aws

Guida per l'utente

AWS Batch

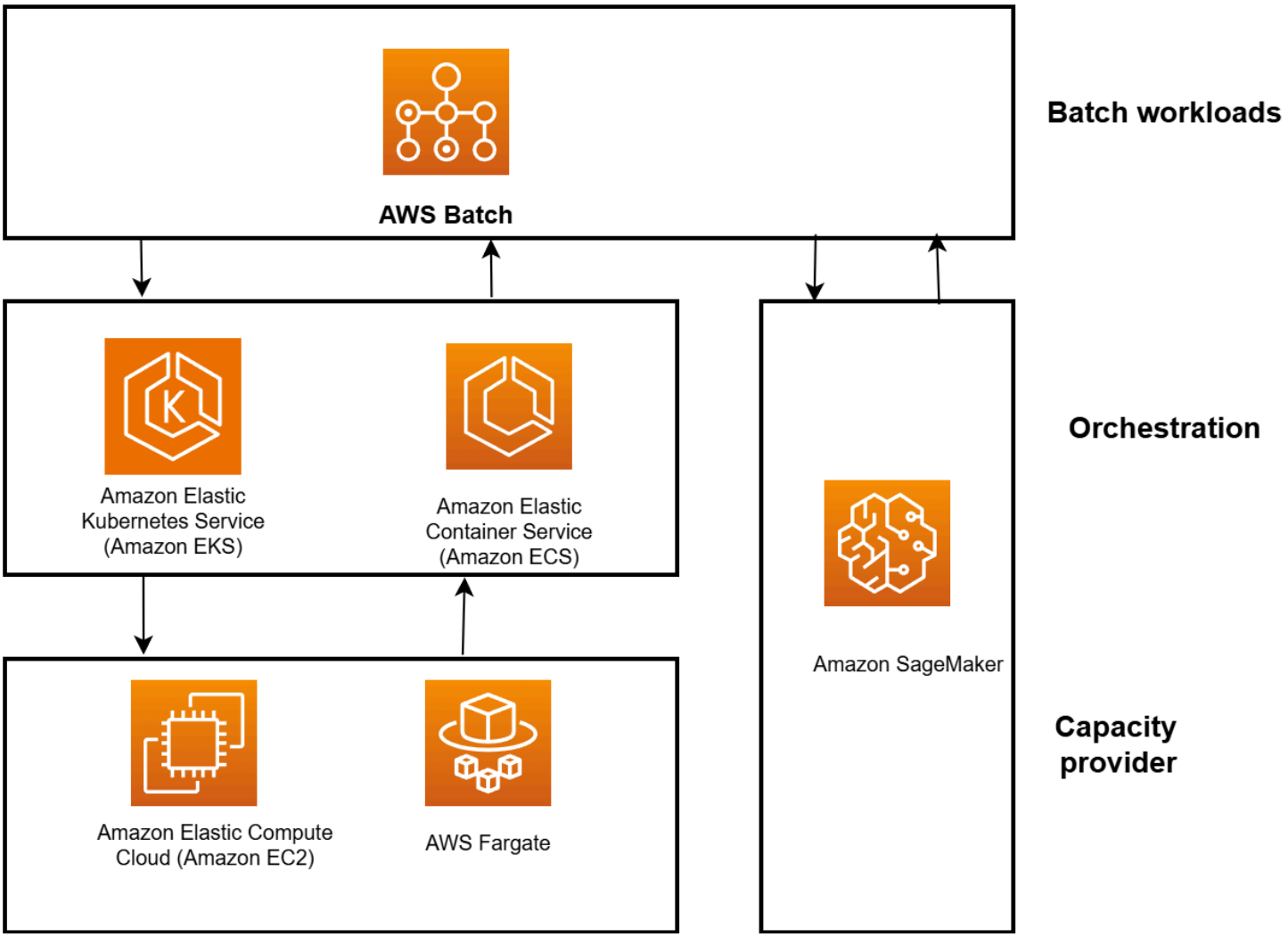
AWS Batch: Guida per l’utente

Table of Contents

Che cos'è AWS Batch?

AWS Batch ti aiuta a eseguire carichi di lavoro di elaborazione in batch su. Cloud AWS Il Batch computing è un modo comune per sviluppatori, scienziati e ingegneri di accedere a grandi quantità di risorse di elaborazione. AWS Batch elimina il peso indifferenziato della configurazione e della gestione dell'infrastruttura richiesta, analogamente al tradizionale software di elaborazione in batch. Questo servizio è in grado di effettuare il provisioning in modo efficiente delle risorse in risposta ai processi inviati per eliminare i vincoli di capacità, ridurre i costi di calcolo e fornire i risultati in modo rapido.

Essendo un servizio completamente gestito, AWS Batch consente di eseguire carichi di lavoro di elaborazione in batch di qualsiasi dimensione. AWS Batch effettua automaticamente il provisioning delle risorse di elaborazione e ottimizza la distribuzione del carico di lavoro in base alla quantità e alla scala dei carichi di lavoro. Con AWS Batch, non è necessario installare o gestire software di elaborazione in batch, quindi puoi concentrare il tuo tempo sull'analisi dei risultati e sulla risoluzione dei problemi.

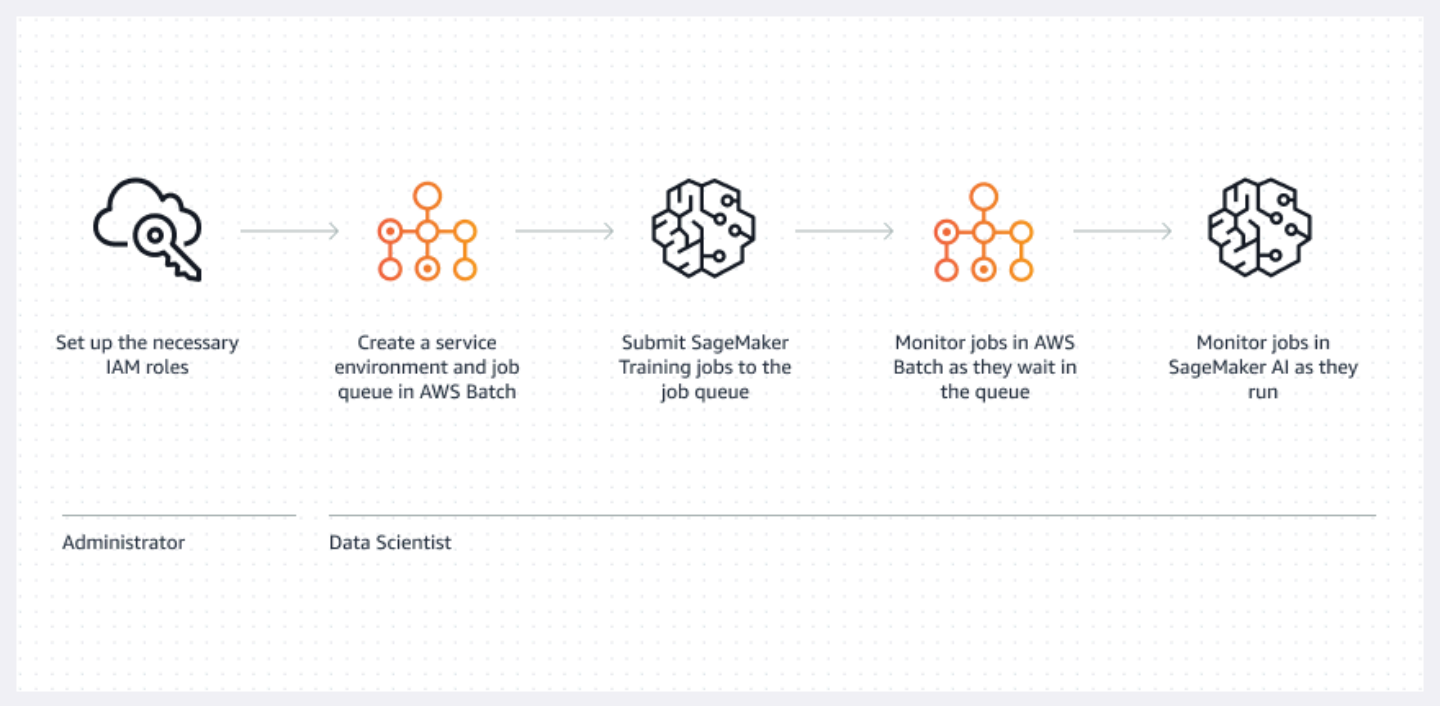


AWS Batch fornisce tutte le funzionalità necessarie per eseguire carichi di lavoro su larga scala e ad alta intensità di calcolo oltre ai servizi di orchestrazione dei container AWS gestiti, Amazon ECS e Amazon EKS. AWS Batch è in grado di scalare la capacità di calcolo su EC2 istanze Amazon e risorse Fargate.

AWS Batch fornisce un servizio completamente gestito per carichi di lavoro in batch e offre le funzionalità operative per ottimizzare questi tipi di carichi di lavoro in termini di produttività, velocità, efficienza delle risorse e costi.

AWS Batch consente inoltre di mettere in coda i lavori di SageMaker formazione, permettendo ai data scientist e agli ingegneri di machine learning di inviare lavori di formazione con priorità a code configurabili. Questa funzionalità garantisce che i carichi di lavoro ML vengano eseguiti automaticamente non appena le risorse diventano disponibili, eliminando la necessità di coordinamento manuale e migliorando l'utilizzo delle risorse.

Per i carichi di lavoro di machine learning, AWS Batch offre funzionalità di accodamento per i lavori di formazione. SageMaker È possibile configurare le code con politiche specifiche per ottimizzare costi, prestazioni e allocazione delle risorse per i carichi di lavoro di formazione ML.



Ciò fornisce un modello di responsabilità condivisa in cui gli amministratori configurano l'infrastruttura e le autorizzazioni, mentre i data scientist possono concentrarsi sull'invio e sul monitoraggio dei propri carichi di lavoro di formazione ML. I lavori vengono messi in coda ed eseguiti automaticamente in base alle priorità configurate e alla disponibilità delle risorse.

Sei un utente alle prime armi AWS Batch ?

Se sei un utente alle prime armi di AWS Batch, ti consigliamo di iniziare leggendo le seguenti sezioni:

- Componenti di AWS Batch
- Crea account IAM e utente amministrativo
- Configurazione AWS Batch
- Guida introduttiva ai AWS Batch tutorial
- Guida introduttiva all'intelligenza artificiale AWS Batch SageMaker

Servizi correlati

AWS Batch è un servizio di elaborazione in batch completamente gestito che pianifica, pianifica ed esegue carichi di lavoro di machine learning, simulazione e analisi in batch containerizzati su tutta la gamma di offerte di AWS elaborazione, come Amazon ECS, Amazon AWS Fargate EKS e istanze Spot o On-Demand. Per ulteriori informazioni su ciascun servizio di elaborazione gestito, consulta:

- Guida per EC2 l'utente di Amazon
- Guida per sviluppatori di AWS Fargate
- Guida per l'utente di Amazon EKS
- Guida per sviluppatori di Amazon SageMaker AI

Accesso AWS Batch

È possibile accedere AWS Batch utilizzando quanto segue:

AWS Batch console

L'interfaccia web in cui è possibile creare e gestire le risorse.

AWS Command Line Interface

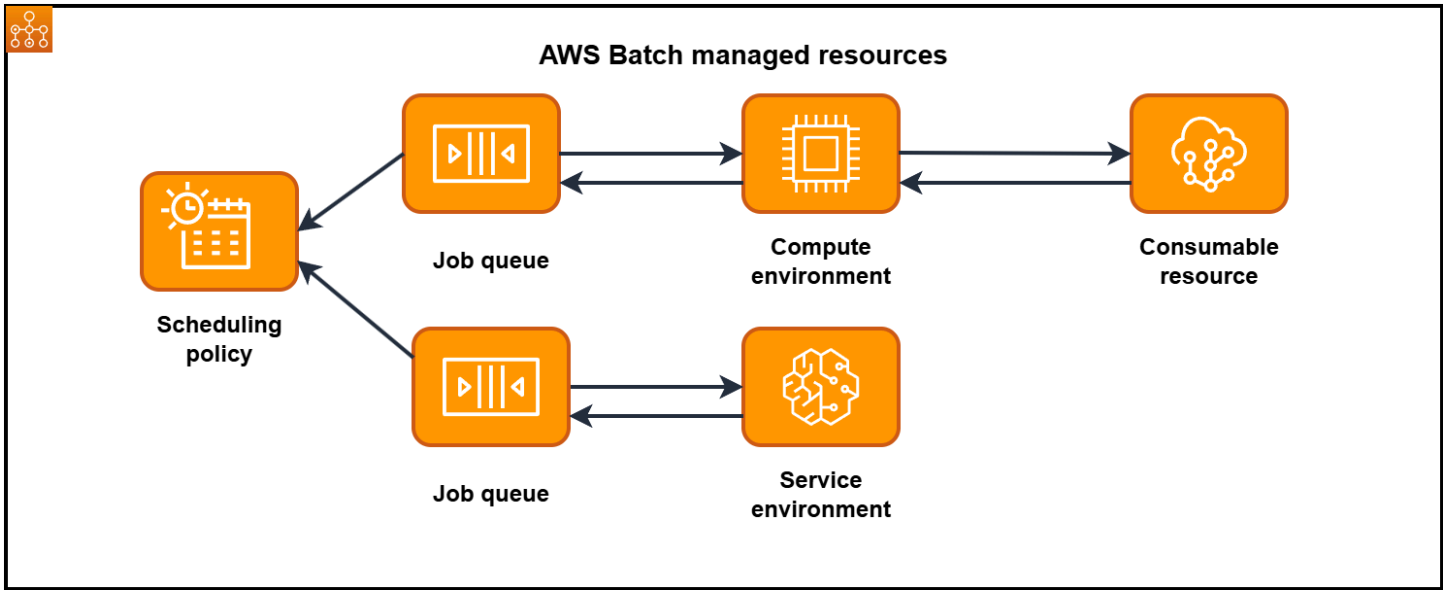
Interagisci Servizi AWS utilizzando i comandi nella shell della riga di comando. AWS Command Line Interface È supportato su Windows, macOS e Linux. Per ulteriori informazioni su AWS CLI, vedere la Guida per AWS Command Line Interface l'utente. È possibile trovare i AWS Batch comandi nella Guida ai AWS CLI comandi.

AWS SDKs

Se preferisci creare applicazioni utilizzando specifiche lingue APIs anziché inviare una richiesta tramite HTTP o HTTPS, utilizza le librerie, il codice di esempio, i tutorial e altre risorse forniti da. AWS Queste librerie forniscono funzioni di base che automatizzano le attività, come la firma crittografica delle richieste, il ritentativo delle richieste e la gestione delle risposte agli errori. Queste funzioni consentono di iniziare in modo più efficiente. Per ulteriori informazioni, consulta Strumenti su cui basarsi AWS.

Componenti di AWS Batch

AWS Batch semplifica l'esecuzione di lavori in batch su più zone di disponibilità all'interno di una regione. È possibile creare ambienti di calcolo AWS Batch con un VPC nuovo o esistente. Dopo avere configurato un ambiente di calcolo e averlo associato a una coda di processi, è possibile creare definizioni di processi che specificano le immagini di container Docker per eseguire i processi. Le immagini di container sono archiviate in ed estratte da registri dei container, che possono essere interni o esterni all'infrastruttura AWS .



Ambiente di calcolo

Un ambiente di calcolo è un set di risorse di calcolo gestite o non gestite usate per eseguire i processi. Con gli ambienti di elaborazione gestiti, puoi specificare il tipo di elaborazione desiderato (Fargate o EC2) a diversi livelli di dettaglio. È possibile configurare ambienti di calcolo che utilizzano un particolare tipo di EC2 istanza, un modello particolare come o. `c5.2xlarge` `m5.10xlarge` In alternativa, puoi scegliere solo di specificare che desideri utilizzare i tipi di istanza più recenti. Puoi anche specificare il numero minimo, desiderato e massimo di v CPUs per l'ambiente, oltre all'importo che sei disposto a pagare per un'istanza Spot come percentuale del prezzo dell'istanza on demand e un set target di sottoreti VPC. AWS Batch avvia, gestisce e termina in modo efficiente i tipi di elaborazione in base alle esigenze. Puoi inoltre gestire i tuoi ambienti di calcolo. Pertanto, sei responsabile della configurazione e del ridimensionamento delle istanze in un cluster Amazon ECS AWS Batch creato per te. Per ulteriori informazioni, consulta Ambienti di calcolo per AWS Batch.

Job queues

Quando invii un AWS Batch lavoro, lo invii a una particolare coda di lavoro, dove il lavoro rimane fino a quando non viene programmato in un ambiente di elaborazione. Associate uno o più ambienti di elaborazione a una coda di lavoro. È inoltre possibile assegnare valori di priorità a questi ambienti di elaborazione e persino tra le code di lavoro stesse. Ad esempio, è possibile avere una coda ad alta priorità a cui inviare i lavori con priorità urgente e una coda a bassa priorità per i lavori che possono essere eseguiti in qualsiasi momento quando le risorse di elaborazione sono più economiche. Per ulteriori informazioni, consulta Job queues.

Definizioni del lavoro

Una definizione di processo specifica come devono essere eseguiti i lavori. Puoi pensare a una definizione di lavoro come a un modello per le risorse del tuo lavoro. Puoi assegnare al tuo lavoro un ruolo IAM per fornire l'accesso ad altre AWS risorse. È inoltre necessario specificare i requisiti di memoria e CPU. La definizione del processo può anche controllare le proprietà del container, le variabili di ambiente e i punti di montaggio per lo storage persistente. Molte specifiche in una definizione di processo possono essere sovrascritte indicando nuovi valori al momento dell'invio dei singoli processi. Per ulteriori informazioni, consulta Definizioni del lavoro

Processi

Un'unità di lavoro (ad esempio uno script shell, un eseguibile Linux o un'immagine di container Docker) inviata a AWS Batch. Ha un nome e viene eseguito come applicazione containerizzata su o EC2 risorse AWS Fargate Amazon nel tuo ambiente di calcolo, utilizzando i parametri specificati in una definizione di processo. I lavori possono fare riferimento ad altri lavori per nome o per ID e possono dipendere dal completamento con successo di altri lavori o dalla disponibilità delle risorse specificate. Per ulteriori informazioni, consulta Jobs.

Politica di pianificazione

È possibile utilizzare le politiche di pianificazione per configurare il modo in cui le risorse di elaborazione in una coda di lavoro vengono allocate tra utenti o carichi di lavoro. Utilizzando politiche di pianificazione con condivisione equa, puoi assegnare identificatori di condivisione diversi a carichi di lavoro o utenti. Per impostazione predefinita, il AWS Batch job scheduler utilizza una strategia FIFO (first-in, first-out). Per ulteriori informazioni, consulta Politiche di pianificazione con condivisione equa.

Risorse consumabili

Una risorsa consumabile è una risorsa necessaria per eseguire i lavori, ad esempio un token di licenza di terze parti, la larghezza di banda per l'accesso al database, la necessità di limitare le chiamate a un'API di terze parti e così via. Si specificano le risorse consumabili necessarie per l'esecuzione di un processo e Batch tiene conto di queste dipendenze tra le risorse quando pianifica un lavoro. È possibile ridurre il sottoutilizzo delle risorse di elaborazione allocando solo i lavori che dispongono di tutte le risorse necessarie. Per ulteriori informazioni, consulta Pianificazione basata sulle risorse .

Ambiente di servizio

Un ambiente di servizio definisce il modo in cui AWS Batch si integra con SageMaker l'esecuzione del lavoro. Gli ambienti di servizio AWS Batch consentono di inviare e gestire i lavori fornendo al SageMaker contempo le funzionalità di coda, pianificazione e gestione delle priorità di. AWS Batch Gli ambienti di servizio definiscono i limiti di capacità per tipi di servizi specifici, ad esempio i lavori di formazione SageMaker . I limiti di capacità controllano le risorse massime che possono essere utilizzate dai lavori di assistenza nell'ambiente. Per ulteriori informazioni, consulta Ambienti di servizio per AWS Batch.

Lavoro di servizio

Un service job è un'unità di lavoro che viene inviata AWS Batch per essere eseguita in un ambiente di servizio. I lavori di assistenza sfruttano le funzionalità AWS Batch di coda e pianificazione di cui dispongono, delegando al contempo l'esecuzione effettiva al servizio esterno. Ad esempio, i lavori di SageMaker formazione inviati come lavori di servizio vengono messi in coda e hanno priorità per AWS Batch, ma l'esecuzione dei lavori di formazione avviene all'interno dell'infrastruttura di intelligenza artificiale. SageMaker SageMaker Questa integrazione consente ai data scientist e agli ingegneri ML di trarre vantaggio dalla gestione automatizzata dei carichi AWS Batch di lavoro e dall'accodamento prioritario per i loro carichi di lavoro di formazione sull'intelligenza artificiale. SageMaker I lavori di assistenza possono fare riferimento ad altri lavori per nome o ID e supportare le dipendenze tra i lavori. Per ulteriori informazioni, consulta Lavori di assistenza in AWS Batch.

Configurazione AWS Batch

Se ti sei già registrato ad Amazon Web Services (AWS) e utilizzi Amazon Elastic Compute Cloud (Amazon EC2) o Amazon Elastic Container Service (Amazon ECS), presto potrai utilizzarlo. AWS Batch Il processo di configurazione per questi servizi è simile. Questo perché AWS Batch utilizza istanze di container Amazon ECS nei suoi ambienti di elaborazione. Per utilizzare AWS CLI with AWS Batch, è necessario utilizzare una versione di AWS CLI che supporti le funzionalità più recenti. AWS Batch Se non vedi il supporto per una AWS Batch funzionalità in AWS CLI, esegui l'aggiornamento alla versione più recente. Per ulteriori informazioni, consulta http://aws.amazon.com/cli/.

> **Note**
>
> Poiché AWS Batch utilizza componenti di Amazon EC2, utilizzi la EC2 console Amazon per molti di questi passaggi.

Completa le seguenti attività per prepararti AWS Batch.

Argomenti

- Crea account IAM e utente amministrativo
- Crea ruoli IAM per i tuoi ambienti di calcolo e le istanze di container
- Crea una key pair per le tue istanze
- Crea un VPC
- Creazione di un gruppo di sicurezza
- Installa il AWS CLI

Crea account IAM e utente amministrativo

Per iniziare, devi creare un AWS account e un singolo utente a cui in genere vengono concessi i diritti amministrativi. A tale scopo, completa i seguenti tutorial:

Registrati 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 https://portal.aws.amazon.com/billing/registrazione.
2. Segui le istruzioni online.

   Nel corso della procedura di registrazione riceverai una telefonata o un messaggio di testo e ti verrà chiesto di inserire un codice di verifica attraverso la tastiera del telefono.

