilizzare AWS Copilot per le tue esigenze di strumenti da riga di comando. Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.	15 settembre 2022
Nuove quote Fargate	Fargate sta passando da quote basate sul numero di attività a quote basate su vCPU. Per ulteriori informazi oni, consulta the section called “AWS Fargate quote di servizio”.	8 settembre 2022
Supporto per warm pool per Amazon EC2 Auto Scaling.	Ora puoi utilizzare i warm pool di Amazon EC2 Auto Scaling per dimensionare orizzontalmente in modo più rapido le applicazioni e risparmiare sui costi. Per ulteriori informazioni, consulta Configurazione di istanze preinizializzate per il tuo gruppo Amazon ECS Auto Scaling.	23 marzo 2022
Supporto per le istanze Windows in ECS Anywhere.	ECS Anywhere ora supporta le istanze Windows. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio esterno.	3 marzo 2022
Aggiunto il supporto ECS Exec per istanze esterne	ECS Exec ora è supportato per le istanze esterne. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS con ECS Exec.	24 gennaio 2022

Modifica	Descrizione	Data
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'annulla mento della registrazione di una definizione di attività. Per ulteriori informazioni, consulta Creazione di un cluster Amazon ECS per il tipo di lancio Fargate, Eliminazione di un cluster Amazon ECS, Aggiornam ento di una definizione di attività Amazon ECS tramite la console e Annullamento della registrazione di una revisione della definizione di attività di Amazon ECS tramite la console.	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 Creazione di una definizione di attività Amazon ECS utilizzando la console.	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 the section called “Definizioni delle attività per carichi di lavoro ARM a 64 bit”.	23 novembre 2021
Supporto Amazon ECS per l'opzione fluentd log-driver-buffer-limit	Amazon ECS supporta l'opzione `log-driver-buffer-limit` fluentd. Per ulteriori informazioni, consulta Inviare i log di Amazon ECS a un servizio o AWSAWS Partner.	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 Script di build per AMI Linux ottimizzata per Amazon ECS.	19 novembre 2021

Modifica	Descrizione	Data
Integrità delle istanze di container	Amazon ECS aggiunge il supporto per il monitoraggio dell'integrità delle istanze di container. Per ulteriori informazioni, consulta Monitora lo stato delle istanze dei container Amazon ECS.	10 novembre 2021
Supporto per Windows Amazon ECS Exec	Amazon ECS Exec supporta Windows. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS con ECS Exec.	1° novembre 2021
Supporto per container Windows su Fargate.	Amazon ECS supporta container Windows su Fargate. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Windows per Amazon ECS.	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, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU e Registrazione di un'istanza esterna in un cluster Amazon ECS.	8 ottobre 2021
Supporto della modalità di rete `awsvpc` su Windows	Amazon ECS supporta la modalità di rete `awsvpc` su Windows. Per ulteriori informazioni, consulta Assegna un'interfaccia di rete per un'attività Amazon ECS.	15 luglio 2021
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 AMI Bottlerocket ottimizzate per Amazon ECS.	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

Modifica	Descrizione	Data
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 AWS politiche gestite per Amazon Elastic Container Service.	8 giugno 2021
Iniziare con AWS CDK	È stata aggiunta una guida introduttiva per l'utilizzo AWS CDK con Amazon ECS. Per ulteriori informazioni, consulta Creazione di risorse Amazon ECS utilizzando AWS CDK.	27 maggio 2021
Amazon ECS Anywhere	Amazon ECS ha aggiunto il supporto per la registrazione di un server on-premise o di una macchina virtuale (VM) con il cluster. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio esterno.	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 AMI Linux ottimizzate per Amazon ECS.	19 aprile 2021
Amazon ECS Exec	Amazon ECS ha rilasciato un nuovo strumento di debug chiamato ECS Exec. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS con ECS Exec.	15 marzo 2021
Supporto per la policy degli endpoint VPC	Amazon ECS ora supporta le policy dell'endpoint VPC. Per ulteriori informazioni, consulta Creazione di una policy per l'endpoint VPC per Amazon ECS.	11 gennaio 2021

Modifica	Descrizione	Data
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, consulta Creazione di un servizio Amazon ECS utilizzando la console e Esecuzione di un'applicazione come attività Amazon ECS.	28 dicembre 2020
Aggiornamento del provider di capacità	Amazon ECS ha aggiunto il supporto per l'aggiorn amento di un provider di capacità del gruppo Auto Scaling esistente.	23 novembre 2020
ECS ora supporta Amazon FSx for Windows File Server per processi Windows	Amazon ECS ha aggiunto il supporto per specifica re i volumi di Amazon FSx for Windows File Server nelle definizioni di processi di Windows. Per ulteriori informazioni, consulta Usa FSx per volumi Windows File Server con Amazon ECS.	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 processi che utilizzan o la modalità di rete `awsvpc`, che offre il supporto per gli indirizzi IPv6. Per ulteriori informazioni, consulta Utilizzo di un VPC in modalità dual-stack.	5 novembre 2020
Aggiornamento endpoint metadati dei processi v4	Amazon ECS ha aggiunto ulteriori metadati all'output dell'endpoint dei metadati del processo v4. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 4.	5 novembre 2020
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 Applicazioni Amazon ECS in sottoreti condivise, Local Zones e Wavelength Zones.	4 settembre 2020

Modifica	Descrizione	Data
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 AMI Bottlerocket ottimizzate per Amazon ECS.	31 agosto 2020
Endpoint metadati dei processi versione 4 aggiornato per le statistiche di velocità di rete	L'endpoint dei metadati dei processi versione 4 è stato aggiornato per fornire statistiche sulla velocità di rete per processi di Amazon ECS che utilizzano le modalità di rete `awsvpc` o `bridge` ospitate su istanze Amazon EC2 che eseguono almeno la versione `1.43.0` dell'agente del container. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 4.	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 Parametri di utilizzo.	3 agosto 2020
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 Creazione di risorse Amazon ECS utilizzando l'interfaccia a riga di AWS comando Copilot.	9 luglio 2020

Modifica	Descrizione	Data
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 AWS Deprecazione della versione della piattaforma Fargate Linux.	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. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	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 Archiviazione effimera delle attività Fargate per Amazon ECS.	28 maggio 2020
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 Passa una singola variabile di ambiente a un contenitore Amazon ECS.	18 maggio 2020

Modifica	Descrizione	Data
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	È stata aggiornata la seguente quota di servizio: • I numero di cluster per account è stato aumentato da `2,000` a `10,000`. Per ulteriori informazioni, consulta Service Quotas di Amazon ECS.	17 aprile 2020

Modifica	Descrizione	Data
AWS Piattaforma Fargate versione 1.4.0	AWS Viene rilasciata la versione 1.4.0 della piattaforma Fargate, che contiene le seguenti funzionalità: • Aggiunto il supporto per l'utilizzo dei volumi del file system Amazon EFS per l'archiviazione dei processi persistenti. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS. • L'archiviazione effimera delle attività è stata aumentata a 20 GB. Per ulteriori informazioni, consulta Archiviazione effimera delle attività Fargate per Amazon ECS. • Il comportamento del traffico di rete da e verso le attività è stato aggiornato. A partire dalla versione 1.4 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 Reti di processi Fargate nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate. • I task ENI aggiungono il supporto per i frame jumbo. Le interfacce di rete sono configurate con un'unità di trasmissione massima (MTU), ovvero la dimension 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	8 aprile 2020

Modifica	Descrizione	Data
	supporta frame jumbo, ad esempio tutto il traffico che rimane all'interno del VPC. • CloudWatch Container Insights includerà metriche delle prestazioni di rete per le attività di Fargate. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights. • 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 Endpoint di metadati delle attività Amazon ECS versione 4. • 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. Questa modifica non dovrebbe avere effetto sull'esecuzione delle attività. • 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. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.	