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

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

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

1. Accedi Console di gestione AWScome 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 il Centro identità IAM.

Per istruzioni, consulta Abilitazione di AWS IAM Identity Center nella Guida per l'utente di AWS IAM Identity Center .

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

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

Accesso come utente amministratore

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

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

Assegna l'accesso a ulteriori utenti

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

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

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

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

Crea ruoli IAM per i tuoi ambienti di calcolo e le istanze di container

I tuoi ambienti di AWS Batch calcolo e le istanze dei container richiedono Account AWS credenziali per effettuare chiamate ad altri utenti per tuo AWS APIs conto. Crea un AWS Identity and Access Management ruolo che fornisca queste credenziali ai tuoi ambienti di calcolo e alle istanze di container, quindi associa quel ruolo ai tuoi ambienti di elaborazione.

> **Note**
>
> Per verificare di disporre delle autorizzazioni richieste, consulta Configurazione iniziale del servizio IAM per il tuo Account AWS account.
> L'ambiente di AWS Batch calcolo e i ruoli delle istanze del contenitore vengono creati automaticamente durante la prima esecuzione della console. Quindi, se intendi utilizzare la AWS Batch console, puoi passare alla sezione successiva. Se intendi utilizzare AWS CLI invece, completa le procedure in Utilizzo di ruoli collegati ai servizi per AWS Batch e Tutorial: Creare il ruolo di esecuzione IAM prima di creare il tuo primo ambiente di elaborazione. Ruolo dell'istanza Amazon ECS

Crea una key pair per le tue istanze

AWS utilizza la crittografia a chiave pubblica per proteggere le informazioni di accesso dell'istanza. Un'istanza Linux, ad esempio un'istanza contenitore per un ambiente di AWS Batch calcolo, non ha alcuna password da utilizzare per l'accesso SSH. Viene utilizzata una coppia di chiavi per accedere in modo sicuro all'istanza. Specifichi il nome della coppia di chiavi al momento della creazione dell'ambiente di calcolo, quindi fornisci la chiave privata quando accedi usando SSH.

Se non hai già creato una key pair, puoi crearne una utilizzando la EC2 console Amazon. Tieni presente che, se prevedi di avviare più istanze Regioni AWS, crea una key pair in ciascuna regione. Per ulteriori informazioni sulle regioni, consulta Regioni e zone di disponibilità nella Amazon EC2 User Guide.

Per creare una coppia di chiavi

1. Apri la EC2 console Amazon all'indirizzo https://console.aws.amazon.com/ec2/.
2. Dalla barra di navigazione, seleziona un nome Regione AWS per la key pair. Puoi selezionare qualsiasi regione disponibile, indipendentemente dalla tua posizione: tuttavia, le coppie di chiavi sono specifiche di una regione. Ad esempio, se prevedi di avviare un'istanza nella regione Stati Uniti occidentali (Oregon), crea una key pair per l'istanza nella stessa regione.
3. Nel riquadro di navigazione scegli Key Pairs (Coppie di chiavi), Create Key Pair (Crea coppia di chiavi).
4. Nella finestra di dialogo Create Key Pair (Crea coppia di chiavi) immetti un nome per la nuova coppia di chiavi in Key pair name (Nome coppia di chiavi) e scegli Create (Crea). Scegli un nome

facile da ricordare, ad esempio il tuo nome utente, seguito da `-key-pair` e più il nome della regione. Ad esempio, me-key-pair-uswest2.

5. Il file della chiave privata viene automaticamente scaricato dal browser. Il nome di base del file è quello specificato come nome della coppia di chiavi e l'estensione è `.pem`. Salvare il file della chiave privata in un luogo sicuro.

   > **Important**
   >
   > Questo è l'unico momento in cui salvare il file della chiave privata. È necessario fornire il nome della coppia di chiavi all'avvio di un'istanza e la chiave privata corrispondente ogni volta che ci si connette all'istanza.

6. Se utilizzi un client SSH su un computer Mac o Linux per connetterti alla tua istanza Linux, usa il comando seguente per impostare le autorizzazioni del tuo file di chiave privata. In questo modo, solo tu puoi leggerlo.

   ```
   $ chmod 400 your_user_name-key-pair-region_name.pem
   ```

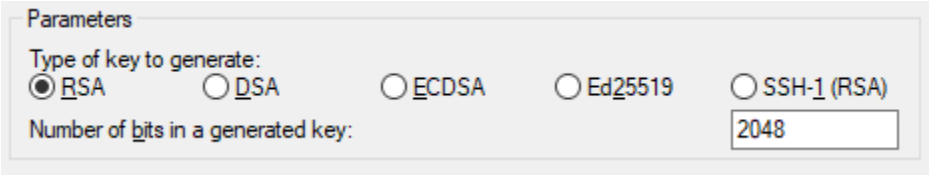
Per ulteriori informazioni, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.

Per stabilire la connessione all'istanza usando la coppia di chiavi

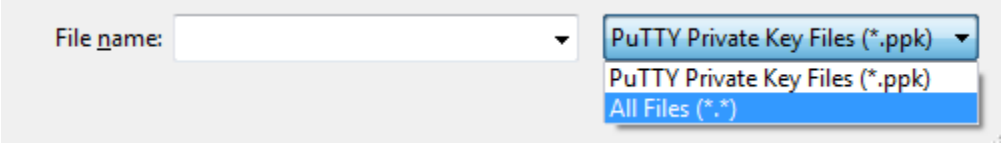
Per connettere l'istanza Linux da un computer che esegue Mac o Linux, specifica il file `.pem` nel client SSH con l'opzione `-i` e il percorso della chiave privata. Per connetterti alla tua istanza Linux da un computer che esegue Windows, usa uno dei due MindTerm o PuTTY. Se si prevede di utilizzare PuTTY, installarlo e utilizzare la procedura seguente per convertire `.pem` il file in un file. `.ppk`

(Facoltativo) Per prepararsi alla connessione a un'istanza Linux da Windows utilizzando PuTTY

1. Scarica e installa PuTTY dal sito all'indirizzo http://www.chiark.greenend.org.uk/~sgtatham/putty/. Assicurarsi di installare l'intera suite.
2. Avvia Pu TTYgen (ad esempio, dal menu Start, scegli Tutti i programmi, PuTTY e Pu). TTYgen
3. In Type of key to generate (Tipo di chiave da generare) scegliere RSA. Se utilizzi una versione precedente di PuTTYgen, scegli SSH-2 RSA.



4. Scegli Carica. Per impostazione predefinita, Pu TTYgen visualizza solo i file con l'estensione. `.ppk` Per individuare il file `.pem`, scegli l'opzione per visualizzare tutti i tipi di file.



5. Selezionare il file di chiave privata creato nella procedura precedente e scegliere Open (Apri). Scegliere OK per chiudere la finestra di dialogo di conferma.
6. Scegli Salva chiave privata. Pu TTYgen visualizza un avviso relativo al salvataggio della chiave senza una passphrase. Scegliere Yes (Sì).
7. Specifica lo stesso nome per la chiave usato per la coppia di chiavi. PuTTY aggiunge automaticamente l'estensione di file `.ppk`.

Crea un VPC

Con Amazon Virtual Private Cloud (Amazon VPC), puoi lanciare AWS risorse in una rete virtuale che hai definito. Ti consigliamo vivamente di avviare le istanze dei container in un VPC.

Se disponi di un VPC predefinito, puoi anche saltare questa sezione e passare all'attività successiva. Creazione di un gruppo di sicurezza Per determinare se disponi di un VPC predefinito, consulta Supported Platforms in Amazon EC2 Console nella Amazon EC2 User Guide

Per informazioni su come creare un Amazon VPC, consulta Create a VPC only nella Amazon VPC User Guide. Fai riferimento alla 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 blocco CIDR	IPv4 Inserimento manuale CIDR La dimensione del blocco CIDR deve essere compresa tra /16 e /28.
IPv6 blocco CIDR	Nessun blocco IPv6 CIDR
Tenancy	Predefinita

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

Creazione di un gruppo di sicurezza

I gruppi di sicurezza fungono da firewall per le istanze di container dell'ambiente di calcolo associate, controllando sia il traffico in entrata che quello in uscita a livello di istanza di container. Un gruppo di sicurezza può essere utilizzato solo nel VPC per cui viene creato.

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.

Tieni presente che se prevedi di avviare istanze di container in più regioni, devi creare un gruppo di sicurezza in ciascuna regione. Per ulteriori informazioni, consulta Regioni e zone di disponibilità nella Amazon EC2 User Guide.

> **Note**
>
> È necessario l'indirizzo IP pubblico del computer locale, che puoi ottenere usando un servizio. Ad esempio, forniamo il seguente servizio: http://checkip.amazonaws.com/ o https://checkip.amazonaws.com/. Per individuare un altro servizio che fornisce l'indirizzo IP, utilizza la frase di ricerca "qual è il mio indirizzo IP". Se ti connetti tramite un provider di servizi

Internet (ISP) o da un firewall senza un indirizzo IP statico, scopri l'intervallo di indirizzi IP utilizzati dai computer client.

Per creare un gruppo di sicurezza tramite console

1. Apri la console Amazon VPC all'indirizzo https://console.aws.amazon.com/vpc/.
2. Fare clic su Security Groups (Gruppi di sicurezza) nel pannello di navigazione.
3. Scegliere Create Security Group (Crea gruppo di sicurezza).
4. Immettere un nome e una descrizione per il gruppo di sicurezza. Non è possibile modificare il nome e la descrizione di un gruppo di sicurezza dopo averlo creato.
5. Da VPC, seleziona il VPC.
6. (Facoltativo) Per impostazione predefinita, i nuovi gruppi di sicurezza iniziano solo con una regola in uscita che consente a tutto il traffico di uscire dalla risorsa. Devi aggiungere le regole per autorizzare qualsiasi tipo di traffico in entrata o per limitare quello in uscita.

   AWS Batch le istanze di container non richiedono l'apertura di alcuna porta in entrata. Tuttavia, potresti voler aggiungere una regola SSH. In questo modo, puoi accedere all'istanza del contenitore ed esaminare i contenitori nei job con i comandi Docker. Se desideri che l'istanza del contenitore ospiti un processo che esegue un server Web, puoi anche aggiungere regole per HTTP. Completa le fasi seguenti per aggiungere queste regole facoltative per i gruppi di sicurezza.

   Nella scheda Inbound (In entrata) crea le regole seguenti e scegli Create (Crea):

   - Selezionare Add Rule (Aggiungi regola). Per Type (Tipo) scegli HTTP. Per Source (Origine) scegli Anywhere (Ovunque) (`0.0.0.0/0`).
   - Selezionare Add Rule (Aggiungi regola). Per Type (Tipo) scegli SSH. Per Source, scegli IP personalizzato e specifica l'indirizzo IP pubblico del tuo computer o della tua rete nella notazione CIDR (Classless Inter-Domain Routing). Se l'azienda alloca gli indirizzi da un intervallo, specificare l'intero intervallo, ad esempio `203.0.113.0/24`. Per specificare un indirizzo IP individuale nella notazione CIDR, scegli Il mio IP. Questo aggiunge il prefisso di routing `/32` all'indirizzo IP pubblico.

> **Note**
>
> Per motivi di sicurezza, sconsigliamo di consentire l'accesso SSH da tutti gli indirizzi IP (`0.0.0.0/0`) all'istanza, ma solo a scopo di test e solo per un breve periodo.

7. È possibile aggiungere tag a questo punto oppure in un secondo momento. Per aggiungere un tag, sceglie Aggiungi tag, quindi specifica la chiave e il valore del tag.
8. Scegliere Create Security Group (Crea gruppo di sicurezza).

Per creare un gruppo di sicurezza utilizzando la riga di comando, vedi > create-security-group ()AWS CLI

Per ulteriori informazioni sui gruppi di sicurezza, consulta Lavorare con i gruppi di sicurezza.

Installa il AWS CLI

Per utilizzarlo AWS CLI con AWS Batch, 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.

Guida introduttiva ai AWS Batch tutorial

È possibile utilizzare la procedura guidata AWS Batch di prima esecuzione per iniziare rapidamente. AWS Batch Dopo aver completato i prerequisiti, è possibile utilizzare la procedura guidata di prima esecuzione per creare un ambiente di calcolo, una definizione di processo e una coda di lavoro.

È inoltre possibile inviare un esempio di job «Hello World» utilizzando la procedura guidata di AWS Batch prima esecuzione per testare la configurazione. Se hai già un'immagine Docker che vuoi lanciare AWS Batch, puoi usare quell'immagine per creare una definizione di lavoro.

Successivamente, puoi utilizzare la procedura guidata di AWS Batch prima esecuzione per creare un ambiente di calcolo, una coda di lavoro e inviare un esempio di lavoro Hello World.

Guida introduttiva all' EC2 orchestrazione di Amazon con la procedura guidata

Amazon Elastic Compute Cloud (Amazon EC2) fornisce capacità di elaborazione scalabile in. Cloud AWS L'utilizzo di Amazon EC2 elimina la necessità di investire in hardware in anticipo, in modo da poter sviluppare e distribuire applicazioni più velocemente.

Puoi usare Amazon EC2 per avviare tutti o pochi server virtuali di cui hai bisogno, configurare sicurezza e rete e gestire lo storage. Amazon ti EC2 consente di scalare verso l'alto o verso il basso per gestire i cambiamenti nei requisiti o i picchi di popolarità, riducendo la necessità di prevedere il traffico.

Panoramica

Questo tutorial dimostra come eseguire la configurazione AWS Batch con la procedura guidata per configurare Amazon EC2 ed eseguire. `Hello World`

Destinatari

Questo tutorial è progettato per gli amministratori di sistema e gli sviluppatori responsabili della configurazione, del test e della distribuzione AWS Batch.

Funzionalità utilizzate

Questo tutorial mostra come utilizzare la procedura guidata della AWS Batch console per:

- Crea e configura un ambiente di EC2 calcolo Amazon
- Crea una coda di lavoro.
- Creazione di una definizione di processo
- Crea e invia un lavoro da eseguire
- Visualizza l'output del lavoro in CloudWatch

Tempo richiesto

Dovrebbero essere necessari circa 10-15 minuti per completare questo tutorial.

Restrizioni regionali

Non esistono restrizioni nazionali o regionali associate all'utilizzo di questa soluzione.

Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, l'implementazione di questa soluzione potrebbe comportare l'addebito di alcuni o tutti i costi elencati nella seguente tabella.

Descrizione	Costo (dollari USA)
EC2 Istanza Amazon	Paghi per ogni EC2 istanza Amazon creata. Per ulteriori informazioni sui prezzi, consulta la pagina EC2 dei prezzi di Amazon.

Prerequisiti

Prima di iniziare:

- Creane uno Account AWS se non ne hai uno.
- Crea il ruolo `ecsInstanceRole` Instance.

Fase 1: Creare un ambiente di elaborazione

> **Important**
>
> Per iniziare nel modo più semplice e veloce possibile, questo tutorial include passaggi con le impostazioni predefinite. Prima di creare per l'uso in produzione, ti consigliamo di acquisire

familiarità con tutte le impostazioni e di implementare le impostazioni che soddisfano i tuoi requisiti.

Per creare un ambiente di calcolo per un' EC2 orchestrazione Amazon, procedi come segue:

1. Apri la procedura guidata per la prima esecuzione della AWS Batch console.
2. Per il tipo di processo e orchestrazione di Configure, scegli Amazon Elastic Compute Cloud (Amazon). EC2
3. Scegli Next (Successivo).
4. Nella sezione Configurazione dell'ambiente di calcolo per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
5. Per il ruolo di istanza, scegli un ruolo di istanza esistente a cui siano associate le autorizzazioni IAM richieste. Questo ruolo di istanza consente alle istanze del contenitore Amazon ECS nel tuo ambiente di calcolo di effettuare chiamate alle operazioni API richieste AWS . Per ulteriori informazioni, consulta Ruolo dell'istanza Amazon ECS.

   Il nome predefinito del ruolo Instance è. `ecsInstanceRole`
6. Per la configurazione di esempio, puoi lasciare le impostazioni predefinite.
7. Per la configurazione di rete, usa il tuo VPC predefinito per. Regione AWS
8. Scegli Next (Successivo).

Fase 2: Creare una coda di lavoro

Una coda di lavoro memorizza i lavori inviati fino a quando lo AWS Batch Scheduler non esegue il lavoro su una risorsa del tuo ambiente di calcolo. Per ulteriori informazioni, consulta Job queues

Per creare una coda di lavoro per un' EC2 orchestrazione Amazon, procedi come segue:

1. Per la configurazione Job queue per Name, specifica un nome univoco per la coda dei lavori. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per tutte le altre opzioni di configurazione puoi lasciare il valore predefinito.
3. Scegli Next (Successivo).

Fase 3: Creare una definizione di lavoro

AWS Batch le definizioni dei processi specificano come devono essere eseguiti i lavori. Anche se ogni processo deve fare riferimento a una definizione di processo, molti dei parametri specificati nella definizione del processo possono essere sovrascritti in fase di esecuzione.

Per creare la definizione del processo:

1. Per creare una definizione di lavoro
    a. per Nome, specifica un nome univoco per la coda dei lavori. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
    b. Per Command: facoltativo, puoi passare `hello world` a un messaggio personalizzato o lasciarlo così com'è.
2. Per tutte le altre opzioni di configurazione puoi lasciare il valore predefinito.
3. Scegli Next (Successivo).

Fase 4: Creare un lavoro

Per creare un lavoro, procedi come segue:

1. Nella sezione Configurazione del lavoro per Nome, specificare un nome univoco per il lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per tutte le altre opzioni di configurazione puoi lasciare il valore predefinito.
3. Scegli Next (Successivo).

Passaggio 5: revisione e creazione

Nella pagina Rivedi e crea, esamina i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea risorse.

1. Per Rivedi e crea scegli Crea risorse.
2. AWS Batch All'inizio si apre una finestra per allocare le risorse. Una volta completato, scegli Vai alla dashboard. Nella dashboard dovresti vedere tutte le risorse allocate e che il lavoro si

trova `Runnable` nello stato in cui ti trovi. L'esecuzione del processo è programmata e dovrebbe essere completata in 2-3 minuti.

Fase 6: Visualizza l'output del job

Per visualizzare l'output del Job, procedi come segue:

1. Nel riquadro di navigazione scegli Jobs.
2. Nel menu a discesa Job queue scegli la coda Job che hai creato per il tutorial.
3. La tabella Jobs elenca tutti i tuoi job e il loro stato attuale. Una volta completato lo stato del Job, scegli il Nome del Job per visualizzare i dettagli del Job.
4. Nel riquadro Dettagli scegli Log stream name. La CloudWatch console del Job si aprirà e dovrebbe esserci un evento con il messaggio di `hello world` o il messaggio personalizzato.

Fase 7: Pulisci le risorse del tutorial

Ti viene addebitato il costo dell' EC2 istanza Amazon mentre è abilitata. Puoi eliminare l'istanza per evitare di incorrere in addebiti.

Per eliminare le risorse che hai creato, procedi come segue:

1. Nel riquadro di navigazione scegli Job queue.
2. Nella tabella Job queue scegli la coda Job che hai creato per il tutorial.
3. Scegliere Disabilita. Una volta che lo stato della coda Job è disabilitato, puoi scegliere Elimina.
4. Una volta eliminata la coda Job, nel pannello di navigazione scegli Ambienti di calcolo.
5. Scegli l'ambiente di calcolo che hai creato per questo tutorial, quindi scegli Disabilita. Potrebbero essere necessari 1-2 minuti per completare la disattivazione dell'ambiente di calcolo.
6. Una volta che lo stato dell'ambiente di calcolo è disabilitato, scegli Elimina. Potrebbero essere necessari 1-2 minuti per eliminare l'ambiente di calcolo.

Risorse aggiuntive

Dopo aver completato il tutorial, potresti voler esplorare i seguenti argomenti:

- Esplora i componenti AWS Batch principali. Per ulteriori informazioni, consulta Componenti di AWS Batch.

- Scopri di più sui diversi ambienti di elaborazione disponibili in AWS Batch.
- Scopri di più sulle Job queues e sulle loro diverse opzioni di pianificazione.
- Scopri di più sulle definizioni di Job e sulle diverse opzioni di configurazione.
- Scopri di più sui diversi tipi di lavori.

Guida introduttiva AWS Batch e orchestrazione di Fargate tramite Wizard

AWS Fargate avvia e ridimensiona l'elaborazione per soddisfare al meglio i requisiti di risorse specificati per il contenitore. Con Fargate, non è necessario fornire troppo o pagare server aggiuntivi. Per ulteriori informazioni, vedere Fargate.

Panoramica di

Questo tutorial dimostra come configurare AWS Fargate ed eseguire la configurazione AWS Batch con la procedura guidata. `Hello World`

Destinatari

Questo tutorial è progettato per gli amministratori di sistema e gli sviluppatori responsabili della configurazione, del test e della distribuzione AWS Batch.

Funzionalità utilizzate

Questo tutorial mostra come utilizzare la procedura guidata della AWS Batch console per:

- Creare e configurare un ambiente di AWS calcolo Fargate
- Crea una coda di lavoro.
- Creazione di una definizione di processo
- Crea e invia un lavoro da eseguire
- Visualizza l'output del lavoro in CloudWatch

Tempo richiesto

Dovrebbero essere necessari circa 10-15 minuti per completare questo tutorial.

Restrizioni regionali

Non esistono restrizioni nazionali o regionali associate all'utilizzo di questa soluzione.

Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, l'implementazione di questa soluzione potrebbe comportare l'addebito di alcuni o tutti i costi elencati nella seguente tabella.

Description	Costo (dollari USA)
I prezzi si basano sulla vCPU, sulla memoria, sui sistemi operativi, sull'architettura della CPU e sulle risorse di storage richieste per il Task o il Pod.	Per ulteriori informazioni sui prezzi, consulta la pagina Prezzi di Fargate.

Prerequisiti

Prima di iniziare:

- Creane uno Account AWS se non ne hai uno.
- Crea il ruolo di esecuzione dell'attività. Se non hai ancora creato il Task Execution Role, puoi crearlo come parte di questo tutorial.

Fase 1: Creare un ambiente di calcolo

> **Important**
>
> Per iniziare nel modo più semplice e veloce possibile, questo tutorial include passaggi con le impostazioni predefinite. Prima di creare per l'uso in produzione, ti consigliamo di acquisire familiarità con tutte le impostazioni e di implementare le impostazioni che soddisfano i tuoi requisiti.

Per creare un ambiente di calcolo per un'orchestrazione Fargate, effettuate le seguenti operazioni:

1. Apri la procedura guidata per la prima esecuzione della console.AWS Batch
2. Per Configura il tipo di lavoro e orchestrazione, scegli Fargate.
3. Scegli Next (Successivo).

4. Nella sezione Configurazione dell'ambiente di calcolo per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
5. Per tutte le altre opzioni di configurazione è possibile lasciare il valore predefinito.
6. Scegli Next (Successivo).