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 Usa i volumi Amazon EFS con Amazon ECS.	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 statistiche Docker. Per ulteriori informazioni, consulta Endpoint di metadati delle attività Amazon ECS versione 4.	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 Trasferisci dati sensibili a un contenitore Amazon ECS.	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 Convalida lo stato di un servizio Amazon ECS prima della distribuzione .	6 febbraio 2020
È stato aggiunto il parametro di definizione dell'attività efsVolume Configuration	Il parametro di definizione dei processi `efsVolumeConfiguration` è in anteprima pubblica, il che semplifica l'utilizzo dei file system Amazon EFS con i processi Amazon ECS. Per ulteriori informazioni, consulta Usa i volumi Amazon EFS con Amazon ECS.	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 Parametri di configurazione del registro dell'agente container Amazon ECS.	13 gennaio 2020
Fargate Spot	Amazon ECS ha aggiunto il supporto per l'esecuzione di processi che utilizzano Fargate Spot. Per ulteriori informazioni, consulta Cluster Amazon ECS per il tipo di lancio Fargate .	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 Gestisci automaticamente la capacità di Amazon ECS con la scalabilità automatica del cluster.	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 Cluster Amazon ECS.	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 the section called “Amazon Elastic Container Service su AWS Outposts”.	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 Eventi di azione del servizio Amazon ECS.	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 Definizioni di attività Amazon ECS per carichi di lavoro GPU.	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 Inviare i log di Amazon ECS a un servizio o AWSAWS Partner.	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. Per ulteriori informazioni, consulta Deep Learning Containers con Elastic Inference su Amazon ECS.	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 Inviare i log di Amazon ECS a un servizio o AWSAWS Partner.	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 Monitora i contenitori Amazon ECS utilizzando Container Insights.	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 può essere limitata. Per ulteriori informazioni, consulta Gestione dello spazio di memoria di swap dei container su Amazon ECS.	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	Aggiunti altri tipi di istanze Amazon EC2 supportati per la funzionalità di trunking dell'ENI. Per ulteriori informazioni, consulta Istanze supportate per interfacce di rete di container Amazon ECS potenziate.	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 Registrazione di più gruppi target con un servizio Amazon ECS.	30 luglio 2019
Specifica di dati sensibili utilizzando segreti di Secrets Manager	Aggiunto un tutorial per la specifica di dati sensibili tramite segreti di Secrets Manager. Per ulteriori informazioni, consulta Specificazione di dati sensibili utilizzando i segreti di Secrets Manager in Amazon ECS.	20 luglio 2019

Modifica	Descrizione	Data
CloudWatch Container Insights	Amazon ECS ha aggiunto il supporto per CloudWatch Container Insights. Per ulteriori informazioni, consulta Monitora i contenitori Amazon ECS utilizzando Container Insights.	9 luglio 2019
Autorizzazioni a livello di risorsa per servizi e set di processi Amazon ECS	Amazon ECS ha esteso il supporto per le autorizza zioni a livello di risorsa per i servizi e i processi di Amazon ECS. Per ulteriori informazioni, consulta Come funziona Amazon Elastic Container Service con IAM.	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 AWS-2019-005.	17 giugno 2019
Trunking dell'interfaccia di rete elastica	Amazon ECS ha introdotto il supporto per l'avvio di istanze di container tramite i tipi di istanze Amazon EC2 supportati che dispongono di densità di interfacc ia di rete elastica (ENI) aumentata. L'uso di questi tipi di istanze e il consenso esplicito all'impostazione dell'account `awsvpcTrunking` 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 Aumento delle interfacce di rete di istanze di container Amazon ECS Linux.	6 giugno 2019
AWS Aggiornamento della piattaforma Fargate versione 1.3.0	A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta i driver di log `splunk` in aggiunta ai driver di log `awslogs`. Per ulteriori informazioni, consulta Archiviazione e registraz ione.	1 maggio 2019

Modifica	Descrizione	Data
AWS Aggiornamento della piattaforma Fargate versione 1.3.0	A partire dal 1 maggio 2019, ogni nuovo processo Fargate che viene avviato supporta il riferimento ai dati sensibili nella configurazione dei log di un container utilizzando il parametro di definizione del container `secretOptions` . Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.	1 maggio 2019
AWS Aggiornamento della piattaforma Fargate versione 1.3.0	A partire dal 2 aprile 2019, qualsiasi nuova attività di Fargate che verrà lanciata supporta l'iniezione di dati sensibili nei contenitori archiviando i dati sensibili in AWS Secrets Manager segreti o AWS Systems Manager parametri Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.	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 definizio ne 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 Configurazione del proxy, Dipendenze per i container e Timeout del container.	27 marzo 2019
Amazon ECS presenta il tipo di implementazione esterna	Il tipo di implementazione esterna consente di usare qualsiasi controller di implementazione di terze parti per il controllo completo sul processo di implement azione per un servizio Amazon ECS. Per ulteriori informazioni, consulta Implementa i servizi Amazon ECS utilizzando un controller di terze parti.	27 marzo 2019