Fase 2: Creare una coda di lavoro

Una coda di lavoro memorizza i lavori inviati fino a quando lo AWS Batch Scheduler non esegue il lavoro su una risorsa del tuo ambiente di calcolo. Per creare una coda di lavoro:

Per creare una coda di lavoro per un'orchestrazione Fargate, effettuate le seguenti operazioni:

1. Nella sezione Job queue configuration per Name, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per Priorità, inserisci 900 per la coda dei lavori.
3. Per tutte le altre opzioni di configurazione è possibile lasciare il valore predefinito.
4. Scegli Next (Successivo).

Fase 3: Creare una definizione di lavoro

Per creare la definizione del processo:

1. Nella sezione Configurazione generale:
    - Nella sezione Configurazione generale per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Nella sezione di configurazione della piattaforma Fargate:
    a. Attiva Assegna IP pubblico per assegnare un indirizzo IP pubblico. È necessario un IP pubblico per scaricare l'immagine del contenitore a meno che non sia stato configurato un archivio di immagini privato.

b. Per il ruolo di esecuzione, scegli un ruolo di esecuzione delle attività che consenta agli agenti di Amazon Elastic Container Service (Amazon ECS) di AWS effettuare chiamate per tuo conto. Scegli ecsTaskExecutionRuolo o BatchEcsTaskExecutionRole.

   Per creare il ruolo di esecuzione, scegli Crea un ruolo di esecuzione. Nella modalità Crea ruolo IAM scegli Crea ruolo IAM.

   i. La console IAM ha l'impostazione di autorizzazione già configurata per la creazione del ruolo di esecuzione.
   ii. Per il tipo di entità affidabile, verifica che AWS il servizio sia selezionato.
   iii. Per Servizio o caso utente, verifica che Elastic Container Service sia selezionato.
   iv. Scegli Next (Successivo).
   v. Per le politiche di autorizzazione, verifica che Amazon ECSTask ExecutionRolePolicy sia selezionato.
   vi. Scegli Next (Successivo).
   vii. Per Nome, revisione e creazione, verifica che il nome del ruolo sia BatchEcsTaskExecutionRole.
   viii. Scegli Crea ruolo.
   ix. Nella AWS Batch console, scegli il pulsante di aggiornamento accanto al ruolo di esecuzione. Scegli il ruolo di BatchEcsTaskExecutionRoleesecuzione.

3. Nella sezione Configurazione del contenitore:

   • Per Command, puoi passare `hello world` a un messaggio personalizzato o lasciarlo così com'è.

4. Per tutte le altre opzioni di configurazione puoi lasciare il valore predefinito.
5. Scegli Next (Successivo).

Fase 4: Creare un lavoro

Per creare un lavoro Fargate, effettuate le seguenti operazioni:

1. Nella sezione Configurazione del lavoro per Nome, specificare un nome univoco per il lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per tutte le altre opzioni di configurazione è possibile lasciare il valore predefinito.

3. Scegli Next (Successivo).

Passaggio 5: revisione e creazione

Nella pagina Rivedi e crea, esamina i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea risorse.

Passaggio 6: Visualizza l'output del lavoro

Per visualizzare l'output del Job, procedi come segue:

1. Nel riquadro di navigazione scegli Jobs.
2. Nel menu a discesa Job queue scegli la coda Job che hai creato per il tutorial.
3. La tabella Jobs elenca tutti i tuoi job e il loro stato attuale. Una volta completato lo stato del Job, scegli il Nome del Job per visualizzare i dettagli del Job.
4. Nel riquadro Dettagli scegli Log stream name. La CloudWatch console per il Job si aprirà e dovrebbe esserci un evento con il messaggio Message of `hello world` o il tuo messaggio personalizzato.

Fase 7: Pulisci le risorse del tutorial

Ti viene addebitato il costo dell' EC2 istanza Amazon mentre è abilitata. Puoi eliminare l'istanza per evitare di incorrere in addebiti.

Per eliminare le risorse che hai creato, procedi come segue:

1. Nel riquadro di navigazione scegli Job queue.
2. Nella tabella Job queue scegli la coda Job che hai creato per il tutorial.
3. Scegliere Disabilita. Una volta che lo stato della coda Job è disabilitato, puoi scegliere Elimina.
4. Una volta eliminata la coda Job, nel pannello di navigazione scegli Ambienti di calcolo.
5. Scegli l'ambiente di calcolo che hai creato per questo tutorial, quindi scegli Disabilita. Potrebbero essere necessari 1-2 minuti per completare la disattivazione dell'ambiente di calcolo.
6. Una volta che lo stato dell'ambiente di calcolo è disabilitato, scegli Elimina. Potrebbero essere necessari 1-2 minuti per eliminare l'ambiente di calcolo.

Risorse aggiuntive

Dopo aver completato il tutorial, potresti voler esplorare i seguenti argomenti:

- Scopri di più sulle best practice.
- Esplora i componenti AWS Batch principali. Per ulteriori informazioni, consulta Componenti di AWS Batch.
- Scopri di più sui diversi ambienti di elaborazione disponibili in AWS Batch.
- Scopri di più sulle Job queues e sulle loro diverse opzioni di pianificazione.
- Scopri di più sulle definizioni di Job e sulle diverse opzioni di configurazione.
- Scopri di più sui diversi tipi di lavori.

Guida introduttiva AWS Batch e Fargate all'utilizzo di AWS CLI

Questo tutorial dimostra come configurare AWS Batch l' AWS Fargate orchestrazione ed eseguire un semplice job «Hello World» utilizzando il comando (). AWS Command Line Interface AWS CLI Imparerai come creare ambienti di calcolo, code di lavoro, definizioni di lavoro e inviare lavori a. AWS Batch

Argomenti

Prerequisiti

Prima di iniziare questo tutorial, assicurati di avere quanto segue.

1. Il AWS CLI. Se devi installarla, segui la guida all'installazione della AWS CLI. Puoi anche usare AWS CloudShell, che include il AWS CLI.
2. Hai configurato il tuo AWS CLI con le credenziali appropriate. Esegui `aws configure` se non hai ancora configurato le credenziali.
3. Familiarità di base con le interfacce a riga di comando e i concetti di containerizzazione.
4. Come AWS Batch funziona con IAMper creare e gestire AWS Batch risorse, ruoli IAM e risorse VPC nel tuo. Account AWS
5. Un ID di sottorete e un ID di gruppo di sicurezza da un VPC nel tuo. Account AWS Se non disponi di un VPC, puoi crearne uno. Per ulteriori informazioni sull'utilizzo di per recuperare queste risorse IDs, vedere describe-subnets e nel Command Reference. AWS CLI describe-security-groupsAWS CLI

Tempo richiesto: circa 15-20 minuti per completare questo tutorial.

Costo: questo tutorial utilizza le risorse di calcolo Fargate. Il costo stimato per il completamento di questo tutorial è inferiore a 0,01 USD, supponendo che seguiate le istruzioni di pulizia per eliminare le risorse subito dopo il completamento. I prezzi di Fargate si basano sulle risorse di vCPU e memoria consumate, addebitati al secondo con un minimo di 1 minuto. Per informazioni aggiornate sui prezzi, consulta la sezione prezzi.AWS Fargate

Crea un ruolo di esecuzione IAM

AWS Batch richiede un ruolo di esecuzione che consenta agli agenti di Amazon Elastic Container Service (Amazon ECS) di AWS effettuare chiamate API per tuo conto. Questo ruolo è necessario per le attività di Fargate per estrarre le immagini dei container e scrivere i log su Amazon. CloudWatch

Crea un documento relativo alla politica di fiducia

Innanzitutto, crea una politica di fiducia che consenta al servizio Amazon ECS Tasks di assumere il ruolo.

```
cat > batch-execution-role-trust-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
```

```
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
EOF
```

Crea il ruolo di esecuzione

Il comando seguente crea un ruolo IAM denominato `BatchEcsTaskExecutionRoleTutorial` utilizzando la policy di fiducia appena creata.

```
aws iam create-role \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --assume-role-policy-document file://batch-execution-role-trust-policy.json
```

Allega la policy richiesta

Allega la policy AWS gestita che fornisce le autorizzazioni necessarie per l'esecuzione delle attività di Amazon ECS.

```
aws iam attach-role-policy \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

Il ruolo è ora pronto per essere utilizzato AWS Batch per l'esecuzione delle attività di Fargate.

Crea un ambiente di elaborazione

Un ambiente di elaborazione definisce le risorse di elaborazione su cui verranno eseguiti i processi in batch. In questo tutorial, creerai un ambiente di calcolo Fargate gestito che fornisce e ridimensiona automaticamente le risorse in base ai requisiti del lavoro.

Crea l'ambiente di elaborazione

Il comando seguente crea un ambiente di calcolo Fargate. Sostituisci la sottorete e il gruppo di sicurezza di esempio IDs con i tuoi per. Prerequisiti

```
aws batch create-compute-environment \
    --compute-environment-name my-fargate-compute-env \
```

```
    --type MANAGED \
    --state ENABLED \
    --compute-resources type=FARGATE,maxvCpus=128,subnets=subnet-
a123456b,securityGroupIds=sg-a12b3456
```

Di seguito viene illustrato l'aspetto dell'output quando il comando viene eseguito correttamente.

```
{
    "computeEnvironmentName": "my-fargate-compute-env",
    "computeEnvironmentArn": "arn:aws:batch:us-west-2:123456789012:compute-environment/
my-fargate-compute-env"
}
```

Attendi che l'ambiente di calcolo sia pronto

Controlla lo stato del tuo ambiente di elaborazione per assicurarti che sia pronto prima di procedere.

```
aws batch describe-compute-environments \
    --compute-environments my-fargate-compute-env \
    --query 'computeEnvironments[0].status'
```

```
"VALID"
```

Quando lo stato viene visualizzato`VALID`, l'ambiente di elaborazione è pronto per accettare lavori.

Crea una coda di lavoro

Una coda di lavoro memorizza i lavori inviati fino a quando lo AWS Batch scheduler non li esegue sulle risorse del tuo ambiente di elaborazione. I lavori vengono elaborati in ordine di priorità all'interno della coda.

Crea la coda dei lavori

Il comando seguente crea una coda di lavoro con priorità 900 che utilizza l'ambiente di calcolo Fargate.

```
aws batch create-job-queue \
    --job-queue-name my-fargate-job-queue \
    --state ENABLED \
    --priority 900 \
```

```
    --compute-environment-order order=1,computeEnvironment=my-fargate-compute-env
```

Di seguito viene illustrato l'aspetto dell'output quando il comando viene eseguito correttamente.

```
{
    "jobQueueName": "my-fargate-job-queue",
    "jobQueueArn": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-
queue"
}
```

Verifica che la coda dei lavori sia pronta

Verifica che la tua coda di lavoro sia nello `ENABLED` stato attuale e sia pronta per accettare offerte di lavoro.

```
aws batch describe-job-queues \
    --job-queues my-fargate-job-queue \
    --query 'jobQueues[0].state' "ENABLED"
```

Creazione di una definizione di processo

Una definizione di processo specifica come devono essere eseguiti i lavori, inclusa l'immagine Docker da utilizzare, i requisiti di risorse e altri parametri. Per Fargate, utilizzerai i requisiti di risorse anziché i tradizionali parametri di vCPU e memoria.

Crea la definizione del lavoro

Il comando seguente crea una definizione di processo che esegue un semplice comando «hello world» utilizzando l'immagine del contenitore busybox. `123456789012`Sostituiscilo con il tuo Account AWS ID attuale e sostituisci l'esempio Regione AWS con il tuo.

```
aws batch register-job-definition \
    --job-definition-name my-fargate-job-def \
    --type container \
    --platform-capabilities FARGATE \
    --container-properties '{
        "image": "busybox",
        "resourceRequirements": [
            {"type": "VCPU", "value": "0.25"},
            {"type": "MEMORY", "value": "512"}
        ],
```

```
        "command": ["echo", "hello world"],
        "networkConfiguration": {
            "assignPublicIp": "ENABLED"
        },
        "executionRoleArn": "arn:aws:iam::123456789012:role/
BatchEcsTaskExecutionRoleTutorial"
    },
{
    "jobDefinitionName": "my-fargate-job-def",
    "jobDefinitionArn": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1",
    "revision": 1
}'
```

La definizione del processo specifica 0,25 vCPU e 512 MB di memoria, che sono le risorse minime per un'attività Fargate. L'`assignPublicIp`impostazione è abilitata in modo che il contenitore possa estrarre l'immagine busybox da Docker Hub.

Invia e monitora un lavoro

Ora che hai tutti i componenti necessari, puoi inviare un lavoro alla tua coda e monitorarne l'avanzamento.

Invia un lavoro

Il comando seguente invia un lavoro alla coda utilizzando la definizione di processo creata.

```
aws batch submit-job \
    --job-name my-hello-world-job \
    --job-queue my-fargate-job-queue \
    --job-definition my-fargate-job-def
```

Di seguito viene illustrato l'aspetto dell'output quando il comando viene eseguito correttamente.

```
{
    "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
    "jobName": "my-hello-world-job",
    "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a"
}
```

Prendi nota del risultato `jobId` restituito nella risposta, poiché lo utilizzerai per monitorare l'avanzamento del lavoro.

Monitora lo stato del lavoro

Usa l'ID del lavoro per controllare lo stato del tuo lavoro. Il lavoro procederà attraverso diversi stati: `SUBMITTED`,`PENDING`,`RUNNABLE`,`STARTING`,`RUNNING`, e infine `SUCCEEDED` o`FAILED`.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Di seguito viene illustrato l'aspetto dell'output quando il comando viene eseguito correttamente.

```
{
    "jobs": [
        {
            "jobArn": "arn:aws:batch:us-west-2:123456789012:job/my-hello-world-job",
            "jobName": "my-hello-world-job",
            "jobId": "1509xmpl-4224-4da6-9ba9-1d1acc96431a",
            "jobQueue": "arn:aws:batch:us-west-2:123456789012:job-queue/my-fargate-job-
queue",
            "status": "SUCCEEDED",
            "createdAt": 1705161908000,
            "jobDefinition": "arn:aws:batch:us-west-2:123456789012:job-definition/my-
fargate-job-def:1"
        }
    ]
}
```

Quando lo stato viene visualizzato`SUCCEEDED`, il processo è stato completato con successo.

Visualizza i risultati del lavoro

Una volta completato il processo, puoi visualizzarne l'output in Amazon CloudWatch Logs.

Ottieni il nome del flusso di log

Innanzitutto, recupera il nome del flusso di registro dai dettagli del lavoro. Sostituisci l'ID del lavoro di esempio con il tuo.

```
aws batch describe-jobs --jobs 1509xmpl-4224-4da6-9ba9-1d1acc96431a \
    --query 'jobs[0].attempts[0].containers[0].logStreamName' \
    --output text
```

```
my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a
```

Visualizza i registri dei lavori

Utilizzate il nome del flusso di log per recuperare l'output del lavoro da CloudWatch Logs.

```
aws logs get-log-events \
    --log-group-name /aws/batch/job \
    --log-stream-name my-fargate-job-def/default/1509xmpl-4224-4da6-9ba9-1d1acc96431a \
    --query 'events[*].message' \
    --output text
```

L'output mostra «hello world», a conferma che il processo è stato eseguito correttamente.

Eseguire la pulizia delle risorse

Per evitare addebiti continui, ripulisci le risorse che hai creato in questo tutorial. È necessario eliminare le risorse nell'ordine corretto a causa delle dipendenze.

Disabilita ed elimina la coda dei lavori

Innanzitutto, disabilita la coda dei lavori, quindi eliminala.

```
aws batch update-job-queue \
    --job-queue my-fargate-job-queue \
    --state DISABLED
```

```
aws batch delete-job-queue \
    --job-queue my-fargate-job-queue
```

Disabilita ed elimina l'ambiente di calcolo

Dopo aver eliminato la coda dei lavori, disabilita ed elimina l'ambiente di calcolo.

```
aws batch update-compute-environment \
    --compute-environment my-fargate-compute-env \
    --state DISABLED
```

```
aws batch delete-compute-environment \
    --compute-environment my-fargate-compute-env
```

Pulisci il ruolo IAM

Rimuovi l'allegato alla policy ed elimina il ruolo IAM.

```
aws iam detach-role-policy \
    --role-name BatchEcsTaskExecutionRoleTutorial \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonECSTaskExecutionRolePolicy
```

```
aws iam delete-role \
    --role-name BatchEcsTaskExecutionRoleTutorial
```

Rimuovi i file temporanei

Elimina il file dei criteri di attendibilità che hai creato.

```
rm batch-execution-role-trust-policy.json
```

Tutte le risorse sono state pulite con successo.

Passa alla produzione

Questo tutorial è stato progettato per aiutarti a capire come AWS Batch funziona con Fargate. Per le implementazioni di produzione, considera i seguenti requisiti aggiuntivi:

Considerazioni sulla sicurezza:

- Crea gruppi di sicurezza dedicati con un accesso minimo richiesto invece di utilizzare gruppi di sicurezza predefiniti
- Utilizza sottoreti private con NAT Gateway anziché l'assegnazione di IP pubblici per i contenitori
- Archivia le immagini dei container in Amazon ECR anziché utilizzare repository pubblici
- Implementa gli endpoint VPC per la comunicazione di AWS servizio per evitare il traffico Internet

Considerazioni sull'architettura:

- Implementa su più zone di disponibilità per un'elevata disponibilità
- Implementa strategie di riprova di lavoro e code con lettere morte per la gestione degli errori
- Utilizza più code di lavoro con priorità diverse per la gestione del carico di lavoro
- Configura le politiche di scalabilità automatica in base alla profondità della coda e all'utilizzo delle risorse

- Implementa il monitoraggio e la generazione di avvisi in caso di errori di lavoro e utilizzo delle risorse

Considerazioni operative:

- Configura CloudWatch dashboard e allarmi per il monitoraggio
- Implementa la registrazione e gli audit trail corretti
- Usa CloudFormation or the AWS CDK for infrastructure come codice
- Stabilisci procedure di backup e disaster recovery

Per una guida completa sulle architetture pronte per la produzione, consulta il AWS Well-Architected Framework and Security Best Practices.AWS

Fasi successive

Ora che hai completato questo tutorial, puoi esplorare funzionalità più avanzate: AWS Batch

- Job queues— Scopri la pianificazione delle code di lavoro e la gestione delle priorità
- Definizioni del lavoro— Esplora configurazioni avanzate di definizione dei processi, tra cui variabili di ambiente, volumi e strategie di riprova
- Ambienti di calcolo per AWS Batch— Comprendi i diversi tipi di ambiente di calcolo e le diverse opzioni di scalabilità
- Lavori paralleli multinodo— Esegui processi che si estendono su più nodi di elaborazione
- Lavori di matrice— Invia un gran numero di lavori simili in modo efficiente
- Le migliori pratiche per AWS Batch— Impara le tecniche di ottimizzazione per i carichi di lavoro di produzione

Guida introduttiva ad AWS Batch Amazon EKS

AWS Batch su Amazon EKS è un servizio gestito per la pianificazione e la scalabilità dei carichi di lavoro in batch in cluster Amazon EKS esistenti. AWS Batch non crea, amministra o esegue operazioni sul ciclo di vita dei tuoi cluster Amazon EKS per tuo conto. AWS Batch l'orchestrazione ridimensiona verso l'alto e verso il basso i nodi gestiti da AWS Batch ed esegue i pod su tali nodi.

AWS Batch non tocca nodi, gruppi di nodi con scalabilità automatica o cicli di vita dei pod che non sono associati agli ambienti di AWS Batch elaborazione all'interno del cluster Amazon EKS. AWS

Batch Per funzionare in modo efficace, il suo ruolo collegato ai servizi necessita di autorizzazioni di controllo degli accessi Kubernetes in base al ruolo (RBAC) nel cluster Amazon EKS esistente. Per ulteriori informazioni, consulta Using RBAC Authorization nella documentazione. Kubernetes

AWS Batch richiede uno spazio Kubernetes dei nomi in cui possa definire i pod come job. AWS Batch Consigliamo uno spazio dei nomi dedicato per isolare i AWS Batch pod dagli altri carichi di lavoro del cluster.

Dopo AWS Batch aver ottenuto l'accesso RBAC e aver stabilito uno spazio dei nomi, puoi associare il cluster Amazon EKS a un ambiente di AWS Batch calcolo utilizzando l'operazione API. CreateComputeEnvironment Una coda di lavoro può essere associata a questo nuovo ambiente di calcolo Amazon EKS. AWS Batch i lavori vengono inviati alla coda dei lavori in base a una definizione di processo Amazon EKS utilizzando l'operazione SubmitJobAPI. AWS Batch quindi avvia i nodi AWS Batch gestiti e inserisce i lavori dalla coda dei lavori come Kubernetes pod nel cluster EKS associato a un ambiente di elaborazione. AWS Batch

Le seguenti sezioni spiegano come eseguire la configurazione AWS Batch su Amazon EKS.

Indice

Panoramica

Questo tutorial dimostra come eseguire la configurazione AWS Batch con Amazon EKS utilizzando AWS CLI, `kubectl` `eeksctl`.

Destinatari

Questo tutorial è progettato per gli amministratori di sistema e gli sviluppatori responsabili della configurazione, del test e della distribuzione AWS Batch.

Funzionalità utilizzate

Questo tutorial mostra come usare AWS CLI, per:

- Crea e configura un ambiente di calcolo Amazon EKS
- Crea una coda di lavoro.
- Creazione di una definizione di processo
- Crea e invia un lavoro da eseguire
- Invia un lavoro con eccezioni

Tempo richiesto

Il completamento di questo tutorial dovrebbe richiedere circa 30-40 minuti.

Restrizioni regionali

Non esistono restrizioni nazionali o regionali associate all'utilizzo di questa soluzione.

Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, l'implementazione di questa soluzione potrebbe comportare l'addebito di alcuni o tutti i costi elencati nella seguente tabella.

Descrizione	Costo (dollari USA)
L'addebito viene calcolato in base all'ora del cluster	Varia a seconda dell'istanza, consulta i prezzi di Amazon EKS

Prerequisiti

Prima di iniziare questo tutorial, devi installare e configurare i seguenti strumenti e risorse necessari per creare e gestire sia AWS Batch le risorse Amazon EKS che quelle di Amazon EKS.

- AWS CLI: uno strumento a riga di comando per usare i servizi AWS , tra cui Amazon EKS. Questa guida richiede l'utilizzo della versione 2.8.6 o successiva o 1.26.0 o successiva. Per

ulteriori informazioni, vedere Installazione, aggiornamento e disinstallazione di nella Guida per l' AWS CLIutente.AWS Command Line Interface Dopo aver installato AWS CLI, ti consigliamo di configurarlo anche. Per ulteriori informazioni, vedere Configurazione rapida con `aws configure` nella Guida AWS Command Line Interface per l'utente.

- **`kubectl`**: uno strumento a riga di comando per lavorare con i cluster Kubernetes. Questa guida richiede l'utilizzo della versione `1.23` o successiva. Per ulteriori informazioni, consulta la pagina Installing or updating `kubectl` nella Guida per l'utente di Amazon EKS.
- **`eksctl`**— Uno strumento da riga di comando per lavorare con i cluster Amazon EKS che automatizza molte attività individuali. Questa guida richiede l'utilizzo della versione `0.115.0` o successiva. Per ulteriori informazioni, consulta la pagina Installing or updating `eksctl` nella Guida per l'utente di Amazon EKS.
- Autorizzazioni IAM richieste: il responsabile della sicurezza IAM che stai utilizzando deve disporre delle autorizzazioni per lavorare con i ruoli IAM di Amazon EKS e i ruoli collegati ai servizi CloudFormation, oltre a un VPC e risorse correlate. Per ulteriori informazioni, consulta Azioni, risorse e chiavi di condizione per Amazon Elastic Kubernetes Service e Using service-linked roles nella IAM User Guide. È necessario che tutti i passaggi di questa guida siano completati dallo stesso utente.
- Autorizzazioni: gli utenti che chiamano l'operazione CreateComputeEnvironmentAPI per creare un ambiente di calcolo che utilizza risorse Amazon EKS richiedono le autorizzazioni per il funzionamento dell'`eks:DescribeCluster`API.
- Account AWS numero: devi conoscere il tuo ID. Account AWS Segui le istruzioni riportate in Visualizzazione del tuo Account AWS ID.
- (Facoltativo) CloudWatch: per esaminare i dettagli di (Facoltativo) Invia un lavoro con modifiche, è necessario configurare la registrazione. Per ulteriori informazioni, consulta Usa CloudWatch Logs per monitorare i lavori AWS Batch su Amazon EKS.

Fase 1: crea il tuo cluster Amazon EKS per AWS Batch

> **Important**
>
> Per iniziare nel modo più semplice e veloce possibile, questo tutorial include passaggi con impostazioni predefinite. Prima di creare per l'uso in produzione, ti consigliamo di acquisire familiarità con tutte le impostazioni e di implementare le impostazioni che soddisfano i tuoi requisiti.

Dopo aver installato i prerequisiti, è necessario creare il cluster utilizzando. `eksctl` La creazione del cluster può richiedere dai 10 ai 15 minuti.