Modifica	Descrizione	Data
AWS Contenitori di Deep Learning su Amazon ECS	AWS I Deep Learning Containers sono un set di immagini Docker per addestrare e servire modelli TensorFlow su Amazon Elastic Container Service (Amazon ECS). I Deep Learning Containers forniscono ambienti ottimizzati con TensorFlow librerie Nvidia CUDA (per istanze GPU) e Intel MKL (per istanze CPU) e sono disponibili in Amazon ECR. Per ulteriori informazioni, consulta Utilizzo di AWS Deep Learning Containers su Amazon ECS.	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 Dipendenze per i container.	7 marzo 2019
Amazon ECS presenta l'API `PutAccountSettingDefault`	Amazon ECS introduce l'API `PutAccountSettingDefault` che consente a un utente di impostare lo stato di attivazione del formato ARN/ID predefinito per tutti gli utenti e i ruoli dell'account. In precedenza, l'impostazione dello stato di attivazione predefinito dell'account richiedeva l'uso del proprietario dell'account. Per ulteriori informazioni, consulta Amazon Resource Name (ARN) e ID.	8 febbraio 2019
Amazon ECS supporta carichi di lavoro per GPU	Amazon ECS introduce il supporto per i carichi di lavoro della GPU consentendo di creare cluster con istanze di container abilitate per GPU. In una definizione di attività è possibile specificare il numero di GPU richieste e l'agente ECS vincolerà le GPU fisiche al container. Per ulteriori informazioni, consulta Definizioni di attività Amazon ECS per carichi di lavoro GPU.	4 febbraio 2019

Modifica	Descrizione	Data
Amazon ECS ha esteso il supporto dei segreti	Amazon ECS ha esteso il supporto per l'utilizzo di AWS Secrets Manager segreti direttamente nelle definizioni delle attività per iniettare dati sensibili nei contenitori. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.	21 gennaio 2019
Endpoint VPC di interfaccia (AWS PrivateLink)	Aggiunto il supporto per la configurazione degli endpoint VPC dell'interfaccia con tecnologia AWS PrivateLink. Ciò consente di creare una connessione privata tra il VPC e Amazon ECS senza richiedere l'accesso tramite Internet, un'istanza NAT, una connessione VPN o AWS Direct Connect. Per ulteriori informazioni, consulta Endpoint VPC di interfaccia (AWS PrivateLink).	26 dicembre 2018

Modifica	Descrizione	Data
AWS Piattaforma Fargate versione 1.3.0	Rilasciata la nuova versione della piattaforma AWS Fargate, che contiene: • È stato aggiunto il supporto per l'utilizzo AWS Systems Manager dei parametri Parameter Store per iniettare dati sensibili nei contenitori. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS. • 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 Manutenzione dei processi nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate. Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.	17 dicembre 2018

Modifica	Descrizione	Data
Restrizioni dei servizi aggiornate	Le seguenti restrizioni dei servizi sono state aggiornate: • Il numero di cluster per regione, per account è stato calcolato tra `1000` e `2000`. • Il numero di istanze di container per cluster è stato calcolato tra `1000` e `2000`. • Il numero di servizi per cluster è stato calcolato tra `500` e `1000`. Per ulteriori informazioni, consulta Service Quotas di Amazon ECS.	14 dicembre 2018
AWS Espansione della regione di Fargate	AWS Fargate con Amazon ECS si è estesa alle regioni di Asia Pacifico (Mumbai) e Canada (Centrale). Per ulteriori informazioni, consulta Regioni supportate per Amazon ECS su AWS Fargate.	7 dicembre 2018
Implementazioni blu/verde di Amazon ECS	Amazon ECS ha aggiunto il supporto per le distribuzioni blu/green utilizzando. CodeDeploy Questo tipo di distribuzione consente di verificare una nuova distribuzione di un servizio prima che questo riceva traffico di produzione. Per ulteriori informazioni, consulta Convalida lo stato di un servizio Amazon ECS prima della distribuzione .	27 novembre 2018
AMI Amazon Linux 2 (arm64) ottimizzata per Amazon ECS rilasciata	Amazon ECS ha rilasciato un'AMI Amazon Linux 2 ottimizzata per Amazon ECS per l'architettura arm64. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	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: • Modalità IPC • Modalità PID	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. Per ulteriori informazioni, consulta Trasferisci dati sensibili a un contenitore Amazon ECS.	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. Per ulteriori informazioni, consulta Taggare le risorse Amazon ECS.	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). Per ulteriori informazioni, consulta AWS Fargateper Amazon ECS.	7 novembre 2018

Modifica	Descrizione	Data
Restrizioni dei servizi aggiornate	Le seguenti restrizioni dei servizi sono state aggiornate: • Il numero di processi che utilizzano il tipo di avvio Fargate, per regione, per account è stato aumentato da 20 a 50. • Il numero di indirizzi IP pubblici per i processi che utilizzano il tipo di avvio Fargate è stato aumentato da 20 a 50. Per ulteriori informazioni, consulta Service Quotas di Amazon ECS.	31 ottobre 2018
AWS Espansione della regione di Fargate	AWS Fargate con Amazon ECS si è estesa alla regione Europa (Londra). Per ulteriori informazioni, consulta AWS Fargateper Amazon ECS.	26 ottobre 2018
AMI Amazon Linux 2 ottimizzata per Amazon ECS rilasciata	Amazon ECS fornisce AMI Linux ottimizzate per il servizio in due varianti. La versione più recente e consigliata si basa su x;. Amazon ECS fornisce anche AMI basate 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. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	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_METADATA_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 statistiche Docker 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 Monitora i carichi di lavoro utilizzando i metadati Amazon ECS.	18 ottobre 2018
Espansione della regione per l'individuazione dei servizi di Amazon ECS	L'individuazione dei servizi di Amazon ECS ha esteso il supporto alle regioni Canada (Centrale), Sud America (San Paolo), Asia Pacifico (Seoul), Asia Pacifico (Mumbai) ed Europa (Parigi). Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.	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: • Controlli di sistema • Interactive • Pseudoterminale	17 settembre 2018

Modifica	Descrizione	Data
Supporto per l'autenti cazione del registro privato per Amazon ECS utilizzando le attività di AWS Fargate	Amazon ECS ha introdotto il supporto per i processi Fargate con l'autenticazione di registri privati tramite AWS Secrets Manager. Questa funzione permette di archiviare in modo sicuro le proprie credenziali e quindi consultarle nella definizione del container, consenten do quindi alle attività di utilizzare immagini private. Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.	10 settembre 2018
Espansione della regione per l'individuazione dei servizi di Amazon ECS	Il servizio di individuazione dei servizi di Amazon ECS ha esteso il supporto per le regioni Asia Pacifico (Singapore), Asia Pacifico (Sydney), Asia Pacifico (Tokyo), Europa (Francoforte) e Europa (Londra). Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.	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 AWS Secrets Manager	Amazon ECS ha introdotto il supporto per l'autenti cazione di registri privati tramite AWS Secrets Manager. Questa funzione permette di archiviare in modo sicuro le proprie credenziali e quindi consultarle nella definizione del container, consentendo quindi alle attività di utilizzare immagini private. Per ulteriori informazioni, consulta Utilizzo di immagini non AWS containerizzate in Amazon ECS.	16 agosto 2018

Modifica	Descrizione	Data
Aggiunta del supporto dei volumi Docker	Amazon ECS ha introdotto il supporto per i volumi Docker. Per ulteriori informazioni, consulta Opzioni di storage per le attività di Amazon ECS.	9 agosto 2018
AWS Espansione della regione di Fargate	AWS Fargate con Amazon ECS si è estesa alle regioni Europa (Francoforte), Asia Pacifico (Singapore) e Asia Pacifico (Sydney). Per ulteriori informazioni, consulta AWS Fargateper Amazon ECS.	19 luglio 2018