```
$  eksctl create cluster --name my-cluster-name --region region-code
```

Nel comando precedente sostituisci:

- Sostituisci *`my-cluster-name`* con il nome che desideri utilizzare per il tuo cluster.
- Sostituisci *`region-code`* con il Regione AWS per creare il cluster, ad esempio`us-west-2`.

Il nome e la regione del cluster sono necessari per la parte successiva di questo tutorial.

Fase 2: prepara il cluster Amazon EKS per AWS Batch

Tutti i passaggi sono obbligatori.

1. Crea un namespace dedicato per i lavori AWS Batch

   Utilizzare `kubectl` per creare un nuovo spazio dei nomi.

   ```
   $ namespace=my-aws-batch-namespace
   ```

   ```
   $ cat - <<EOF | kubectl create -f -
   {
     "apiVersion": "v1",
     "kind": "Namespace",
     "metadata": {
       "name": "${namespace}",
       "labels": {
         "name": "${namespace}"
       }
     }
   }
   EOF
   ```

   Output:

   ```
   namespace/my-aws-batch-namespace created
   ```

2. Abilita l'accesso tramite il controllo degli accessi basato sui ruoli (RBAC)

   `kubectl`Utilizzalo per creare un Kubernetes ruolo per il cluster che AWS Batch consenta di controllare nodi e pod e di associare il ruolo. È necessario eseguire questa operazione una volta per ogni cluster EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
```

```
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

3. Crea un Kubernetes ruolo con ambito namespace per la gestione e il ciclo AWS Batch di vita dei pod e associalo. È necessario eseguire questa operazione una volta per ogni namespace univoco.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
```

```
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

4. Aggiorna la mappa Kubernetes `aws-auth` di configurazione per mappare le autorizzazioni RBAC precedenti al ruolo collegato al servizio. AWS Batch

   Nel comando seguente sostituisci:

   - Sostituisci *<your-account-number>* con il tuo Account AWS numero.

   ```
   $ eksctl create iamidentitymapping \
       --cluster my-cluster-name \
       --arn "arn:aws:iam::<your-account-number>:role/AWSServiceRoleForBatch" \
       --username aws-batch
   ```

   Output:

   ```
   2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account-number>:role/
   AWSServiceRoleForBatch" to auth ConfigMap
   ```

   > **Note**
   >
   > Il percorso `aws-service-role/batch.amazonaws.com/` è stato rimosso dall'ARN del ruolo collegato al servizio. Ciò è dovuto a un problema con la `aws-auth` mappa di configurazione. Per ulteriori informazioni, consulta Ruoli con percorsi non funzionano quando il percorso è incluso nel relativo ARN in. aws-auth configmap

Fase 3: creare un ambiente di calcolo Amazon EKS

AWS Batch gli ambienti di calcolo definiscono i parametri delle risorse di calcolo per soddisfare le esigenze di carico di lavoro in batch. In un ambiente di elaborazione gestito, ti AWS Batch aiuta a gestire la capacità e i tipi di istanze delle risorse di calcolo (Kubernetesnodi) all'interno del tuo cluster Amazon EKS. Si basa sulla specifica delle risorse di calcolo che definisci al momento della creazione dell'ambiente di calcolo. È possibile utilizzare istanze EC2 On-Demand o Istanze Spot. EC2

Ora che il ruolo AWSServiceRoleForBatchcollegato al servizio ha accesso al tuo cluster Amazon EKS, puoi creare AWS Batch risorse. Innanzitutto, crea un ambiente di elaborazione che punti al tuo cluster Amazon EKS.

- `eksctl get cluster` *`my-cluster-name`*Da `subnets` eseguire per ottenere le sottoreti utilizzate dal cluster.
- Per il `securityGroupIds` parametro puoi utilizzare lo stesso gruppo di sicurezza del cluster Amazon EKS. Questo comando recupera l'ID del gruppo di sicurezza per il cluster.

  ```
  $ aws eks describe-cluster \
      --name my-cluster-name \
      --query cluster.resourcesVpcConfig.clusterSecurityGroupId
  ```

- `instanceRole`viene creato quando si crea il cluster. Per trovare l'`instanceRole`elenco di tutte le entità che utilizzano la `AmazonEKSWorkerNodePolicy` policy:

  ```
  $  aws iam list-entities-for-policy --policy-arn arn:aws:iam::aws:policy/
  AmazonEKSWorkerNodePolicy
  ```

  Il nome del ruolo politico contiene il nome del cluster creato`eksctl-`*`my-cluster-name`*`-nodegroup-example`.

  Per trovare l'`instanceRole`arn esegui il seguente comando:

  ```
  $  aws iam list-instance-profiles-for-role --role-name eksctl-my-cluster-name-
  nodegroup-example
  ```

  Output:

  ```
  INSTANCEPROFILES        arn:aws:iam::<your-account-number>:instance-profile/
  eks-04cb2200-94b9-c297-8dbe-87f12example
  ```

Per ulteriori informazioni, consulta Creazione del ruolo IAM del nodo Amazon EKS e Abilitazione dell'accesso principale IAM al cluster nella Guida per l'utente di Amazon EKS. Se utilizzi il pod networking, consulta Configurazione del plug-in Amazon VPC CNI per l'utilizzo dei ruoli IAM Kubernetes per gli account di servizio nella Amazon EKS User Guide.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:region-code:your-account-number:cluster/my-cluster-
name",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-
environment.json
```

Note

- La manutenzione di un ambiente di calcolo Amazon EKS è una responsabilità condivisa. Per ulteriori informazioni, consulta Responsabilità condivisa dei nodi Kubernetes.

Fase 4: creare una coda di lavoro e collegare l'ambiente di calcolo

> **Important**
>
> È importante verificare che l'ambiente di elaborazione sia integro prima di procedere. A tale scopo è possibile utilizzare l'operazione DescribeComputeEnvironmentsAPI.
>
> ```
> $ aws batch describe-compute-environments --compute-environments My-Eks-CE1
> ```
>
> Conferma che il `status` parametro non lo sia`INVALID`. Se lo è, esamina il `statusReason` parametro relativo alla causa. Per ulteriori informazioni, consulta Risoluzione dei problemi AWS Batch.

I lavori inviati a questa nuova coda di lavoro vengono eseguiti come pod su nodi AWS Batch gestiti che si sono uniti al cluster Amazon EKS associato al tuo ambiente di elaborazione.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
```

```
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Fase 5: Creare una definizione di lavoro

La seguente definizione di Job indica al pod di dormire per 60 secondi.

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-
definition.json
```

Note

- Esistono alcune considerazioni relative ai parametri `cpu` and`memory`. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.

Fase 6: Inviare un lavoro

Esegui il AWS CLI comando seguente per inviare un nuovo Job.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
```

Per verificare lo stato di un Job:

```
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Note

- Per ulteriori informazioni sull'esecuzione di processi sulle risorse di Amazon EKS, consultaOfferte di lavoro Amazon EKS.

Fase 7: Visualizza l'output del job

Per visualizzare l'output del Job, procedi come segue:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione scegli Jobs.
3. Nel menu a discesa Job queue scegli la coda Job che hai creato per il tutorial.
4. La tabella Jobs elenca tutti i tuoi job e il loro stato attuale. Una volta completato lo stato del lavoro*My-Eks-JQ1*, scegli il nome del lavoro per visualizzare i dettagli del lavoro.
5. Nel riquadro Dettagli, gli orari Iniziato e Interrotto a devono trovarsi a un minuto di distanza l'uno dall'altro.

Fase 8: (Facoltativo) Inviare un lavoro con eccezioni

Questo lavoro sostituisce il comando passato al contenitore. AWS Batch pulisce in modo aggressivo i pod dopo il completamento dei lavori per ridurre il carico a. Kubernetes Per esaminare i dettagli di un

lavoro, è necessario configurare la registrazione. Per ulteriori informazioni, consulta Usa CloudWatch Logs per monitorare i lavori AWS Batch su Amazon EKS.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job --cli-input-json file://./submit-job-override.json
```

Note

- Per una migliore visibilità dei dettagli delle operazioni, abilita la registrazione del piano di controllo di Amazon EKS. Per ulteriori informazioni, consulta la registrazione del piano di controllo di Amazon EKS nella Guida per l'utente di Amazon EKS.
- Daemonsetse il kubelets sovraccarico influisce sulle risorse di vCPU e memoria disponibili, in particolare sulla scalabilità e sul posizionamento lavorativo. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.

Per visualizzare l'output del job, effettuate le seguenti operazioni:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.

2. Nel riquadro di navigazione scegli Jobs.
3. Nel menu a discesa Job queue scegli la coda Job che hai creato per il tutorial.
4. La tabella Jobs elenca tutti i tuoi job e il loro stato attuale. Una volta completato lo stato del Job, scegli il Nome del Job per visualizzare i dettagli del Job.
5. Nel riquadro Dettagli scegli Log stream name. La CloudWatch console del Job si aprirà e dovrebbe esserci un evento con il messaggio di `hello world` o il messaggio personalizzato.

Fase 9: Pulisci le risorse del tutorial

Ti viene addebitato il costo dell' EC2 istanza Amazon mentre è abilitata. Puoi eliminare l'istanza per evitare di incorrere in addebiti.

Per eliminare le risorse che hai creato, procedi come segue:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione scegli Job queue.
3. Nella tabella Job queue scegli la coda Job che hai creato per il tutorial.
4. Scegliere Disabilita. Una volta che lo stato della coda Job è disabilitato, puoi scegliere Elimina.
5. Una volta eliminata la coda Job, nel pannello di navigazione scegli Ambienti di calcolo.
6. Scegli l'ambiente di calcolo che hai creato per questo tutorial, quindi scegli Disabilita. Potrebbero essere necessari 1-2 minuti per completare la disattivazione dell'ambiente di calcolo.
7. Una volta che lo stato dell'ambiente di calcolo è disabilitato, scegli Elimina. Potrebbero essere necessari 1-2 minuti per eliminare l'ambiente di calcolo.

Risorse aggiuntive

Dopo aver completato il tutorial, potresti voler esplorare i seguenti argomenti:

- Scopri di più sulle best practice.
- Esplora i componenti AWS Batch principali. Per ulteriori informazioni, consulta Componenti di AWS Batch.
- Scopri di più sui diversi ambienti di elaborazione disponibili in AWS Batch.
- Scopri di più sulle Job queues e sulle loro diverse opzioni di pianificazione.

- Scopri di più sulle definizioni di Job e sulle diverse opzioni di configurazione.
- Scopri di più sui diversi tipi di lavori.

Guida introduttiva ad AWS Batch Amazon EKS Private Clusters

AWS Batch è un servizio gestito che orchestra carichi di lavoro in batch nei cluster Amazon Elastic Kubernetes Service (Amazon EKS). Ciò include l'accodamento, il monitoraggio delle dipendenze, la gestione delle priorità e dei nuovi tentativi di lavoro, la gestione dei pod e il ridimensionamento dei nodi. Questa funzionalità collega il tuo cluster privato esistente di Amazon EKS AWS Batch per eseguire i tuoi lavori su larga scala. Puoi utilizzare `eksctl`(un'interfaccia a riga di comando per Amazon EKS), la AWS console o AWS Command Line Interfacecreare un cluster Amazon EKS privato con tutte le altre risorse necessarie.

Per impostazione predefinita, i cluster solo privati di Amazon EKS non hanno accesso a inbound/outbound Internet e puoi accedere al server API solo dall'interno del tuo VPC o da una rete connessa. Gli endpoint Amazon VPC vengono utilizzati per consentire l'accesso privato ad altri servizi. AWS `eksctl`supporta la creazione di cluster completamente privati utilizzando un Amazon VPC e sottoreti preesistenti. `eksctl`crea inoltre endpoint Amazon VPC nell'Amazon VPC fornito e modifica le tabelle di routing per le sottoreti fornite.

A ogni sottorete deve essere associata una tabella di routing esplicita, poiché `eksctl` non modifica la tabella di routing principale. Il tuo cluster deve estrarre immagini da un registro di container che si trova nel tuo Amazon VPC. Inoltre, puoi creare un Amazon Elastic Container Registry nel tuo Amazon VPC e copiarvi le immagini dei container per i nodi da cui estrarre. Per ulteriori informazioni, consulta Copiare l'immagine di un contenitore da un repository a un altro. Per iniziare a usare gli archivi privati di Amazon ECR, consulta la sezione Archivi privati di Amazon ECR.

Facoltativamente, puoi creare una regola pull through cache con Amazon ECR. Una volta creata una regola pull through cache per un registro pubblico esterno, puoi estrarre un'immagine da quel registro pubblico esterno utilizzando il tuo URI (Uniform Resource Identifier) del registro privato Amazon ECR. Quindi Amazon ECR crea un repository e memorizza l'immagine nella cache. Quando un'immagine memorizzata nella cache viene estratta utilizzando l'URI del registro privato Amazon ECR, Amazon ECR controlla il registro remoto per verificare se è disponibile una nuova versione dell'immagine e aggiorna il registro privato fino a una volta ogni 24 ore.

Indice

Panoramica

Questo tutorial dimostra come eseguire la configurazione AWS Batch con un Amazon EKS privato utilizzando AWS CloudShell, `kubectl` `eeksctl`.

Destinatari

Questo tutorial è progettato per gli amministratori di sistema e gli sviluppatori responsabili della configurazione, del test e della distribuzione AWS Batch.

Funzionalità utilizzate

Questo tutorial mostra come usare AWS CLI, per:

- Usa Amazon Elastic Container Registry (Amazon ECR) per archiviare le immagini dei container
- Crea e configura un ambiente di calcolo Amazon EKS
- Crea una coda di lavoro.
- Creazione di una definizione di processo
- Crea e invia un lavoro da eseguire
- Invia un lavoro con eccezioni

Tempo richiesto

Dovrebbero essere necessari circa 40-50 minuti per completare questo tutorial.

Restrizioni regionali

Non esistono restrizioni nazionali o regionali associate all'utilizzo di questa soluzione.

Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, l'implementazione di questa soluzione potrebbe comportare l'addebito di alcuni o tutti i costi elencati nella seguente tabella.

Descrizione	Costo (dollari USA)
L'addebito viene calcolato in base all'ora del cluster	Varia a seconda dell'istanza, consulta i prezzi di Amazon EKS
EC2 Istanza Amazon	Paghi per ogni EC2 istanza Amazon creata. Per ulteriori informazioni sui prezzi, consulta la pagina EC2 dei prezzi di Amazon.

Prerequisiti

Questo tutorial utilizza AWS CloudShell una shell preautenticata basata su browser che si avvia direttamente da. Console di gestione AWS Ciò consente l'accesso al cluster una volta che non dispone più dell'accesso pubblico a Internet. AWS CLI`kubectl`, e `eksctl` potrebbe essere già installato come parte di AWS CloudShell. Per ulteriori informazioni su AWS CloudShell, consulta la Guida AWS CloudShell per l'utente. Un'alternativa AWS CloudShell è connettersi al VPC del cluster o a una rete connessa.

Per eseguire i comandi kubectl, è necessario l'accesso privato al cluster Amazon EKS. Ciò significa che tutto il traffico verso il server API del cluster deve provenire dal VPC del cluster o da una rete connessa.

- AWS CLI— Uno strumento da riga di comando per lavorare con AWS i servizi, incluso Amazon EKS. Questa guida richiede l'utilizzo della versione 2.8.6 o successiva o 1.26.0 o successiva. Per ulteriori informazioni, vedere Installazione, aggiornamento e disinstallazione di nella Guida per l' AWS CLIutente.AWS Command Line Interface Dopo aver installato AWS CLI, ti consigliamo di

configurarlo anche. Per ulteriori informazioni, vedere Configurazione rapida con `aws configure` nella Guida AWS Command Line Interface per l'utente.

- **`kubectl`**: uno strumento a riga di comando per lavorare con i cluster Kubernetes. Questa guida richiede l'utilizzo della versione `1.23` o successiva. Per ulteriori informazioni, consulta la pagina Installing or updating `kubectl` nella Guida per l'utente di Amazon EKS.
- **`eksctl`**— Uno strumento da riga di comando per lavorare con i cluster Amazon EKS che automatizza molte attività individuali. Questa guida richiede l'utilizzo della versione `0.115.0` o successiva. Per ulteriori informazioni, consulta la pagina Installing or updating `eksctl` nella Guida per l'utente di Amazon EKS.
- Autorizzazioni: gli utenti che chiamano l'operazione CreateComputeEnvironmentAPI per creare un ambiente di calcolo che utilizza risorse Amazon EKS richiedono le autorizzazioni per il funzionamento dell'API `eks:DescribeCluster` and`eks:ListClusters`. Puoi allegare la policy AWSBatchFullAccessgestita al tuo account utente seguendo le istruzioni Aggiungere e rimuovere le autorizzazioni di identità IAM nella Guida per l'utente IAM.
- InstanceRole— È necessario creare un account `InstanceRole` per i nodi Amazon EKS con le `AmazonEC2ContainerRegistryPullOnly` policy `AmazonEKSWorkerNodePolicy` e. Per istruzioni su come creare il ruolo IAM del nodo Amazon EKS, `InstanceRole` vedi Creazione del ruolo IAM del nodo Amazon EKS. Avrai bisogno dell'ARN di. `InstanceRole`
- Account AWS ID: devi conoscere il tuo Account AWS ID. Segui le istruzioni riportate in Visualizzazione del tuo Account AWS ID.
- (Facoltativo) CloudWatch — Per esaminare i dettagli di (Facoltativo) Invia un lavoro con modifiche, è necessario configurare la registrazione. Per ulteriori informazioni, consulta Usa CloudWatch Logs per monitorare i lavori AWS Batch su Amazon EKS.

Fase 1: Crea il tuo cluster EKS per AWS Batch

> **Important**
>
> Per iniziare nel modo più semplice e veloce possibile, questo tutorial include passaggi con impostazioni predefinite. Prima di creare per l'uso in produzione, ti consigliamo di acquisire familiarità con tutte le impostazioni e di implementare le impostazioni che soddisfano i tuoi requisiti.

Ti consigliamo di utilizzare `eksctl` il seguente file di configurazione per creare il cluster. Per configurare manualmente il cluster, segui le istruzioni in Distribuisci cluster privati con accesso limitato a Internet nella Amazon EKS User Guide.

1. Apri la AWS CloudShell console e imposta la regione su. `us-east-1` Per il resto del tutorial assicurati di utilizzarlo`us-east-1`.
2. Crea un cluster EKS privato nella `us-east-1` regione utilizzando il file di `eksctl` configurazione di esempio. Salva il file yaml nel tuo AWS CloudShell ambiente e assegnagli un nome`clusterConfig.yaml`. Puoi cambiarlo *`my-test-cluster`* con il nome che desideri usare per il tuo cluster.

   ```
   kind: ClusterConfig
   apiVersion: eksctl.io/v1alpha5
   metadata:
       name: my-test-cluster
       region: us-east-1
   availabilityZones:
       - us-east-1a
       - us-east-1b
       - us-east-1c
   managedNodeGroups:
       - name: ng-1
         privateNetworking: true
   privateCluster:
       enabled: true
       skipEndpointCreation: false
   ```

3. Crea le tue risorse usando il comando:. `eksctl create cluster -f clusterConfig.yaml` La creazione del cluster può richiedere dai 10 ai 15 minuti.
4. Una volta completata la creazione del cluster, è necessario aggiungere il proprio indirizzo AWS CloudShell IP all'elenco degli indirizzi consentiti. Per trovare il tuo indirizzo AWS CloudShell IP esegui il seguente comando:

   ```
   curl http://checkip.amazonaws.com
   ```

   Una volta ottenuto l'indirizzo IP pubblico, devi creare una regola per l'elenco degli indirizzi consentiti:

   ```
   aws eks update-cluster-config \
     --name my-test-cluster \
   ```

```
  --region us-east-1 \
  --resources-vpc-config
endpointPublicAccess=true,endpointPrivateAccess=true,publicAccessCidrs=["<Public
IP>/32"]
```

Quindi applica l'aggiornamento al file di configurazione kubectl:

```
aws eks update-kubeconfig --name my-test-cluster --region us-east-1
```

5. Per verificare di avere accesso ai nodi esegui il seguente comando:

```
kubectl get nodes
```

L'output del comando è:

```
NAME                                STATUS   ROLES    AGE     VERSION
ip-192-168-107-235.ec2.internal   Ready    none     1h   v1.32.3-eks-473151a
ip-192-168-165-40.ec2.internal    Ready    none     1h   v1.32.3-eks-473151a
ip-192-168-98-54.ec2.internal     Ready    none     1h   v1.32.1-eks-5d632ec
```

Fase 2: Preparare il cluster EKS per AWS Batch

Tutti i passaggi sono obbligatori e devono essere eseguiti in AWS CloudShell.