Modifica	Descrizione	Data
Strategie del pianificatore del servizio Amazon ECS aggiunte	Amazon ECS ha introdotto il concetto di strategie del pianificatore di servizi. Sono disponibili due strategie del pianificatore del servizio: • `REPLICA`: la strategia di pianificazione delle repliche colloca e gestisce il numero desiderato di attività nel cluster. Di default, il pianificatore del servizio distribuisce le attività tra le zone di disponibilità. Puoi utilizzare vincoli e strategie di posizionamento delle attività per personalizzare le decisione riguardo al posizionamento delle attività. Per ulteriori informazi oni, consulta Strategia di replica. • `DAEMON`: la strategia di pianificazione del daemon distribuisce esattamente un'attività in ciascuna istanza di container attiva, che soddisfa tutti i vincoli di posizionamento delle attività specificati nel cluster. Quando si utilizza questa strategia, non è necessari o specificare un numero di attività desiderato o una strategia di posizionamento delle attività, né utilizzar e le policy di Auto Scaling del servizio. Per ulteriori informazioni, consulta Strategia daemon. **Note** I processi Fargate non supportano la strategia di pianificazione `DAEMON` .	12 giugno 2018

Modifica	Descrizione	Data
Agente del container Amazon ECS v1.18.0	Rilascio della nuova versione dell'agente del container Amazon ECS, con aggiunta della seguente funzionalità: • Aggiunto il supporto per la personalizzazione del comportamento pull dell'immagine dell'agente del container utilizzando il parametro `ECS_IMAGE_PULL_BEHAVIOR` . Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS. Per ulteriori informazioni, consulta amazon-ecs-agent github.	24 maggio 2018
Aggiunto il supporto per le modalità di rete `bridge` e `host` quando si configura l'individuazione dei servizi	Aggiunto il supporto per la configurazione dell'individuazione dei servizi per i servizi Amazon ECS tramite le definizioni di processi che specificano le modalità di rete `bridge` o `host`. Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.	22 maggio 2018
Aggiunto il supporto per altri parametri di metadati dell'AMI ottimizzata per Amazon ECS	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 Recupero di metadati AMI Linux ottimizzati per Amazon ECS.	9 maggio 2018

Modifica	Descrizione	Data
AWS Espansione della regione di Fargate	AWS Fargate con Amazon ECS si è estesa alle regioni Stati Uniti orientali (Ohio), Stati Uniti occidentali (Oregon) e UE occidentale (Irlanda). Per ulteriori informazioni, consulta AWS Fargateper Amazon ECS.	26 Aprile 2018
Recupero dei metadati dell'AMI ottimizzata per Amazon ECS	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 Recupero di metadati AMI Linux ottimizzati per Amazon ECS.	10 aprile 2018
AWS Versione della piattaforma Fargate	Rilasciata la nuova versione della piattaforma AWS Fargate, che contiene: • Aggiunta del supporto per Monitora i carichi di lavoro utilizzando i metadati Amazon ECS. • Aggiunta del supporto per Controllo dello stato. • Aggiunto supporto per Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS Per ulteriori informazioni, consulta Versioni della piattaforma Fargate Linux per Amazon ECS.	26 marzo 2018
Individuazione dei servizi di Amazon ECS	Aggiunta dell'integrazione con Route 53 per supportar e l'individuazione dei servizi di Amazon ECS. Per ulteriori informazioni, consulta Usa Service Discovery per connettere i servizi Amazon ECS con i nomi DNS.	22 marzo 2018

Modifica	Descrizione	Data
Supporto per i parametri Docker shm-size e tmpfs	Aggiunto il supporto per i parametri Docker shm-size e tmpfs nelle definizioni di processo di Amazon ECS. Per ulteriori informazioni sulla sintassi della CLI ECS aggiornata, consulta Parametri Linux.	20 marzo 2018
Controlli dell'integrità dei container	Aggiunta del supporto per i controlli dello stato Docker nelle definizioni dei container. Per ulteriori informazi oni, consulta Controllo dello stato.	8 marzo 2018
AWS Fargate	È stata aggiunta una panoramica per Amazon ECS con AWS Fargate. Per ulteriori informazioni, consulta AWS Fargateper Amazon ECS.	22 febbraio 2018
Endpoint di metadati dei processi Amazon ECS	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. Per ulteriori informazioni, consulta Monitora i carichi di lavoro utilizzando i metadati Amazon ECS.	8 febbraio 2018
Auto Scaling del servizio Amazon ECS tramite policy di monitoraggio degli obiettivi	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 Scala il tuo servizio Amazon ECS utilizzando un valore metrico target. Rimozione del tutorial precedente per il dimension amento 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.	8 febbraio 2018
Supporto Docker 17.09	Supporto aggiunto per Docker 17.09. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	18 gennaio 2018

Modifica	Descrizione	Data
Nuovo comportamento del pianificatore del servizio	Aggiornamento delle informazioni sul comportamento per le attività di servizio che non è stato possibile avviare. Nuovo messaggio documentato relativo a un evento di servizio che si attiva quando un'attività di servizio presenta errori consecutivi.	11 gennaio 2018
Periodo di attesa per l'inizializzazione del controllo dell'integrità di Elastic Load Balancing	Aggiunta della possibilità di specificare un periodo di attesa per i controlli dello stato.	27 dicembre 2017
CPU e memoria a livello di attività	Aggiunta del supporto per la specifica di CPU e memoria a livello di attività nelle definizioni di processo. Per ulteriori informazioni, vedere. TaskDefinition	12 dicembre 2017
Ruolo di esecuzione di attività	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: • Chiamate ad Amazon ECR per recuperare l'immagine del container • Chiamate per CloudWatch archiviare i registri delle applicazioni del contenitore Per ulteriori informazioni, consulta Ruolo IAM di esecuzione di attività Amazon ECS.	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 Varianti AMI ottimizzate per Amazon ECS.	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 Tipi di avvio di Amazon ECS.	29 novembre 2017
Modifica del nome Amazon ECS	Amazon Elastic Container Service è stato rinominato (in precedenza era Amazon EC2 Container Service).	21 Novembre 2017
Reti di attività	Le funzionalità di rete dei processi fornite dalla modalità di rete awsvpc forniscono ai processi 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 reti di attività semplifica le reti dei container e offre maggiore controllo sul modo in cui le applicazioni containerizzate comunicano tra loro e con gli altri servizi all'interno dei VPC. Per ulteriori informazioni, consulta Opzioni di task networking di Amazon ECS per il tipo di lancio EC2.	14 Novembre 2017
Metadati di container di Amazon ECS	I container Amazon ECS sono ora in grado di accedere a metadati quali il container Docker o l'ID immagine, la configurazione di rete o gli ARN di Amazon. Per ulteriori informazioni, consulta File di metadati di container di Amazon ECS.	2 Novembre 2017
Supporto Docker 17.06	Supporto aggiunto per Docker 17.06. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	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 `LinuxParameters` (`devices` e `initProcessEnabled`). Per ulteriori informazioni, vedere LinuxParameters.	2 Novembre 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 `LinuxParameters` (`capabilities`). Per ulteriori informazioni, vedere LinuxParameters.	22 settembre 2017
Supporto per Network Load Balancer	Amazon ECS ha aggiunto il supporto per i Network Load Balancer nella console Amazon ECS.	7 settembre 2017
RunTask sostituzioni	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 Esecuzione di un'applicazione come attività Amazon ECS.	27 giugno 2017
Processi pianificati di Amazon ECS	Aggiunta del supporto per le attività di pianificazione tramite cron.	7 giugno 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 Avvio di un'istanza di container Linux di Amazon ECS.	6 giugno 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 2017