1. Crea un namespace dedicato per i lavori AWS Batch

   Utilizzare `kubectl` per creare un nuovo spazio dei nomi.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl create -f -
{
  "apiVersion": "v1",
  "kind": "Namespace",
  "metadata": {
    "name": "${namespace}",
    "labels": {
      "name": "${namespace}"
    }
  }
```

```
}
EOF
```

Output:

```
namespace/my-aws-batch-namespace created
```

2. Abilita l'accesso tramite il controllo degli accessi basato sui ruoli (RBAC)

   `kubectl`Utilizzalo per creare un Kubernetes ruolo per il cluster che AWS Batch consenta di controllare nodi e pod e di associare il ruolo. È necessario eseguire questa operazione una volta per ogni cluster Amazon EKS.

```
$ cat - <<EOF | kubectl apply -f -
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRole
metadata:
  name: aws-batch-cluster-role
rules:
  - apiGroups: [""]
    resources: ["namespaces"]
    verbs: ["get"]
  - apiGroups: [""]
    resources: ["nodes"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list"]
  - apiGroups: [""]
    resources: ["configmaps"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["apps"]
    resources: ["daemonsets", "deployments", "statefulsets", "replicasets"]
    verbs: ["get", "list", "watch"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["clusterroles", "clusterrolebindings"]
    verbs: ["get", "list"]
---
apiVersion: rbac.authorization.k8s.io/v1
```

```
kind: ClusterRoleBinding
metadata:
  name: aws-batch-cluster-role-binding
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: ClusterRole
  name: aws-batch-cluster-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
clusterrole.rbac.authorization.k8s.io/aws-batch-cluster-role created
clusterrolebinding.rbac.authorization.k8s.io/aws-batch-cluster-role-binding created
```

Crea un Kubernetes ruolo con ambito namespace per la gestione e il ciclo AWS Batch di vita dei pod e associalo. È necessario eseguire questa operazione una volta per ogni namespace univoco.

```
$ namespace=my-aws-batch-namespace
```

```
$ cat - <<EOF | kubectl apply -f - --namespace "${namespace}"
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: aws-batch-compute-environment-role
  namespace: ${namespace}
rules:
  - apiGroups: [""]
    resources: ["pods"]
    verbs: ["create", "get", "list", "watch", "delete", "patch"]
  - apiGroups: [""]
    resources: ["serviceaccounts"]
    verbs: ["get", "list"]
  - apiGroups: ["rbac.authorization.k8s.io"]
    resources: ["roles", "rolebindings"]
    verbs: ["get", "list"]
---
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: aws-batch-compute-environment-role-binding
  namespace: ${namespace}
subjects:
- kind: User
  name: aws-batch
  apiGroup: rbac.authorization.k8s.io
roleRef:
  kind: Role
  name: aws-batch-compute-environment-role
  apiGroup: rbac.authorization.k8s.io
EOF
```

Output:

```
role.rbac.authorization.k8s.io/aws-batch-compute-environment-role created
rolebinding.rbac.authorization.k8s.io/aws-batch-compute-environment-role-binding
 created
```

Aggiorna la mappa Kubernetes `aws-auth` di configurazione per mappare le autorizzazioni RBAC precedenti al ruolo collegato al servizio. AWS Batch

```
$ eksctl create iamidentitymapping \
    --cluster my-test-cluster \
    --arn "arn:aws:iam::<your-account-ID>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Output:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account-ID>:role/
AWSServiceRoleForBatch" to auth ConfigMap
```

> **Note**
>
> Il percorso `aws-service-role/batch.amazonaws.com/` è stato rimosso dall'ARN del ruolo collegato al servizio. Ciò è dovuto a un problema con la `aws-auth` mappa di

configurazione. Per ulteriori informazioni, consulta Ruoli con percorsi non funzionano quando il percorso è incluso nel relativo ARN in. aws-auth configmap

Fase 3: creare un ambiente di calcolo Amazon EKS

AWS Batch gli ambienti di calcolo definiscono i parametri delle risorse di calcolo per soddisfare le esigenze di carico di lavoro in batch. In un ambiente di elaborazione gestito, ti AWS Batch aiuta a gestire la capacità e i tipi di istanze delle risorse di calcolo (Kubernetesnodi) all'interno del tuo cluster Amazon EKS. Si basa sulla specifica delle risorse di calcolo che definisci al momento della creazione dell'ambiente di calcolo. È possibile utilizzare istanze EC2 On-Demand o Istanze Spot. EC2

Ora che il ruolo AWSServiceRoleForBatchcollegato al servizio ha accesso al tuo cluster Amazon EKS, puoi creare AWS Batch risorse. Innanzitutto, crea un ambiente di elaborazione che punti al tuo cluster Amazon EKS.

- `eksctl get cluster` *`my-test-cluster`*Da `subnets` eseguire per ottenere le sottoreti utilizzate dal cluster.
- Per il `securityGroupIds` parametro puoi utilizzare lo stesso gruppo di sicurezza del cluster Amazon EKS. Questo comando recupera l'ID del gruppo di sicurezza per il cluster.

```
$ aws eks describe-cluster \
    --name my-test-cluster \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

- Usa l'ARN che `instanceRole` hai creato nei Prerequisiti.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:us-east-1:<your-account-ID>:cluster/my-test-cluster",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
```

```
    "minvCpus": 0,
    "maxvCpus": 128,
    "instanceTypes": [
        "m5"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-the-image-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file://./batch-eks-compute-environment.json
```

Note

- La manutenzione di un ambiente di calcolo Amazon EKS è una responsabilità condivisa. Per ulteriori informazioni, consulta la sezione Sicurezza in Amazon EKS.

Fase 4: Creare una coda di lavoro e collegare l'ambiente di elaborazione

Important

È importante verificare che l'ambiente di elaborazione sia integro prima di procedere. A tale scopo è possibile utilizzare l'operazione DescribeComputeEnvironmentsAPI.

```
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1
```

Conferma che il `status` parametro non lo sia`INVALID`. Se lo è, esamina il `statusReason` parametro relativo alla causa. Per ulteriori informazioni, consulta Risoluzione dei problemi AWS Batch.

I lavori inviati a questa nuova coda di lavoro vengono eseguiti come pod su nodi AWS Batch gestiti che si sono uniti al cluster Amazon EKS associato al tuo ambiente di elaborazione.

```
$ cat <<EOF > ./batch-eks-job-queue.json
 {
    "jobQueueName": "My-Eks-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-CE1"
      }
    ]
  }
EOF
```

```
$ aws batch create-job-queue --cli-input-json file://./batch-eks-job-queue.json
```

Fase 5: creare un Amazon ECR con cache pull through

Poiché il cluster non dispone di accesso pubblico a Internet, è necessario creare un Amazon ECR per le immagini dei container. Le seguenti istruzioni creano un Amazon ECR con una regola di cache pull-through per archiviare l'immagine.

1. Il comando seguente crea la regola della cache pull-through. È possibile sostituirlo *tutorial-prefix* con un prefisso diverso.

   ```
   aws ecr create-pull-through-cache-rule \
       --ecr-repository-prefix "my-prefix" \
       --upstream-registry-url "public.ecr.aws" \
       --region us-east-1
   ```

2. Effettua l'autenticazione con l'ECR pubblico.

   ```
   aws ecr get-login-password --region us-east-1 | docker login --username AWS --
   password-stdin <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com
   ```

   Ora puoi estrarre un'immagine.

```
docker pull <your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
amazonlinux/amazonlinux:2
```

3. È possibile verificare il repository e l'immagine eseguendo i seguenti comandi:

```
aws ecr describe-repositories
```

```
aws ecr describe-images --repository-name my-prefix/amazonlinux/amazonlinux
```

4. La stringa di immagine da utilizzare per estrarre il contenitore è nel seguente formato:

```
<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/amazonlinux/
amazonlinux:2
```

Fase 6: Registrare una definizione di lavoro

La seguente definizione di Job indica al pod di dormire per 60 secondi.

Nel campo immagine della definizione del processo, invece di fornire un collegamento all'immagine in un archivio ECR pubblico, inserisci il link all'immagine memorizzata nel nostro archivio ECR privato. Vedi la seguente definizione di processo di esempio:

```
$ cat <<EOF > ./batch-eks-job-definition.json
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "<your-account-ID>.dkr.ecr.us-east-1.amazonaws.com/my-prefix/
amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "60"
          ],
          "resources": {
            "limits": {
              "cpu": "1",
```

```
              "memory": "1024Mi"
            }
          }
        }
      ],
      "metadata": {
        "labels": {
          "environment": "test"
        }
      }
    }
  }
}
EOF
```

```
$ aws batch register-job-definition --cli-input-json file://./batch-eks-job-definition.json
```

Note

- Esistono considerazioni relative ai `memory` parametri `cpu` and. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.

Fase 7: Inviare un lavoro da eseguire

Esegui il seguente AWS CLI comando AWS CloudShell per inviare un nuovo Job e restituisce l'unico JobID.

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
```

Note

- Per ulteriori informazioni sull'esecuzione di processi sulle risorse di Amazon EKS, consultaOfferte di lavoro Amazon EKS.

Fase 8: Visualizza l'output del job

Per verificare lo stato di un Job:

```
$ aws batch describe-jobs --job <JobID-from-submit-response>
```

I `startedAt` due `stoppedAt` dovrebbero trovarsi a un minuto di distanza l'uno dall'altro.

Fase 9: (Facoltativo) Inviare un lavoro con modifiche

Questo lavoro sostituisce il comando passato al contenitore.

```
$ cat <<EOF > ./submit-job-override.json
{
  "jobName": "EksWithOverrides",
  "jobQueue": "My-Eks-JQ1",
  "jobDefinition": "MyJobOnEks_Sleep",
  "eksPropertiesOverride": {
    "podProperties": {
      "containers": [
        {
          "command": [
            "/bin/sh"
          ],
          "args": [
            "-c",
            "echo hello world"
          ]
        }
      ]
    }
  }
}
EOF
```

```
$ aws batch submit-job - -cli-input-json file://./submit-job-override.json
```

Note

- Per una migliore visibilità dei dettagli delle operazioni, abilita la registrazione del piano di controllo di Amazon EKS. Per ulteriori informazioni, consulta la registrazione del piano di controllo di Amazon EKS nella Guida per l'utente di Amazon EKS.

- Daemonsetse il kubelets sovraccarico influisce sulle risorse di vCPU e memoria disponibili, in particolare sulla scalabilità e sul posizionamento lavorativo. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.

Passaggio 10: ripulisci le risorse del tutorial

Ti viene addebitato il costo dell' EC2 istanza Amazon mentre è abilitata. Puoi eliminare l'istanza per evitare di incorrere in addebiti.

Per eliminare le risorse che hai creato, procedi come segue:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione scegli Job queue.
3. Nella tabella Job queue scegli la coda Job che hai creato per il tutorial.
4. Da Azioni scegli Disabilita. Una volta che lo stato della coda Job è disabilitato, puoi scegliere Elimina.
5. Una volta eliminata la coda Job, nel pannello di navigazione scegli Ambienti di calcolo.
6. Scegli l'ambiente di calcolo che hai creato per questo tutorial, quindi scegli Disabilita dalle azioni. Potrebbero essere necessari 1-2 minuti per completare la disattivazione dell'ambiente di calcolo.
7. Una volta che lo stato dell'ambiente di calcolo è disabilitato, scegli Elimina. Potrebbero essere necessari 1-2 minuti per eliminare l'ambiente di calcolo.

Risorse aggiuntive

Dopo aver completato il tutorial, potresti voler esplorare i seguenti argomenti:

- Scopri di più sulle best practice.
- Esplora i componenti AWS Batch principali. Per ulteriori informazioni, consulta Componenti di AWS Batch.
- Scopri di più sui diversi ambienti di elaborazione disponibili in AWS Batch.
- Scopri di più sulle Job queues e sulle loro diverse opzioni di pianificazione.
- Scopri di più sulle definizioni di Job e sulle diverse opzioni di configurazione.
- Scopri di più sui diversi tipi di lavori.

Risoluzione dei problemi

Se i nodi avviati da AWS Batch non hanno accesso al repository Amazon ECR (o a qualsiasi altro repository) che memorizza l'immagine, i lavori potrebbero rimanere nello stato STARTING. Questo perché il pod non sarà in grado di scaricare l'immagine ed eseguire il processo. AWS Batch Se fai clic sul nome del pod lanciato da, AWS Batch dovresti essere in grado di vedere il messaggio di errore e confermare il problema. Il messaggio di errore dovrebbe essere simile al seguente:

```
Failed to pull image "public.ecr.aws/amazonlinux/amazonlinux:2": rpc error: code =
Unknown desc = failed to pull and unpack image
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to resolve reference
"public.ecr.aws/amazonlinux/amazonlinux:2": failed to do request: Head
"https://public.ecr.aws/v2/amazonlinux/amazonlinux/manifests/2": dial tcp: i/o timeout
```

Per altri scenari di risoluzione dei problemi comuni, consulta Risoluzione dei problemi AWS Batch. Per la risoluzione dei problemi in base allo stato del pod, vedi Come posso risolvere lo stato del pod in Amazon EKS? .

Guida introduttiva all'intelligenza artificiale AWS Batch SageMaker

AWS Batch i lavori di assistenza consentono di inviare lavori di SageMaker formazione tramite code di AWS Batch lavoro con funzionalità di pianificazione, prioritizzazione e messa in coda. Questo tutorial dimostra come configurare ed eseguire un semplice processo di formazione utilizzando i job di servizio. SageMaker AWS Batch

Indice

Panoramica di

Questo tutorial illustra come configurare i lavori di AWS Batch servizio per i lavori di SageMaker formazione utilizzando. AWS CLI

Destinatari

Questo tutorial è progettato per data scientist e sviluppatori responsabili della creazione e dell'esecuzione di corsi di formazione sull'apprendimento automatico su larga scala.

Funzionalità utilizzate

Questo tutorial mostra come utilizzare il file AWS CLI per:

- Creare un ambiente di servizio per i lavori di SageMaker formazione
- Crea una coda SageMaker di lavori di formazione
- Invia lavori di assistenza utilizzando l'API `SubmitServiceJob`
- Monitora lo stato del lavoro e visualizza i risultati
- Registri di CloudWatch accesso per i lavori di formazione

Tempo richiesto

Il completamento di questo tutorial richiede circa 15 minuti.

Restrizioni regionali

Questo tutorial può essere completato in qualsiasi AWS regione in cui siano disponibili AWS Batch sia l'IA che l' SageMaker intelligenza artificiale.

Costi di utilizzo delle risorse

La creazione di un AWS account è gratuita. Tuttavia, l'implementazione di questa soluzione potrebbe comportare costi per le seguenti risorse:

Description	Costo (dollari USA)
SageMaker Istanze di formazione basata sull'intelligenza artificiale	Paghi per ogni istanza di SageMaker AI Training utilizzata. Per ulteriori informazioni sui prezzi, consulta SageMaker AI Pricing.
Archiviazione Amazon S3	Costo minimo per l'archiviazione dei risultati dei lavori di formazione. Per ulteriori informazioni, consulta Prezzi di Amazon S3.

Prerequisiti

Prima di iniziare questo tutorial, devi installare e configurare i seguenti strumenti e risorse necessari per creare e gestire sia AWS Batch le risorse di SageMaker intelligenza artificiale.

- AWS CLI— Uno strumento a riga di comando per lavorare con AWS i servizi, inclusa AWS Batch l' SageMaker intelligenza artificiale. Questa guida richiede l'utilizzo della versione 2.8.6 o successiva. Per ulteriori informazioni, vedere Installazione, aggiornamento e disinstallazione di AWS CLI nella Guida per l'utente.AWS Command Line Interface Dopo aver installato AWS CLI, ti consigliamo di configurarlo anche. Per ulteriori informazioni, vedere Configurazione rapida con `aws configure` nella Guida AWS Command Line Interface per l'utente.

Fase 1: Creare un ruolo di esecuzione SageMaker AI

SageMaker L'intelligenza artificiale utilizza i ruoli di esecuzione per eseguire operazioni per tuo conto utilizzando altri AWS servizi. È necessario creare un ruolo di esecuzione e concedere all' SageMaker IA le autorizzazioni per utilizzare i servizi e le risorse necessari per i lavori di formazione. Utilizza la policy `AmazonSageMakerFullAccess` gestita in quanto include le autorizzazioni per Amazon S3.

> **Note**
>
> Utilizza le seguenti istruzioni per creare il ruolo di esecuzione SageMaker AI per questo tutorial.
> Prima di creare un ruolo di esecuzione per il tuo ambiente di produzione, ti consigliamo di leggere «Come usare i ruoli di esecuzione SageMaker AI» nella guida per sviluppatori SageMaker AI.

1. Crea il ruolo IAM

   Crea un file JSON denominato `sagemaker-trust-policy.json` con la seguente politica di fiducia:

   JSON

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

```
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "sagemaker.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

Crea il ruolo IAM utilizzando la policy di fiducia:

```
aws iam create-role \
    --role-name SageMakerExecutionRole \
    --assume-role-policy-document file://sagemaker-trust-policy.json \
    --description "Execution role for SageMaker training jobs"
```

2. Allega politiche gestite

   Allega le politiche gestite richieste al ruolo:

```
aws iam attach-role-policy \
    --role-name SageMakerExecutionRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonSageMakerFullAccess
```

```
aws iam attach-role-policy \
    --role-name SageMakerExecutionRole \
    --policy-arn arn:aws:iam::aws:policy/AmazonS3FullAccess
```

3. Nota il ruolo ARN

   Ottieni il ruolo ARN, che ti servirà nei passaggi successivi:

```
aws iam get-role --role-name SageMakerExecutionRole --query 'Role.Arn' --output
 text
```

   Salva questo ARN perché lo utilizzerai per creare il payload del tuo lavoro di formazione.

Fase 2: Crea il tuo ambiente di servizio

Un ambiente di servizio definisce i vincoli di capacità per i lavori di SageMaker formazione. L'ambiente di servizio incapsula il numero massimo di istanze di addestramento che possono essere eseguite contemporaneamente.

> **Important**
>
> Quando crei il tuo primo ambiente di servizio per SageMaker Training, crea AWS Batch automaticamente un ruolo collegato al servizio chiamato nel tuo account. `AWSServiceRoleForAWSBatchWithSagemaker` Questo ruolo consente di AWS Batch mettere in coda e gestire i lavori di SageMaker formazione per tuo conto. Per ulteriori informazioni su questo ruolo collegato al servizio e sulle relative autorizzazioni, consulta. the section called "SageMaker ruolo di integrazione"

Crea un ambiente di servizio in grado di gestire fino a 5 istanze:

```
aws batch create-service-environment \
    --service-environment-name TutorialServiceEnvironment \
    --service-environment-type SAGEMAKER_TRAINING \
    --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=5
```

Output:

```
{
    "serviceEnvironmentName": "TutorialServiceEnvironment",
    "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-id:service-
environment/TutorialServiceEnvironment"
}
```

Verifica che l'ambiente di servizio sia stato creato correttamente:

```
aws batch describe-service-environments --service-
environments TutorialServiceEnvironment
```

Output:

```
{
```

```
    "serviceEnvironments": [
        {
            "serviceEnvironmentName": "TutorialServiceEnvironment",
            "serviceEnvironmentArn": "arn:aws:batch:your-region:your-account-
id:service-environment/TutorialServiceEnvironment",
            "serviceEnvironmentType": "SAGEMAKER_TRAINING",
            "state": "ENABLED",
            "status": "VALID",
            "capacityLimits": [
                {
                    "maxCapacity": 5,
                    "capacityUnit": "NUM_INSTANCES"
                }
            ],
            "tags": {}
        }
    ]
}
```

Per ulteriori informazioni sugli ambienti di servizio, vedereAmbienti di servizio.

Fase 3: Crea la tua coda SageMaker di lavoro

Una coda di SageMaker lavoro gestisce la pianificazione e l'esecuzione dei lavori di servizio. I lavori inviati a questa coda verranno inviati all'ambiente di servizio in base alla capacità disponibile.

Crea una coda per i lavori di SageMaker formazione:

```
aws batch create-job-queue \
    --job-queue-name my-sm-training-fifo-jq \
    --job-queue-type SAGEMAKER_TRAINING \
    --priority 1 \
    --service-environment-order order=1,serviceEnvironment=TutorialServiceEnvironment
```

Output:

```
{
    "jobQueueName": "my-sm-training-fifo-jq",
    "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-
fifo-jq"
}
```

Verifica che la tua coda di lavoro sia stata creata correttamente:

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

Output:

```
{
    "jobQueues": [
        {
            "jobQueueName": "my-sm-training-fifo-jq",
            "jobQueueArn": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-
training-fifo-jq",
            "state": "ENABLED",
            "status": "VALID",
            "statusReason": "JobQueue Healthy",
            "priority": 1,
            "computeEnvironmentOrder": [],
            "serviceEnvironmentOrder": [
                {
                    "order": 1,
                    "serviceEnvironment": "arn:aws:batch:your-region:your-account-
id:service-environment/TutorialServiceEnvironment"
                }
            ],
            "jobQueueType": "SAGEMAKER_TRAINING",
            "tags": {}
        }
    ]
}
```

Per ulteriori informazioni sulle code di SageMaker lavoro, consulta. the section called "Crea una coda di SageMaker lavoro"

Fase 4: Creare e inviare un lavoro di formazione

Ora creerai un semplice lavoro di formazione e lo invierai alla tua lista di lavoro. Questo esempio utilizza un corso di formazione di base «hello world» che dimostra la funzionalità di Service Job.

Crea un archivio denominato *my_training_job.json* con i seguenti contenuti. Sostituisci *your-account-id* con l'ID AWS del tuo account:

> **Note**
>
> `S3OutputPath`è necessario per creare il processo di SageMaker formazione, ma i risultati di questo tutorial non sono archiviati nel bucket Amazon S3 ed è possibile utilizzare il percorso nel seguente formato JSON. Nel tuo ambiente di produzione avrai bisogno di un bucket Amazon S3 valido per archiviare lì l'output, se lo desideri.