Modifica	Descrizione	Data
Microservizi e attività in batch	Aggiunta la documentazione per due casi di utilizzo comune per Amazon ECS: microservizi e processi batch. Per ulteriori informazioni, consulta Informazioni relative ad Amazon ECS.	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 Drenaggio delle istanze di container Amazon ECS.	24 gennaio 2017
Supporto Docker 1.12	Supporto aggiunto per Docker 1.12. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	24 gennaio 2017
Nuovo 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 Usa strategie per definire il posizionamento delle attività di Amazon ECS.	29 dicembre 2016
Supporto del container Windows in versione beta	Aggiunto il supporto per i container Windows 2016 (beta). Per ulteriori informazioni, consulta Varianti AMI ottimizzate per Amazon ECS.	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 Pianifica i tuoi contenitori su Amazon ECS.	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 Automatizza le risposte agli errori di Amazon ECS utilizzando EventBridge.	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. CloudWatc h Per ulteriori informazioni, consulta Invia i log di Amazon ECS a CloudWatch .	12 settembre 2016
Servizi Amazon ECS con supporto Elastic Load Balancing per porte dinamiche	Aggiunta del supporto per un load balancer per supportare più combinazioni di istanza/porta per listener, con aumento della flessibilità per i container . Ora è possibile consentire a Docker di definire in modo dinamico la porta host del container; il pianifica tore ECS registra la combinazione istanza/porta nel load balancer. Per ulteriori informazioni, consulta Usa il bilanciamento del carico per distribuire il traffico del servizio Amazon ECS.	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 Ruolo IAM dell'attività Amazon ECS.	13 luglio 2016
Supporto Docker 1.11	Supporto aggiunto per Docker 1.11. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	31 maggio 2016
Scalabilità automatica attività	Amazon ECS ha aggiunto il supporto per il dimension amento automatico dei processi eseguiti da un servizio. Per ulteriori informazioni, consulta Ridimensi ona automaticamente il tuo servizio Amazon ECS.	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 definizio ne attività. Per ulteriori informazioni, consulta. ListTaskDefinitions	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 AMI Linux ottimizzate per Amazon ECS.	5 aprile 2016
Supporto Docker 1.9	Supporto aggiunto per Docker 1.9. Per ulteriori informazioni, consulta AMI Linux ottimizzate per Amazon ECS.	22 dicembre 2015
CloudWatch metriche per la prenotazione della CPU e della memoria del cluster	Amazon ECS ha aggiunto CloudWatch parametri personalizzati per la prenotazione di CPU e memoria.	22 dicembre 2015
Nuova esperienza per la prima esecuzione di Amazon ECS	Nell'esperienza per la prima esecuzione della console Amazon ECS è stata aggiunta la creazione di ruoli con zero clic.	23 novembre 2015
Posizionamento delle attività nelle zone di disponibilità	Nel pianificatore del servizio Amazon ECS è stato aggiunto il supporto per il posizionamento dei processi tra le zone di disponibilità.	8 ottobre 2015
CloudWatch metriche per cluster e servizi Amazon ECS	Amazon ECS ha aggiunto CloudWatch parametri personalizzati per l'utilizzo della CPU e della memoria per ogni istanza di container, servizio e famiglia di definizione delle attività in un cluster. Queste nuove metriche possono essere utilizzate per ridimensionare le istanze di container in un cluster utilizzando i gruppi di Auto Scaling o per creare allarmi personalizzati. CloudWatch	17 agosto 2015

Modifica	Descrizione	Data
Supporto per la porta UDP	Aggiunto il supporto per porte UDP nelle definizioni di processo.	7 luglio 2015
Sostituzioni delle variabili di ambiente	È stato aggiunto il supporto deregisterTaskDefinition e l'override delle variabili di ambiente per RunTask.	18 giugno 2015
Aggiornamenti automatici dell'agente Amazon ECS	Aggiunta della possibilità di visualizzare la versione dell'agente ECS in esecuzione in un'istanza di container. Inoltre, è in grado di aggiornare l'agente ECS da, e SDK. AWS Management Console AWS CLI	11 giugno 2015
Integrazione del pianificatore del servizio Amazon ECS e Elastic Load Balancing	Aggiunta della possibilità di definire un servizio e associare tale servizio a un load balancer di Elastic Load Balancing.	9 aprile 2015
GA di Amazon ECS	Disponibilità generale di Amazon ECS nelle regioni Stati Uniti orientali (Virginia settentrionale), Stati Uniti occidentali (Oregon), Asia Pacifico (Tokyo) ed Europa (Irlanda).	9 aprile 2015

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.