```
{
    "TrainingJobName": "my-simple-training-job",
    "RoleArn": "arn:aws:iam::your-account-id:role/SageMakerExecutionRole",
    "AlgorithmSpecification": {
        "TrainingInputMode": "File",
        "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
        "ContainerEntrypoint": [
            "echo",
            "hello world"
        ]
    },
    "ResourceConfig": {
        "InstanceType": "ml.c5.xlarge",
        "InstanceCount": 1,
        "VolumeSizeInGB": 1
    },
    "OutputDataConfig": {
        "S3OutputPath": "s3://your-s3-bucket/output"
    },
    "StoppingCondition": {
        "MaxRuntimeInSeconds": 30
    }
}
```

Invia il lavoro di formazione utilizzando l'SubmitServiceJobAPI:

```
aws batch submit-service-job \
    --job-queue my-sm-training-fifo-jq \
    --job-name my-batch-sm-job \
    --service-job-type SAGEMAKER_TRAINING \
    --retry-strategy attempts=1 \
    --timeout-config attemptDurationSeconds=60 \
```

```
    --service-request-payload file://my_training_job.json
```

Output:

```
{
    "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
    "jobName": "my-batch-sm-job",
    "jobId": "your-job-id"
}
```

Per ulteriori informazioni sui payload dei lavori di assistenza, consultathe section called "Carichi utili per lavori di assistenza". Per ulteriori informazioni sull'invio di lavori di assistenza, vedere. the section called "Invia un lavoro di assistenza"

Fase 5: Monitoraggio dello stato del lavoro

Puoi monitorare i tuoi lavori di formazione utilizzando quanto segue AWS Batch APIs: DescribeServiceJob, ListServiceJobs, e GetJobQueueSnapshot. Questa sezione mostra diversi modi per controllare lo stato dei lavori e le informazioni sulla coda.

Visualizza i lavori in esecuzione nella coda:

```
aws batch list-service-jobs \
    --job-queue my-sm-training-fifo-jq --job-status RUNNING
```

Output:

```
{
    "jobSummaryList": [
        {
            "latestAttempt": {
                "serviceResourceId": {
                    "name": "TrainingJobArn",
                    "value": "arn:aws:sagemaker:your-region:your-account-id:training-
job/AWSBatch<my-simple-training-job><your-attempt-id>"
                }
            },
            "createdAt": 1753718760,
            "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-
id",
            "jobId": "your-job-id",
            "jobName": "my-batch-sm-job",
```

```
            "serviceJobType": "SAGEMAKER_TRAINING",
            "status": "RUNNING",
            "startedAt": 1753718820
        }
    ]
}
```

Visualizza i lavori che si trovano nello RUNNABLE stato:

```
aws batch list-service-jobs \
    --job-queue my-sm-training-fifo-jq --job-status RUNNABLE
```

Ottieni un'istantanea dei lavori imminenti in coda:

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Output:

```
{
    "frontOfQueue": {
        "jobs": [
            {
                "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
                "earliestTimeAtPosition": 1753718880
            },
            {
                "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id-2",
                "earliestTimeAtPosition": 1753718940
            }
        ],
        "lastUpdatedAt": 1753718970
    }
}
```

Cerca offerte di lavoro per nome:

```
aws batch list-service-jobs \
    --job-queue my-sm-training-fifo-jq \
    --filters name=JOB_NAME,values="my-batch-sm-job"
```

Output:

```
{
    "jobSummaryList": [
        {
            "latestAttempt": {
                "serviceResourceId": {
                    "name": "TrainingJobArn",
                    "value": "arn:aws:sagemaker:your-region:your-account-id:training-
job/AWSBatch<my-simple-training-job><your-attempt-id>"
                }
            },
            "createdAt": 1753718760,
            "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-
id",
            "jobId": "your-job-id",
            "jobName": "my-batch-sm-job",
            "serviceJobType": "SAGEMAKER_TRAINING",
            "status": "RUNNING"
        }
    ]
}
```

Per ulteriori informazioni sulla mappatura dello stato del lavoro, vederethe section called "Stato della mansione di servizio".

Fase 6: Visualizzazione dell'output del lavoro

Una volta completato il lavoro, puoi visualizzarne l'output e i log sia tramite l'intelligenza artificiale AWS Batch . SageMaker APIs

Ottieni informazioni dettagliate sul tuo lavoro da: AWS Batch

```
aws batch describe-service-job \
    --job-id your-job-id
```

Output:

```
{
    "attempts": [
        {
            "serviceResourceId": {
```

```
                "name": "TrainingJobArn",
                "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
            },
            "startedAt": 1753718820,
            "stoppedAt": 1753718880,
            "statusReason": "Received status from SageMaker: Training job completed"
        }
    ],
    "createdAt": 1753718760,
    "jobArn": "arn:aws:batch:your-region:your-account-id:service-job/your-job-id",
    "jobId": "your-job-id",
    "jobName": "my-batch-sm-job",
    "jobQueue": "arn:aws:batch:your-region:your-account-id:job-queue/my-sm-training-
fifo-jq",
    "latestAttempt": {
        "serviceResourceId": {
            "name": "TrainingJobArn",
            "value": "arn:aws:sagemaker:your-region:your-account-id:training-job/
AWSBatch<my-simple-training-job><your-attempt-id>"
        }
    },
    "retryStrategy": {
        "attempts": 1,
        "evaluateOnExit": []
    },
    "serviceRequestPayload": "your-training-job-request-json",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "startedAt": 1753718820,
    "status": "SUCCEEDED",
    "statusReason": "Received status from SageMaker: Training job completed",
    "stoppedAt": 1753718880,
    "tags": {},
    "timeoutConfig": {
        "attemptDurationSeconds": 60
    }
}
```

Questo comando restituisce informazioni complete sul lavoro, incluso l'ARN del lavoro di SageMaker formazione, che puoi utilizzare per accedere al lavoro direttamente tramite SageMaker AI:

```
aws sagemaker describe-training-job \
    --training-job-name AWSBatch<my-simple-training-job><your-attempt-id>
```

Per visualizzare CloudWatch i log del tuo processo di formazione, per prima cosa ottieni il nome del flusso di log:

```
aws logs describe-log-streams \
    --log-group-name /aws/sagemaker/TrainingJobs \
    --log-stream-name-prefix AWSBatchmy-simple-training-job
```

Output:

```
{
    "logStreams": [
        {
            "logStreamName": "your-log-stream-name",
            "creationTime": 1753718830,
            "firstEventTimestamp": 1753718840,
            "lastEventTimestamp": 1753718850,
            "lastIngestionTime": 1753718860,
            "uploadSequenceToken": upload-sequence-token,
            "arn": "arn:aws:logs:your-region:your-account-id:log-group:/aws/sagemaker/
TrainingJobs:log-stream:AWSBatch<my-simple-training-job><your-attempt-id>/algo-1-algo-
id",
            "storedBytes": 0
        }
    ]
}
```

Quindi recupera i log utilizzando il nome del flusso di log della risposta precedente:

```
aws logs get-log-events \
    --log-group-name /aws/sagemaker/TrainingJobs \
    --log-stream-name your-log-stream-name
```

Output:

```
{
    "events": [
        {
            "timestamp": 1753718845,
            "message": "hello world",
            "ingestionTime": 1753718865
        }
    ],
```

```
    "nextForwardToken": "next-forward-token",
    "nextBackwardToken": "next-backward-token"
}
```

L'output del registro mostra il messaggio «hello world» del processo di formazione, che conferma che il processo è stato eseguito correttamente.

Passaggio 7: ripulisci le risorse del tutorial

Quando hai finito con il tutorial, pulisci le risorse che hai creato per evitare addebiti continui.

Innanzitutto, disabilita ed elimina la coda dei lavori:

```
aws batch update-job-queue \
    --job-queue my-sm-training-fifo-jq \
    --state DISABLED
```

Attendi che la coda dei lavori sia disabilitata, quindi eliminala:

```
aws batch delete-job-queue \
    --job-queue my-sm-training-fifo-jq
```

Quindi, disabilita ed elimina l'ambiente di servizio:

```
aws batch update-service-environment \
    --service-environment TutorialServiceEnvironment \
    --state DISABLED
```

Attendi che l'ambiente di servizio sia disabilitato, quindi eliminalo:

```
aws batch delete-service-environment \
    --service-environment TutorialServiceEnvironment
```

Risorse aggiuntive

Dopo aver completato il tutorial, potresti voler esplorare i seguenti argomenti:

- Ti consigliamo di utilizzare PySDK per la creazione di lavori di servizio e l'invio alla coda dei lavori perché PySDK ha classi e utilità di supporto. Per un esempio di utilizzo di PySDK, consulta Esempi di intelligenza artificiale su. SageMaker GitHub

- Ulteriori informazioni su the section called "Lavori di servizio".
- Esplora configurazioni Carichi utili per lavori di assistenza in AWS Batch di lavoro di formazione più complesse.
- Scopri di più sull'API Invia un lavoro di assistenza in AWS Batch e sull'`SubmitServiceJob`API.
- Rivedi Mappatura dello stato AWS Batch del lavoro del servizio allo stato dell' SageMaker IA per comprendere le transizioni dello stato lavorativo.
- Consulta la documentazione di SageMaker AI Python SDK per ulteriori modi ricchi di funzionalità per creare e inviare lavori di formazione SageMaker utilizzando Python.
- Esplora taccuini di SageMaker esempio per flussi di lavoro di machine learning più complessi.

La dashboard AWS Batch dei widget

Tramite la AWS Batch dashboard, puoi monitorare i lavori recenti, le code di lavoro e gli ambienti di calcolo. Per impostazione predefinita, vengono visualizzati i seguenti widget del pannello di controllo:

- Panoramica dei lavori: per ulteriori informazioni sui AWS Batch lavori, vedereJobs.
- Panoramica delle code di lavoro: per ulteriori informazioni sulle code di AWS Batch lavoro, vedere. Job queues
- Panoramica dell'ambiente di calcolo: per ulteriori informazioni sugli ambienti di AWS Batch elaborazione, vedere. Ambienti di calcolo per AWS Batch

È possibile personalizzare i widget visualizzati nella pagina Dashboard. Le seguenti sezioni descrivono i widget aggiuntivi che è possibile aggiungere.

Argomenti

Come aggiungere il widget Single job queue alla dashboard AWS Batch

Il widget Single job queue mostra informazioni dettagliate su una singola coda di lavoro.

Per aggiungere questo widget, segui questi passaggi.

1. Apri la AWS Batch console.
2. Dalla barra di navigazione, scegli Regione AWS quello che desideri.
3. Nel pannello di navigazione seleziona Pannello di controllo.
4. Seleziona Add widget (Aggiungi widget).
5. Per Single job queue, scegli Aggiungi widget.
6. Per Job queue, seleziona la coda lavori che desideri.
7. Per Job status, scegli gli stati del lavoro che desideri visualizzare.

8. (Facoltativo) Disattiva Mostra ambienti di calcolo connessi se non desideri visualizzare le proprietà degli ambienti di calcolo.
9. Per le proprietà dell'ambiente di calcolo, seleziona le proprietà che desideri.
10. Scegli Aggiungi.

Come aggiungere il widget CloudWatch Container Insights alla dashboard AWS Batch

Questo widget mostra metriche aggregate per ambienti di AWS Batch calcolo e lavori. Per ulteriori informazioni su Container Insights, consulta the section called "CloudWatch Informazioni sui container".

Per aggiungere questo widget, segui questi passaggi.

1. Apri la AWS Batch console.
2. Dalla barra di navigazione, scegli Regione AWS quello che desideri.
3. Nel pannello di navigazione seleziona Pannello di controllo.
4. Seleziona Add widget (Aggiungi widget).
5. Per Container Insights, scegli Aggiungi widget.
6. Per Ambiente di calcolo, scegli l'ambiente di calcolo che desideri.
7. Scegli Aggiungi.

Come aggiungere il widget Job logs alla dashboard AWS Batch

Questo widget mostra diversi log dei tuoi lavori in un'unica comoda posizione. Per ulteriori informazioni sui registri dei lavori, consulta. the section called "Visualizza i registri dei lavori nei registri CloudWatch "

Per aggiungere questo widget, segui questi passaggi.

1. Apri la AWS Batch console.
2. Dalla barra di navigazione, scegli Regione AWS quello che desideri.
3. Nel pannello di navigazione seleziona Pannello di controllo.
4. Seleziona Add widget (Aggiungi widget).

5. Per Job logs, scegli Aggiungi widget.
6. Per Job id, inserisci l'ID del lavoro che desideri.
7. Scegli Aggiungi.

Ambienti di calcolo per AWS Batch

Le code di lavoro sono mappate su uno o più ambienti di elaborazione. Gli ambienti di calcolo contengono le istanze di container Amazon ECS utilizzate per eseguire processi batch containerizzati. Un ambiente di calcolo specifico può anche essere mappato su una o più di una coda di lavoro. All'interno di una coda di lavoro, gli ambienti di calcolo associati hanno ciascuno un ordine che viene utilizzato dallo scheduler per determinare dove verranno eseguiti i lavori pronti per essere eseguiti. Se il primo ambiente di calcolo ha lo stato di `VALID` e dispone di risorse disponibili, il processo viene programmato su un'istanza di contenitore all'interno di quell'ambiente di calcolo. Se il primo ambiente di calcolo ha lo stato di `INVALID` o non è in grado di fornire una risorsa di elaborazione adeguata, lo scheduler tenta di eseguire il processo nell'ambiente di calcolo successivo.

Argomenti

Ambienti di elaborazione gestiti

È possibile utilizzare un ambiente di elaborazione gestito per AWS Batch gestire la capacità e i tipi di istanze delle risorse di elaborazione all'interno dell'ambiente. Questo si basa sulle specifiche delle risorse di calcolo che definisci al momento della creazione dell'ambiente di calcolo. Puoi scegliere di utilizzare le istanze Amazon EC2 On-Demand e le istanze Amazon EC2 Spot. In alternativa, puoi utilizzare la capacità di Fargate e Fargate Spot nel tuo ambiente di elaborazione gestito. Quando

utilizzi le istanze Spot, puoi facoltativamente impostare un prezzo massimo. In questo modo, le istanze Spot vengono avviate solo quando il prezzo delle istanze Spot è inferiore a una percentuale specificata del prezzo on demand.

> **Important**
>
> Le istanze Fargate Spot non sono supportate su. Windows containers on AWS Fargate Una coda di lavoro verrà bloccata se un FargateWindows lavoro viene inviato a una coda di lavoro che utilizza solo ambienti di elaborazione Fargate Spot.

> **Important**
>
> AWS Batch crea e gestisce più AWS risorse per tuo conto e all'interno del tuo account, tra cui Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets e Amazon ECS Clusters. Queste risorse gestite sono configurate specificamente per garantire un funzionamento ottimale. AWS Batch La modifica manuale di queste risorse gestite in batch, a meno che non sia esplicitamente indicato nella AWS Batch documentazione, può comportare un comportamento imprevisto con conseguente ambiente di `INVALID` calcolo, un comportamento di scalabilità delle istanze non ottimale, un'elaborazione ritardata del carico di lavoro o costi imprevisti. Queste modifiche manuali non possono essere supportate in modo deterministico dal servizio. AWS Batch Usa sempre il Batch supportato APIs o la console Batch per gestire i tuoi ambienti di elaborazione.

Gli ambienti di elaborazione gestiti avviano EC2 le istanze Amazon nel VPC e nelle sottoreti specificate, quindi le registrano in un cluster Amazon ECS. Le EC2 istanze Amazon richiedono l'accesso alla rete esterna per comunicare con l'endpoint del servizio Amazon ECS. Alcune sottoreti non forniscono alle EC2 istanze Amazon indirizzi IP pubblici. Se le tue EC2 istanze Amazon non dispongono di un indirizzo IP pubblico, devono utilizzare la traduzione degli indirizzi di rete (NAT) per ottenere questo accesso. Per ulteriori informazioni, consulta Gateway NAT nella Guida per l'utente di Amazon VPC. Per ulteriori informazioni su come creare un VPC, vedere. Crea un cloud privato virtuale

Per impostazione predefinita, gli ambienti di elaborazione AWS Batch gestiti utilizzano una versione recente e approvata dell'AMI ottimizzata Amazon ECS per le risorse di calcolo. Tuttavia, potresti voler creare la tua AMI da utilizzare per i tuoi ambienti di elaborazione gestiti per vari motivi. Per ulteriori informazioni, consulta Risorsa di calcolo AMIs.

Note

AWS Batch non la aggiorna automaticamente AMIs in un ambiente di calcolo dopo la sua creazione. Ad esempio, non aggiorna il tuo ambiente AMIs di calcolo quando viene rilasciata una versione più recente dell'AMI ottimizzata per Amazon ECS. Sei responsabile della gestione del sistema operativo guest. Ciò include eventuali aggiornamenti e patch di sicurezza. Sei inoltre responsabile di qualsiasi software applicativo o utilità aggiuntivo che installi sulle risorse di elaborazione. Esistono due modi per utilizzare una nuova AMI per i tuoi AWS Batch lavori. Il metodo originale consiste nel completare questi passaggi:

1. Creare un nuovo ambiente di calcolo con la nuova AMI.
2. Aggiungere l'ambiente di calcolo a una coda di processi esistente.
3. Rimuovere il precedente ambiente di calcolo dalla coda di processi.
4. Eliminare l'ambiente di calcolo precedente.

Nell'aprile 2022, è AWS Batch stato aggiunto un supporto avanzato per l'aggiornamento degli ambienti di elaborazione. Per ulteriori informazioni, consulta Aggiornare un ambiente di calcolo in AWS Batch. Per utilizzare l'aggiornamento avanzato degli ambienti di elaborazione per l'aggiornamento AMIs, segui queste regole:

- Non impostate il parametro service role (`serviceRole`) o impostatelo sul ruolo collegato al AWSServiceRoleForBatchservizio.
- Imposta il parametro allocation strategy (`allocationStrategy`) su, o`BEST_FIT_PROGRESSIVE`. `SPOT_CAPACITY_OPTIMIZED` `SPOT_PRICE_CAPACITY_OPTIMIZED`
- Imposta il parametro di aggiornamento all'ultima versione dell'immagine (`updateToLatestImageVersion`) su`true`.
- Non specificare un ID AMI in `imageId`, `imageIdOverride`(in `ec2Configuration`) o nel modello di avvio (`launchTemplate`). In tal caso, AWS Batch seleziona l'AMI ottimizzata Amazon ECS più recente supportata da AWS Batch al momento dell'avvio dell'aggiornamento dell'infrastruttura. In alternativa, puoi specificare l'ID AMI nei `imageIdOverride` parametri `imageId` o o il modello di avvio identificato dalle `LaunchTemplate` proprietà. La modifica di una di queste proprietà avvia un aggiornamento dell'infrastruttura. Se l'ID AMI è specificato nel modello di avvio, non può essere sostituito specificando un ID AMI nei `imageIdOverride` parametri `imageId` o. Può essere sostituito solo specificando un modello di lancio diverso. Oppure, se la

versione del modello di lancio è impostata su `$Default` o`$Latest`, impostando una nuova versione predefinita per il modello di lancio (se lo è`$Default`) o aggiungendo una nuova versione al modello di lancio (se lo è`$Latest`).

Se vengono seguite queste regole, qualsiasi aggiornamento che avvia un aggiornamento dell'infrastruttura causerà la riselezione dell'ID AMI. Se l'`version`impostazione nel modello di avvio (`launchTemplate`) è impostata su `$Latest` o`$Default`, la versione più recente o predefinita del modello di lancio viene valutata al momento dell'aggiornamento dell'infrastruttura, anche se non `launchTemplate`è stata aggiornata.

Considerazione da prendere in considerazione durante la creazione di lavori paralleli a più nodi

AWS Batch consiglia di creare ambienti di elaborazione dedicati per l'esecuzione di job multi-node parallel (MNP) e processi non MNP. Ciò è dovuto al modo in cui viene creata la capacità di elaborazione nell'ambiente di elaborazione gestito. Quando si crea un nuovo ambiente di elaborazione gestito, se si specifica un `minvCpu` valore maggiore di zero, viene AWS Batch creato un pool di istanze da utilizzare solo con processi non MNP. Se viene inviato un processo parallelo multinodo, AWS Batch crea una nuova capacità di istanza per eseguire i processi paralleli multinodo. Nei casi in cui vi siano processi paralleli a nodo singolo e multinodo in esecuzione nello stesso ambiente di calcolo in cui è impostato un `maxvCpus` valore `minvCpus` o, se le risorse di elaborazione richieste non sono disponibili, AWS Batch aspetterà il completamento dei processi correnti prima di creare le risorse di elaborazione necessarie per eseguire i nuovi processi.

Ambienti di elaborazione non gestiti

In un ambiente di calcolo non gestito occorre gestire le proprie risorse di calcolo. Devi verificare che l'AMI che usi per le tue risorse di calcolo soddisfi le specifiche AMI dell'istanza di container Amazon ECS. Per ulteriori informazioni, consultare Specifiche dell'AMI delle risorse di calcolo e Tutorial: Creare un'AMI per risorse di calcolo.

> **Note**
>
> AWS Le risorse Fargate non sono supportate negli ambienti di elaborazione non gestiti.

Dopo aver creato l'ambiente di elaborazione non gestito, utilizza l'operazione DescribeComputeEnvironmentsAPI per visualizzare i dettagli dell'ambiente di calcolo. Trova il cluster Amazon ECS associato all'ambiente e poi avvia manualmente le istanze di container in quel cluster Amazon ECS.

Il AWS CLI comando seguente fornisce anche l'ARN del cluster Amazon ECS.

```
$ aws batch describe-compute-environments \
    --compute-environments unmanagedCE \
    --query "computeEnvironments[].ecsClusterArn"
```

Per maggiori informazioni, consulta Avvio di un'istanza di container Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Container Service. Quando avvii le risorse di elaborazione, specifica l'ARN del cluster Amazon ECS che le risorse registrano con i seguenti dati utente Amazon. EC2 Sostituire *ecsClusterArn* con l'ARN del cluster ottenuto con il comando precedente.

```
#!/bin/bash
echo "ECS_CLUSTER=ecsClusterArn" >> /etc/ecs/ecs.config
```

Crea un ambiente di elaborazione

Prima di poter eseguire i lavori AWS Batch, è necessario creare un ambiente di elaborazione. Puoi creare un ambiente di elaborazione gestito in cui AWS Batch gestire EC2 le istanze Amazon o le risorse AWS Fargate all'interno dell'ambiente in base alle tue specifiche. In alternativa, puoi creare un ambiente di elaborazione non gestito in cui gestire la configurazione delle EC2 istanze Amazon all'interno dell'ambiente.

Important

Le istanze Fargate Spot non sono supportate nei seguenti scenari:

- Windows containers on AWS Fargate

Una coda di lavoro verrà bloccata in questi scenari se un lavoro viene inviato a una coda di lavoro che utilizza solo ambienti di elaborazione Fargate Spot.

Argomenti

- Tutorial: Creare un ambiente di elaborazione gestito utilizzando le risorse Fargate
- Tutorial: crea un ambiente di elaborazione gestito utilizzando le risorse Amazon EC2
- Tutorial: crea un ambiente di elaborazione non gestito utilizzando le risorse Amazon EC2
- Tutorial: crea un ambiente di elaborazione gestito utilizzando le risorse Amazon EKS
- Risorsa: modello di ambiente di calcolo
- Tabella di calcolo del tipo di istanza

Tutorial: Creare un ambiente di elaborazione gestito utilizzando le risorse Fargate

Completa i seguenti passaggi per creare un ambiente di elaborazione gestito utilizzando le risorse. AWS Fargate

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel riquadro di navigazione, seleziona Compute environments (Ambienti di calcolo).
4. Scegli Create (Crea).
5. Configura l'ambiente di calcolo.

   > **Note**
   >
   > Gli ambienti di elaborazione per i Windows containers on AWS Fargate lavori devono avere almeno una vCPU.

   a. Per la configurazione dell'ambiente di calcolo, scegli Fargate.

   b. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri di lunghezza. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   c. Per il ruolo di servizio, scegli il ruolo collegato al servizio che consente al AWS Batch servizio di effettuare chiamate alle operazioni AWS API richieste per tuo conto. Ad esempio, scegli AWSServiceRoleForBatch. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Batch.

d. (Facoltativo) Espandi i tag. Per aggiungere un tag, scegli Add tag (Aggiungi tag). Quindi, inserisci un nome chiave e un valore opzionale. Seleziona Aggiungi tag.

e. Scegli Pagina successiva.

6. Nella sezione Configurazione dell'istanza:

a. (Facoltativo) Per utilizzare la capacità di Fargate Spot, attivate Fargate Spot. Per informazioni su Fargate Spot, consulta Using Amazon Spot e EC2 Fargate_spot.

b. Per Maximum v CPUs, scegli il numero massimo di v fino a CPUs cui il tuo ambiente di elaborazione può scalare orizzontalmente, indipendentemente dalla domanda di lavoro in coda.

c. Scegli Pagina successiva.

7. Configurare le reti.

> **Important**
>
> Le risorse di calcolo richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite risorse di calcolo con indirizzi IP pubblici.
> Per ulteriori informazioni sugli endpoint di interfaccia Amazon ECR, consulta Endpoint VPC dell'interfaccia Amazon ECS (AWS PrivateLink) nella Guida per gli sviluppatori di Amazon Elastic Container Service.
> Se non disponi di un endpoint VPC di interfaccia configurato e le risorse di calcolo non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta Gateway NATnella Guida per l'utente di Amazon VPC. Per ulteriori informazioni, consulta the section called “Crea un VPC”.

a. Per l'ID Virtual Private Cloud (VPC), scegli un VPC in cui desideri avviare le tue istanze.

b. Per le sottoreti, scegli le sottoreti da utilizzare. Per impostazione predefinita, sono disponibili tutte le sottoreti all'interno del VPC selezionato.

> **Note**
>
> AWS Batch on Fargate attualmente non supporta Local Zones. Per ulteriori informazioni, consulta i cluster Amazon ECS in Local Zones, Wavelength Zones e nella Amazon Elastic Container Service AWS Outposts Developer Guide.

c. Per Security groups (Gruppi di sicurezza), scegli un gruppo di sicurezza da collegare alle tue istanze. Per impostazione predefinita, viene scelto il gruppo di sicurezza predefinito per il tuo VPC.

d. Scegli Pagina successiva.

8. Per Revisione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea ambiente di calcolo.

Tutorial: crea un ambiente di elaborazione gestito utilizzando le risorse Amazon EC2

Completa i seguenti passaggi per creare un ambiente di calcolo gestito utilizzando le risorse Amazon Elastic Compute Cloud EC2 (Amazon).

1. Apri la AWS Batch console all'indirizzo. https://console.aws.amazon.com/batch/
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
4. Scegli Crea ambiente e poi Ambiente di calcolo.
5. Configura l'ambiente.

   a. Per la configurazione dell'ambiente di calcolo, scegli Amazon Elastic Compute Cloud (Amazon EC2).

   b. Per il tipo di orchestrazione, scegli Managed.

   c. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri di lunghezza. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   d. Per il ruolo di servizio, scegli il ruolo collegato al servizio che consente al AWS Batch servizio di effettuare chiamate alle operazioni AWS API richieste per tuo conto. Ad esempio,

scegli AWSServiceRoleForBatch. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Batch.

e. Per Instance role (Ruolo istanza), scegli se creare un nuovo profilo dell'istanza o se utilizzare un profilo dell'istanza esistente che includa le autorizzazioni IAM necessarie. Questo profilo di istanza consente alle istanze del contenitore Amazon ECS create per il tuo ambiente di calcolo di effettuare chiamate alle operazioni AWS API richieste per tuo conto. Per ulteriori informazioni, consulta Ruolo dell'istanza Amazon ECS. Se scegli di creare un nuovo profilo dell'istanza, il ruolo richiesto (`ecsInstanceRole`) viene creato per te.

f. (Facoltativo) Espandi i tag.

   i. (Facoltativo) Per i EC2 tag, scegli Aggiungi tag per aggiungere un tag alle risorse che vengono lanciate nell'ambiente di calcolo. Quindi, inserisci un nome chiave e un valore opzionale. Seleziona Aggiungi tag.

   ii. (Facoltativo) Per Tag, scegli Aggiungi tag. Quindi, inserisci un nome chiave e un valore opzionale. Seleziona Aggiungi tag.

   Per ulteriori informazioni, consulta Tagga le tue AWS Batch risorse.

g. Scegli Next (Successivo).

6. Nella sezione Configurazione dell'istanza:

   a. (Facoltativo) Per abilitare l'utilizzo delle istanze Spot, attiva Spot. Per ulteriori informazioni, consulta Istanze spot.

   b. (Solo Spot) Per ottenere una percentuale massima del prezzo on demand, scegli la percentuale massima che può rappresentare il prezzo di un'istanza Spot rispetto al prezzo on demand per quel tipo di istanza prima del lancio delle istanze. Ad esempio, se il prezzo massimo è del 20%, il prezzo Spot deve essere inferiore al 20% del prezzo on demand corrente per quell'istanza. EC2 Il prezzo da corrispondere sarà sempre il prezzo (di mercato) più basso, mai superiore alla percentuale massima impostata. Se lasci questo campo vuoto, il valore di default è 100% del prezzo on demand.

   c. (Solo Spot) Per il ruolo della flotta Spot, scegli un ruolo IAM di Amazon EC2 Spot Fleet esistente da applicare al tuo ambiente di calcolo Spot. Se non disponi già di un ruolo IAM di Amazon EC2 Spot Fleet esistente, devi prima crearne uno. Per ulteriori informazioni, consulta Ruolo della flotta di Amazon EC2 spot.

> **Important**
>
> Per etichettare le istanze Spot al momento della creazione, il ruolo IAM di Amazon EC2 Spot Fleet deve utilizzare la più recente policy EC2 SpotFleetTaggingRole gestita da Amazon. La policy EC2 SpotFleetRole gestita da Amazon non dispone delle autorizzazioni necessarie per etichettare le istanze Spot. Per ulteriori informazioni, consultare Istanze Spot non taggate al momento della creazione e the section called "Assegnazione di tag alle risorse".

d. Per Minimum v CPUs, scegli il numero minimo di v CPUs che il tuo ambiente di elaborazione mantiene, indipendentemente dalla domanda di lavoro in coda.

e. Per Desired v CPUs, scegli il numero di v con CPUs cui il tuo ambiente di calcolo viene avviato. All'aumentare della domanda in coda di lavoro, AWS Batch è possibile aumentare il numero desiderato di v CPUs nell'ambiente di calcolo e aggiungere EC2 istanze, fino al massimo v. CPUs Al diminuire della domanda, AWS Batch è possibile ridurre il numero desiderato di v CPUs nell'ambiente di calcolo e rimuovere le istanze, fino al minimo v. CPUs

f. Per Maximum v CPUs, scegli il numero massimo di v verso CPUs cui il tuo ambiente di elaborazione può scalare orizzontalmente, indipendentemente dalla domanda di lavoro in coda.

g. Per i tipi di istanze consentiti, scegli i tipi di EC2 istanze Amazon che possono essere avviati. Puoi specificare famiglie di istanze per avviare qualsiasi tipo di istanza all'interno di tali famiglie (ad esempio `c5`c5n,, op3). In alternativa, potete specificare dimensioni specifiche all'interno di una famiglia (ad esempio`c5.8xlarge`). I tipi di istanze in metallo non rientrano nelle famiglie di istanze. Ad esempio, `c5` non include`c5.metal`.

   AWS Batch può selezionare il tipo di istanza che fa per voi se scegliete una delle seguenti opzioni:

   - `optimal`per selezionare i tipi di istanze (dalle famiglie `c4` `m4`r4,`c5`,`m5`, e di `r5` istanze) che corrispondono alla domanda delle vostre code di lavoro.
   - `default_x86_64`per scegliere tipi di istanze basati su x86 (dalle famiglie di c7i istanze m6i c6ir6i,, e) che soddisfino le richieste di risorse della coda di lavoro.
   - `default_arm64`per scegliere tipi di istanze basati su x86 (dalle famiglie di c7g istanze m6g c6gr6g,, e) che soddisfino le richieste di risorse della coda di lavoro.

> **Note**
>
> A partire dal 01/11/2025, il comportamento di `optimal` verrà modificato per corrispondere. `default_x86_64` Durante la modifica, le famiglie di istanze potrebbero essere aggiornate a una nuova generazione. Non è necessario eseguire alcuna azione affinché l'aggiornamento avvenga. Per ulteriori informazioni sulla modifica, vedere
>
> > **Note**
> >
> > A partire dall'11/01/2025, il comportamento di `optimal` verrà modificato per corrispondere. `default_x86_64`
>
> Durante la modifica, le famiglie di istanze potrebbero essere aggiornate a una nuova generazione. Non è necessario eseguire alcuna azione affinché l'aggiornamento avvenga.
>
> AWS Batch supportava un'unica opzione in InstanceTypes `optimal` per soddisfare la domanda delle vostre code di lavoro. Abbiamo introdotto due nuove opzioni di tipo di istanza: e. `default_x86_64` `default_arm64` Le useremo `default_x86_64` se non effettui alcuna selezione del tipo di istanza. Queste nuove opzioni selezioneranno automaticamente tipi di istanze convenienti tra diverse famiglie e generazioni in base ai requisiti della coda di lavoro, consentendoti di far funzionare rapidamente i tuoi carichi di lavoro.
>
> Non appena sarà disponibile una capacità sufficiente per nuovi tipi di istanze in un Regione AWS, il pool predefinito corrispondente verrà aggiornato automaticamente con il nuovo tipo di istanza. L'`optimal`opzione esistente continuerà a essere supportata e non è obsoleta, in quanto sarà supportata dai pool predefiniti sottostanti per fornire istanze aggiornate in futuro. Se stai usando '`optimal`, non è necessaria alcuna azione da parte tua.

Tuttavia, tieni presente che solo `ENABLED` and `VALID` Compute Environments (CEs) verrà aggiornato automaticamente con nuovi tipi di istanze. Se ne hai uno `DISABLED` o `INVALID` CEs, riceveranno aggiornamenti una volta riattivati e impostati su uno `VALID` stato.

.

Note

- La disponibilità della famiglia di istanze varia a seconda Regione AWS. Per esempio, alcuni Regione AWS potrebbero non avere famiglie di istanze di quarta generazione ma avere famiglie di istanze di quinta e sesta generazione.
- Quando si utilizzano i `default_x86_64` nostri pacchetti di `default_arm64` istanze, AWS Batch seleziona le famiglie di istanze in base a un equilibrio tra economicità e prestazioni. Sebbene le istanze di nuova generazione offrano spesso un miglior rapporto prezzo/prestazioni, è AWS Batch possibile scegliere una famiglia di istanze della generazione precedente se offre la combinazione ottimale di disponibilità, costi e prestazioni per il carico di lavoro. Ad esempio, in un Regione AWS paese in cui sono disponibili sia le istanze c6i che le istanze c7i, è AWS Batch possibile selezionare le istanze c6i se offrono un miglior rapporto qualità-prezzo per le specifiche esigenze lavorative. Per ulteriori informazioni sui tipi di istanze e sulla disponibilità, consulta Tabella di calcolo dei tipi di AWS Batch istanza. Regione AWS
- AWS Batch aggiorna periodicamente le istanze nei pacchetti predefiniti con opzioni più nuove e più convenienti. Gli aggiornamenti avvengono automaticamente senza richiedere alcuna azione da parte dell'utente. I carichi di lavoro continuano a funzionare durante gli aggiornamenti senza interruzioni.

> **Note**
>
> Quando crei un ambiente di calcolo, i tipi di istanza selezionati per l'ambiente di calcolo devono condividere la stessa architettura. Ad esempio, non puoi combinare istanze x86 e ARM nello stesso ambiente di calcolo.

> **Note**
>
> AWS Batch verrà GPUs scalato in base alla quantità richiesta nelle code di lavoro. Per utilizzare la pianificazione tramite GPU, l'ambiente di calcolo deve includere tipi di istanze appartenenti alle famigliep3,p4,p5,p6,g3, g3sg4, g5 o. g6

h. Per Allocation strategy (Strategia di allocazione), scegli la strategia di allocazione da utilizzare quando si selezionano i tipi di istanza dall'elenco dei tipi di istanza consentiti. BEST_FIT_PROGRESSIVE è in genere la scelta migliore per gli ambienti di calcolo EC2 On-Demand, SPOT_CAPACITY_OPTIMIZED e SPOT_PRICE_CAPACITY_OPTIMIZED per gli ambienti di calcolo Spot. EC2 Per ulteriori informazioni, consulta the section called "Strategie di allocazione del tipo di istanza".

i. Espandere Additional configuration (Configurazione aggiuntiva).

   i. (Facoltativo) In Placement group, inserite il nome del gruppo di posizionamento per raggruppare le risorse nell'ambiente di calcolo.

   ii. (Facoltativo) Per la coppia di EC2 chiavi, scegli una coppia di chiavi pubblica e una privata come credenziali di sicurezza quando ti connetti all'istanza. Per ulteriori informazioni sulle coppie di EC2 chiavi Amazon, consulta Coppie di EC2 chiavi Amazon e istanze Linux.

   iii. (Facoltativo) Per la EC2 configurazione scegli i valori Image type e Image ID override per AWS Batch fornire informazioni su come selezionare Amazon Machine Images (AMIs) per le istanze nell'ambiente di calcolo. Se l'override dell'ID immagine non è specificato per ogni tipo di immagine, AWS Batch seleziona un'AMI ottimizzata per Amazon ECS recente. Se non viene specificato alcun tipo di immagine, l'impostazione predefinita è Amazon Linux 2 per istanze non GPU e non AWS Graviton.

> **Important**
>
> Per utilizzare un'AMI personalizzata, scegli il tipo di immagine, quindi inserisci l'ID AMI personalizzato nella casella Ignora ID immagine.

Amazon Linux 2

È predefinito per tutte le famiglie di istanze AWS basate su Graviton (ad esempio, `C6g` `M6gR6g`, `eT4g`) e può essere utilizzato per tutti i tipi di istanze non GPU.

Amazon Linux 2 (GPU)

È predefinita per tutte le famiglie di istanze GPU (ad esempio `P4` `eG4`) e può essere utilizzata per tutti i tipi di istanze non basati su Graviton. AWS

Amazon Linux 2023

AWS Batch supporta Amazon Linux 2023.

> **Note**
>
> Amazon Linux 2023 non supporta `A1` le istanze.

Amazon Linux 2023 (GPU)

È predefinita per tutte le famiglie di istanze GPU (ad esempio `P4` `eG4`) e può essere utilizzata per tutti i tipi di istanze non basati su AWS Graviton.

> **Note**
>
> L'AMI scelta per un ambiente di elaborazione deve corrispondere all'architettura dei tipi di istanza che intendi utilizzare per quell'ambiente di elaborazione. Ad esempio, se l'ambiente di elaborazione utilizza tipi di istanza A1, l'AMI della risorsa di calcolo scelta deve supportare le istanze ARM. Amazon ECS offre versioni x86 e ARM dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Per

ulteriori informazioni, consulta l'AMI Amazon Linux 2 ottimizzata per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

j. (Opzionale) Espandi i modelli di Launch

i. Per Modello di lancio predefinito, seleziona un modello di EC2 lancio Amazon esistente per configurare le tue risorse di calcolo. La versione predefinita del modello viene compilata automaticamente. Per ulteriori informazioni, consulta Usa i modelli di EC2 lancio di Amazon con AWS Batch.

> **Note**
>
> In un modello di lancio, puoi specificare un AMI personalizzato che hai creato.

ii. (Facoltativo) Per la versione predefinita `$Default$Latest`, immettete o un numero di versione specifico da utilizzare.

> **Note**
>
> Nota: se utilizzi una delle variabili di sostituzione ($Default o $Latest), applicheranno il numero di versione corrente predefinito o più recente al momento del salvataggio della configurazione. Se la versione predefinita o la versione più recente verranno modificate in futuro, è necessario aggiornare le informazioni, che non verranno aggiornate automaticamente.

> **Important**
>
> Se il parametro di versione del modello di avvio è `$Default` o`$Latest`, la versione predefinita o più recente del modello di avvio specificato viene valutata durante un aggiornamento dell'infrastruttura. Se per impostazione predefinita è selezionato un ID AMI diverso o è selezionata la versione più recente del modello di avvio, tale ID AMI viene utilizzato nell'aggiornamento. Per ulteriori informazioni, consulta the section called “Selezione dell'AMI durante gli aggiornamenti dell'infrastruttura”.

iii. (Facoltativo) Per Sostituisci modello di lancio, scegli Aggiungi modello di lancio sostitutivo

A. (Facoltativo) Per il modello di lancio di Amazon, seleziona un modello di EC2 lancio Amazon esistente da utilizzare per tipi e famiglie di istanze specifici.

B. (Facoltativo) Per la versione predefinita, inserisci un numero di versione specifico da utilizzare`$Default`, oppure`$Latest`.

> **Note**
>
> Se si utilizza la `$Latest` variabile `$Default` o, AWS Batch applicherà le informazioni correnti al momento della creazione dell'ambiente di calcolo. Se la versione predefinita o più recente verrà modificata in futuro, è necessario aggiornare le informazioni tramite UpdateComputeEnvironmento tramite il pulsante Console di gestione AWS - AWS Batch.

C. (Facoltativo) Per i tipi di istanze di Target, seleziona il tipo o la famiglia di istanza a cui desideri applicare il modello di avvio override.

> **Note**
>
> Se specifichi un modello di avvio sostitutivo, sono necessari i tipi di istanze di Target. Per ulteriori informazioni, consultaLaunchTemplateSpecificationOverride. targetInstanceTypes.

> **Note**
>
> Se il tipo o la famiglia di istanze che desideri selezionare non è presente in questo elenco, rivedi le selezioni effettuate. `Allowed instance types`

k. Scegli Next (Successivo).

7. Nella sezione Configurazione di rete:

> **Important**
>
> Le risorse di calcolo richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite risorse di calcolo con indirizzi IP pubblici.

Per ulteriori informazioni sugli endpoint di interfaccia Amazon ECR, consulta Endpoint VPC dell'interfaccia Amazon ECS (AWS PrivateLink) nella Guida per gli sviluppatori di Amazon Elastic Container Service.
Se non disponi di un endpoint VPC di interfaccia configurato e le risorse di calcolo non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta Gateway NATnella Guida per l'utente di Amazon VPC. Per ulteriori informazioni, consulta the section called "Crea un VPC".

a. Per l'ID Virtual Private Cloud (VPC), scegli un VPC su cui avviare le tue istanze.

b. Per le sottoreti, scegli le sottoreti da utilizzare. Per impostazione predefinita, sono disponibili tutte le sottoreti all'interno del VPC selezionato.

> **Note**
>
> AWS Batch su Amazon EC2 supporta Local Zones. Per ulteriori informazioni, consulta Local Zones nella Amazon EC2 User Guide e i cluster Amazon ECS in Local Zones, Wavelength Zones e nella Amazon Elastic Container Service AWS Outposts Developer Guide.

c. (Facoltativo) Per i gruppi di sicurezza, scegli un gruppo di sicurezza da collegare alle tue istanze. Per impostazione predefinita, viene scelto il gruppo di sicurezza predefinito per il tuo VPC.

> **Note**
>
> Nota: se utilizzi una delle variabili di sostituzione ($Default o $Latest), applicheranno il numero di versione corrente predefinito o più recente al momento del salvataggio della configurazione. Se la versione predefinita o la versione più recente verranno modificate in futuro, è necessario aggiornare le informazioni, che non verranno aggiornate automaticamente.

8. Scegli Pagina successiva.

9. Per Revisione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea ambiente di calcolo.

Tutorial: crea un ambiente di elaborazione non gestito utilizzando le risorse Amazon EC2

Completa i seguenti passaggi per creare un ambiente di calcolo non gestito utilizzando le risorse Amazon Elastic Compute Cloud (Amazon EC2).

1. Apri la console all' AWS Batch indirizzo. https://console.aws.amazon.com/batch/
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nella pagina Ambienti di calcolo, scegli Crea.
4. Configura l'ambiente.
   a. Per la configurazione dell'ambiente di calcolo, scegli Amazon Elastic Compute Cloud (Amazon EC2).
   b. Per il tipo di orchestrazione, scegli Unmanaged.
5. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
6. Per Ruolo di servizio, scegli un ruolo che consenta al AWS Batch servizio di effettuare chiamate alle operazioni AWS API richieste per tuo conto.

   > **Note**
   >
   > Non puoi utilizzarlo `AWSServiceRoleForBatch` per ambienti di elaborazione non gestiti.

7. Per Maximum v CPUs, scegli il numero massimo di v a CPUs cui il tuo ambiente di elaborazione può scalare orizzontalmente, indipendentemente dalla domanda di lavoro in coda.
8. (Facoltativo) Espandi i tag. Per aggiungere un tag, scegli Add tag (Aggiungi tag). Quindi, inserisci un nome chiave e un valore opzionale. Seleziona Aggiungi tag. Per ulteriori informazioni, consulta Tagga le tue AWS Batch risorse.
9. Scegli Pagina successiva.
10. Per Revisione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea ambiente di calcolo.

Tutorial: crea un ambiente di elaborazione gestito utilizzando le risorse Amazon EKS

Completa i seguenti passaggi per creare un ambiente di elaborazione gestito utilizzando le risorse Amazon Elastic Kubernetes Service (Amazon EKS).

1. Apri la console all'indirizzo. AWS Batch https://console.aws.amazon.com/batch/
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
3. Nel riquadro di navigazione, seleziona Compute environments (Ambienti di calcolo).
4. Scegli Create (Crea).
5. Per la configurazione dell'ambiente di calcolo, scegli Amazon Elastic Kubernetes Service (Amazon EKS).
6. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
7. Per il ruolo Instance, scegli un profilo di istanza esistente a cui siano associate le autorizzazioni IAM richieste.

   > **Note**
   >
   > Per creare un ambiente di calcolo nella AWS Batch console, scegli un profilo di istanza con le autorizzazioni `eks:ListClusters` and`eks:DescribeCluster`.

8. Per il cluster EKS, scegli un cluster Amazon EKS esistente.
9. Per Namespace, inserisci uno spazio Kubernetes dei nomi per raggruppare i AWS Batch processi nel cluster.
10. (Facoltativo) Espandi i tag. Scegli Aggiungi tag, quindi inserisci una coppia chiave-valore.
11. Scegli Pagina successiva.
12. (Facoltativo) Per utilizzare le istanze EC2 Spot, attiva Abilita l'utilizzo delle istanze Spot per utilizzare le istanze Amazon EC2 Spot.
13. (Solo Spot) Per una percentuale massima di prezzo on demand, scegli la percentuale massima che può rappresentare il prezzo di un'istanza Spot rispetto al prezzo on demand per quel tipo di istanza prima del lancio delle istanze. Ad esempio, se il prezzo massimo è del 20%, il prezzo Spot deve essere inferiore al 20% del prezzo on demand corrente per quell'istanza. EC2 Il prezzo da corrispondere sarà sempre il prezzo (di mercato) più basso, mai superiore alla

percentuale massima impostata. Se lasci questo campo vuoto, il valore di default è 100% del prezzo on demand.

14. (Solo Spot) Per il ruolo della flotta Spot, scegli il ruolo IAM della flotta Amazon EC2 Spot per l'ambiente di `SPOT` elaborazione.

    > **Important**
    >
    > Questo ruolo è obbligatorio se la strategia di allocazione è impostata `BEST_FIT` o non è specificata.

15. (Facoltativo) Per Minimum v CPUs, scegli il numero minimo di v CPUs che l'ambiente di elaborazione mantiene, indipendentemente dalla domanda di lavoro in coda.
16. (Facoltativo) Per Maximum v CPUs, scegli il numero massimo di v a CPUs cui l'ambiente di elaborazione può scalare orizzontalmente, indipendentemente dalla domanda di lavoro in coda.
17. Per i tipi di istanze consentiti, scegli i tipi di EC2 istanze Amazon che possono essere avviati. Puoi specificare famiglie di istanze per avviare qualsiasi tipo di istanza all'interno di tali famiglie (ad esempio `c5` `c5n`,, `op3`). In alternativa, potete specificare dimensioni specifiche all'interno di una famiglia (ad esempio`c5.8xlarge`). I tipi di istanze in metallo non rientrano nelle famiglie di istanze. Ad esempio, `c5` non include`c5.metal`.

    AWS Batch può selezionare il tipo di istanza che fa per te se scegli una delle seguenti opzioni:

    - `optimal`per selezionare i tipi di istanze (dalle famiglie `c4` `m4` `r4`,`c5`,`m5`, e di `r5` istanze) che corrispondono alla domanda delle vostre code di lavoro.
    - `default_x86_64`per scegliere tipi di istanze basati su x86 (dalle famiglie di c7i istanze m6i c6ir6i,, e) che soddisfino le richieste di risorse della coda di lavoro.
    - `default_arm64`per scegliere tipi di istanze basati su x86 (dalle famiglie di c7g istanze m6g c6gr6g,, e) che soddisfino le richieste di risorse della coda di lavoro.

    > **Note**
    >
    > A partire dal 01/11/2025, il comportamento di `optimal` verrà modificato per corrispondere. `default_x86_64` Durante la modifica, le famiglie di istanze potrebbero essere aggiornate a una nuova generazione. Non è necessario eseguire alcuna azione affinché l'aggiornamento avvenga. Per ulteriori informazioni sulla modifica,

vedereConfigurazione ottimale del tipo di istanza per ricevere aggiornamenti automatici sulla famiglia di istanze.

> **Note**
>
> - La disponibilità della famiglia di istanze varia a seconda Regione AWS. Per esempio, alcuni Regione AWS potrebbero non avere famiglie di istanze di quarta generazione ma avere famiglie di istanze di quinta e sesta generazione.
> - Quando si utilizzano i `default_x86_64` nostri pacchetti di `default_arm64` istanze, AWS Batch seleziona le famiglie di istanze in base a un equilibrio tra economicità e prestazioni. Sebbene le istanze di nuova generazione offrano spesso un miglior rapporto prezzo/prestazioni, è AWS Batch possibile scegliere una famiglia di istanze della generazione precedente se offre la combinazione ottimale di disponibilità, costi e prestazioni per il carico di lavoro. Ad esempio, in un Regione AWS paese in cui sono disponibili sia le istanze c6i che le istanze c7i, è AWS Batch possibile selezionare le istanze c6i se offrono un miglior rapporto qualità-prezzo per le specifiche esigenze lavorative. Per ulteriori informazioni sui tipi di istanze e sulla disponibilità, consulta Tabella di calcolo dei tipi di AWS Batch istanza. Regione AWS
> - AWS Batch aggiorna periodicamente le istanze nei pacchetti predefiniti con opzioni più nuove e più convenienti. Gli aggiornamenti avvengono automaticamente senza richiedere alcuna azione da parte dell'utente. I carichi di lavoro continuano a funzionare senza interruzioni durante gli aggiornamenti

> **Note**
>
> Quando crei un ambiente di calcolo, i tipi di istanza selezionati per l'ambiente di calcolo devono condividere la stessa architettura. Ad esempio, non puoi combinare istanze x86 e ARM nello stesso ambiente di calcolo.

> **Note**
>
> AWS Batch verrà GPUs scalato in base alla quantità richiesta nelle code di lavoro. Per utilizzare la pianificazione tramite GPU, l'ambiente di calcolo deve includere tipi di istanze appartenenti alle famigliep3,p4,p5,p6,g3, g3sg4, g5 o. g6

18. (Facoltativo) Espandi la configurazione aggiuntiva.

    a. (Facoltativo) Per Gruppo di collocamento, inserite il nome del gruppo di posizionamento per raggruppare le risorse nell'ambiente di calcolo.

    b. Per Strategia di allocazione, scegliete BEST_FIT_PROGRESSIVE.

    c. (Facoltativo) Per la configurazione di Amazon Machine Images (AMIs), scegli Aggiungi configurazione Amazon Machine Images (amis).

       Puoi utilizzare un'AMI Amazon Linux ottimizzata per Amazon EKS o un'AMI personalizzata.

       i. Per utilizzare un'AMI Amazon Linux ottimizzata per Amazon EKS:

          A. Per Tipo di immagine scegli una delle seguenti opzioni:

             - Amazon Linux 2: impostazione predefinita per tutte le famiglie di istanze AWS basate su Graviton (ad esempio, C6g M6gR6g, eT4g) e può essere utilizzato per tutti i tipi di istanze non GPU.
             - Amazon Linux 2 (accelerato): impostazione predefinita per tutte le famiglie di istanze GPU (ad esempio P4 eG4) e può essere utilizzato per tutti i tipi di istanze non basati su AWS Graviton.
             - Amazon Linux 2023: AWS Batch supporta Amazon Linux 2023 (AL2023).
             - Amazon Linux 2023 (accelerato): famiglie di istanze GPU e può essere utilizzato per tutti i tipi di istanze non basate su AWS Graviton.

          B. Per la Kubernetesversione, inserisci un numero di versione. Kubernetes

       ii. Per utilizzare un'AMI personalizzata:

          A. Per Tipo di immagine scegli il tipo di AMI su cui si basa l'AMI personalizzata:

- Amazon Linux 2: impostazione predefinita per tutte le famiglie di istanze AWS basate su Graviton (ad esempio, `C6g` `M6gR6g`, `eT4g`) e può essere utilizzato per tutti i tipi di istanze non GPU.
- Amazon Linux 2 (accelerato): impostazione predefinita per tutte le famiglie di istanze GPU (ad esempio `P4` `eG4`) e può essere utilizzato per tutti i tipi di istanze non basati su AWS Graviton.
- Amazon Linux 2023: AWS Batch supporta AL2 023.
- Amazon Linux 2023 (accelerato): famiglie di istanze GPU e può essere utilizzato per tutti i tipi di istanze non basate su AWS Graviton.

B. Per Image ID override, inserisci l'ID AMI personalizzato.

C. Per la Kubernetesversione, inserisci un numero di Kubernetes versione.

d. (Facoltativo) Per il modello di lancio, scegli un modello di lancio esistente.

e. (Facoltativo) Per la versione del modello Launch **`$Default$Latest`**, inserisci o un numero di versione.

f. (Facoltativo) Per Sostituisci modello di lancio, per aggiungere un override scegli Aggiungi sostituisci modello di lancio:

i. (Facoltativo) Per Launch template, scegli il modello di lancio a cui aggiungere l'override.

ii. (Facoltativo) Per la versione del modello di lancio, `$Default` scegli il numero di versione del modello di lancio oppure`$Latest`.

iii. (Facoltativo) Per i tipi di istanze Target, scegli il tipo o la famiglia di istanze a cui applicare l'override. Questo può riguardare solo i tipi e le famiglie di istanze inclusi nei tipi di istanze consentiti.

iv. (Facoltativo) Per UserDataType scegliete l'inizializzazione del nodo EKS. Utilizza questo campo solo se hai un AMI specificato nel Launch Template o come Launch Template Override. Scegliete EKS_NODEADM per personalizzazione in AMIs base a or oppure EKS_BOOSTRAP_SH per and**`EKS_AL2023`**. **`EKS_AL2023_NVIDIA`** `EKS_AL2` `EKS_AL_NVIDIA` Il valore predefinito è EKS_BOOSTRAP_SH.

Utilizzerai UserDataType in un ambiente misto in cui utilizzi entrambi i metodi personalizzati basati su 023 nello stesso ambiente di calcolo. AL2 AL2 AMIs

19. Scegli Pagina successiva.
20. Per l'ID Virtual Private Cloud (VPC), scegli un VPC in cui avviare le istanze.

21. Per le sottoreti, scegli le sottoreti da utilizzare. Per impostazione predefinita, sono disponibili tutte le sottoreti all'interno del VPC selezionato.

> **Note**
>
> AWS Batch su Amazon EKS supporta Local Zones. Per ulteriori informazioni, consulta Amazon EKS and AWS Local Zones nella Amazon EKS User Guide.

22. (Facoltativo) Per i gruppi di sicurezza, scegli un gruppo di sicurezza da collegare alle tue istanze. Per impostazione predefinita, è selezionato il gruppo di sicurezza predefinito per il tuo VPC.
23. Scegli Pagina successiva.
24. Per Revisione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea ambiente di calcolo.

Risorsa: modello di ambiente di calcolo

L'esempio seguente mostra un modello di ambiente di calcolo vuoto. È possibile utilizzare questo modello per creare l'ambiente di calcolo che può quindi essere salvato in un file e utilizzato con l' AWS CLI `--cli-input-json`opzione. Per ulteriori informazioni su questi parametri, consulta CreateComputeEnvironmentl'AWS Batch API Reference.

> **Note**
>
> È possibile generare un modello di ambiente di calcolo con il seguente AWS CLI comando.
>
> ```
> $ aws batch create-compute-environment --generate-cli-skeleton
> ```

```
{
    "computeEnvironmentName": "",
    "type": "UNMANAGED",
    "state": "DISABLED",
    "unmanagedvCpus": 0,
    "computeResources": {
        "type": "EC2",
        "allocationStrategy": "BEST_FIT_PROGRESSIVE",
        "minvCpus": 0,
        "maxvCpus": 0,
```

```
        "desiredvCpus": 0,
        "instanceTypes": [
            ""
        ],
        "imageId": "",
        "subnets": [
            ""
        ],
        "securityGroupIds": [
            ""
        ],
        "ec2KeyPair": "",
        "instanceRole": "",
        "tags": {
            "KeyName": ""
        },
        "placementGroup": "",
        "bidPercentage": 0,
        "spotIamFleetRole": "",
        "launchTemplate": {
            "launchTemplateId": "",
            "launchTemplateName": "",
            "version": ""
        },
        "ec2Configuration": [
            {
                "imageType": "",
                "imageIdOverride": "",
                "imageKubernetesVersion": ""
            }
        ]
    },
    "serviceRole": "",
    "tags": {
        "KeyName": ""
    },
    "eksConfiguration": {
        "eksClusterArn": "",
        "kubernetesNamespace": ""
    }
}
```

Tabella di calcolo del tipo di istanza

La tabella seguente elenca la Regione AWS parola chiave della famiglia di istanze e le famiglie di istanze disponibili. AWS Batch cercherà di allocare un'istanza della famiglia di istanze più recente, ma poiché la disponibilità della famiglia di istanze varia a seconda dell'istanza, Regione AWS è possibile che venga generata una famiglia di istanze precedente.

default_x86_64

Regione	Famiglie di istanze
Tutto questo è supporto Regione AWSAWS Batch	m6i, c6i, r6i c7i

default_arm64

Regione	Famiglie di istanze
Tutto questo è supporto Regione AWSAWS Batch	6 mg, 6 g, 6 g 7 g

Ottimale

Regione	Famiglie di istanze
• ap-northeast-1 • ap-northeast-2 • ap-south-1 • ap-southeast-1 • ap-southeast-2 • ca-central-1 • cn-north-1 • cn-northwest-1 • eu-central-1 • eu-west-1	m4, c4, r4

Regione	Famiglie di istanze
• eu-west-2 • sa-east-1 • us-east-1 • us-east-2 • us-gov-west-1 • us-west-1 • us-west-2	

Regione	Famiglie di istanze
• af-south-1 • ap-east-1 • ap-northeast-3 • ap-south-2 • ap-southeast-3 • ap-southeast-4 • ca-west-1 • eu-central-2 • eu-north-1 • eu-south-1 • eu-south-2 • eu-west-3 • il-central-1 • me-central-1 • me-south-1 • us-gov-east-1 • us-isob-east-1 • us-iso-east-1 • us-isof-south-1 • us-isof-east-1 • eu-isoe-west-1 • us-northeast-1	m5, 5c, 5r
• ap-southeast-5 • ap-southeast-7 • ap-east-2 • mx-central-1	m6, c6, r6

Aggiornare un ambiente di calcolo in AWS Batch

AWS Batch offre diverse strategie per l'aggiornamento degli ambienti di elaborazione, ognuna progettata per scenari e requisiti di aggiornamento specifici. Questi approcci utilizzano la stessa API di aggiornamento sottostante ma rappresentano metodi prescrittivi diversi per gestire gli aggiornamenti in modo efficace. È possibile gestire questi aggiornamenti utilizzando la AWS Batch console o il. AWS CLI La comprensione di queste strategie consente di scegliere il metodo più appropriato per le proprie esigenze, riducendo al minimo le interruzioni dei carichi di lavoro.

Questo argomento fornisce una panoramica delle strategie di aggiornamento disponibili e indicazioni su quando utilizzare ciascun approccio. Per le procedure dettagliate, consulta le singole sezioni relative a ciascuna strategia di aggiornamento.

> **Important**
>
> AWS Batch crea e gestisce più AWS risorse per tuo conto e all'interno del tuo account, tra cui Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon EC2 Spot Fleets e Amazon ECS Clusters. Queste risorse gestite sono configurate specificamente per garantire un funzionamento ottimale. AWS Batch La modifica manuale di queste risorse AWS Batch gestite, a meno che non sia esplicitamente indicato nella AWS Batch documentazione, può comportare comportamenti imprevisti, tra cui ambienti di `INVALID` calcolo, un comportamento di scalabilità delle istanze non ottimale, un ritardo nell'elaborazione del carico di lavoro o costi imprevisti. Queste modifiche manuali non possono essere supportate in modo deterministico dal servizio. AWS Batch Usa sempre il supporto AWS Batch APIs o la AWS Batch console per gestire i tuoi ambienti di elaborazione.

Argomenti

Strategie di aggiornamento dell'ambiente di calcolo

Quando utilizzi la scalabilità o gli aggiornamenti dell'infrastruttura, l'ambiente di elaborazione viene aggiornato sul posto. Per la strategia di blue/green aggiornamento, stai creando un nuovo ambiente di calcolo (verde) e quindi migrando il carico di lavoro dal vecchio ambiente di calcolo (blu) al nuovo ambiente di calcolo (verde).

AWS Batch offre tre diverse strategie per gli aggiornamenti dell'ambiente di calcolo:

Aggiornamenti di scalabilità

Gli aggiornamenti di scalabilità regolano la capacità dell'ambiente di calcolo aggiungendo o rimuovendo istanze senza sostituire le istanze esistenti. Questo è lo scenario di aggiornamento più rapido e non richiede tempi di inattività. Utilizza gli aggiornamenti di scalabilità quando devi modificare le impostazioni di capacità (vCPUs). Questi aggiornamenti in genere vengono completati in pochi minuti.

Gli aggiornamenti Fargate vengono eseguiti utilizzando le stesse procedure degli aggiornamenti in scala. Per ulteriori informazioni, consulta Esegui aggiornamenti di scalabilità.

Aggiornamenti dell'infrastruttura

Gli aggiornamenti dell'infrastruttura sostituiscono le istanze nell'ambiente di elaborazione con nuove istanze con impostazioni aggiornate. Questi aggiornamenti richiedono configurazioni specifiche del ruolo di servizio e della strategia di allocazione, ma garantiscono tempi di inattività minimi, con processi in esecuzione potenzialmente interrotti. Utilizza gli aggiornamenti dell'infrastruttura quando devi modificare i tipi di istanza, la configurazione AMI, le impostazioni di rete, il ruolo del servizio, lo stato dell'ambiente o altri componenti dell'infrastruttura. Questi aggiornamenti vengono generalmente completati in 10-30 minuti a seconda del completamento del processo.

Per ulteriori informazioni, consulta Esegui aggiornamenti dell'infrastruttura.

Aggiornamenti blu/verdi

Blue/green updates create a new compute environment alongside your existing environment, allowing gradual workload transition with zero downtime. This approach provides the safest update path but requires running two environments temporarily. Use blue/greenaggiornamenti quando non sono necessari tempi di inattività, si desidera testare le modifiche prima della distribuzione completa, si richiede una funzionalità di rollback rapido o si utilizzano configurazioni

non supportate per gli aggiornamenti dell'infrastruttura. Il tempo di completamento è variabile e controllato dall'utente.

Per ulteriori informazioni, consulta Esegui blue/green aggiornamenti per gli ambienti di elaborazione.

Scelta della giusta strategia di aggiornamento

Utilizza questa guida decisionale per selezionare la strategia di aggiornamento più adatta alle tue esigenze:

Scegli la scalabilità degli aggiornamenti quando

Scegli la strategia di aggiornamento della scalabilità quando devi solo regolare la capacità di elaborazione (v). CPUs Gli aggiornamenti di scalabilità sono ideali quando sono necessari aggiornamenti rapidi senza tempi di inattività e non sono necessarie modifiche alla configurazione dell'infrastruttura.

Per le procedure dettagliate, consulta Esegui aggiornamenti di scalabilità.

Scegli gli aggiornamenti dell'infrastruttura quando

Scegli la strategia di aggiornamento dell'infrastruttura quando devi modificare i tipi di istanza, le impostazioni AMI, il ruolo del servizio, lo stato dell'ambiente o la configurazione di rete. L'ambiente deve utilizzare il ruolo AWSServiceRoleForBatchcollegato al servizio e una strategia di allocazione di`BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o. `SPOT_PRICE_CAPACITY_OPTIMIZED` Gli aggiornamenti dell'infrastruttura funzionano bene quando alcune interruzioni del lavoro sono accettabili durante l'aggiornamento e desideri aggiornamenti automatici all'ultima AMI ottimizzata per Amazon ECS.

Per le procedure dettagliate, consulta Esegui aggiornamenti dell'infrastruttura.

Scegli gli aggiornamenti quando blue/green

Scegli la strategia di blue/green aggiornamento quando non sono necessari tempi di inattività per i tuoi carichi di lavoro o devi testare le modifiche prima di passare ai carichi di lavoro di produzione. Questo approccio è essenziale quando è importante una capacità di rollback rapido, l'ambiente utilizza una strategia di `BEST_FIT` allocazione o non utilizza il ruolo collegato al servizio. AWSServiceRoleForBatch Blue/green gli aggiornamenti sono anche la scelta migliore quando

utilizzi soluzioni personalizzate AMIs che richiedono aggiornamenti manuali o richiedono modifiche importanti alla configurazione.

Per le procedure dettagliate, consulta Esegui blue/green aggiornamenti per gli ambienti di elaborazione.

Considerazioni sull'aggiornamento delle AMI

AWS Batch può eseguire l'aggiornamento all'ultima AMI ottimizzata per Amazon ECS durante gli aggiornamenti dell'infrastruttura quando tutte queste condizioni sono soddisfatte:

> **Note**
>
> Al termine dell'aggiornamento dell'infrastruttura `updateToLatestImageVersion` è impostato su. `false` Per avviare un altro aggiornamento `updateToLatestImageVersion` è necessario impostare `true` su.

- L'ambiente di calcolo utilizza il ruolo collegato al servizio AWSServiceRoleForBatch
- La strategia di allocazione è impostata su, o `BEST_FIT_PROGRESSIVE` `SPOT_CAPACITY_OPTIMIZED` `SPOT_PRICE_CAPACITY_OPTIMIZED`
- Nessun ID AMI è specificato esplicitamente in `imageIdimageIdOverride`, o nel modello di avvio
- `updateToLatestImageVersion`È impostato su `true`

Aggiornamenti AMI tramite blue/green distribuzione

È necessario utilizzare la blue/green distribuzione per l'aggiornamento AMIs in questi scenari:

- Quando si utilizza una versione specifica dell'AMI ottimizzata per Amazon ECS
- Quando l'ID AMI è specificato in uno dei seguenti campi:
  - Avvia il modello (è necessario aggiornare il modello o rimuoverlo)
  - Parametro `imageId`
  - Il `imageIdOverride` parametro in EC2 configurazione
- Quando si utilizza la strategia `BEST_FIT` di allocazione (non supporta gli aggiornamenti dell'infrastruttura)
- Quando non si utilizza il ruolo collegato al AWSServiceRoleForBatchservizio

Esegui aggiornamenti di scalabilità

Gli aggiornamenti di scalabilità regolano la capacità dell'ambiente di calcolo aggiungendo o rimuovendo istanze. Questa è la strategia di aggiornamento più rapida e non richiede la sostituzione delle istanze esistenti. Gli aggiornamenti scalabili funzionano con qualsiasi tipo di ruolo di servizio e strategia di allocazione, il che li rende l'opzione di aggiornamento più flessibile.

Modifiche che attivano un aggiornamento scalabile

Quando si modificano solo le seguenti impostazioni, AWS Batch esegue un aggiornamento di ridimensionamento. Se modifichi una di queste impostazioni insieme ad altre impostazioni dell'ambiente di calcolo, AWS Batch esegue invece un aggiornamento dell'infrastruttura.

Le seguenti impostazioni attivano gli aggiornamenti di ridimensionamento quando vengono modificate esclusivamente:

- `desiredvCpus`— Imposta il numero target di v CPUs per l'ambiente.
- `maxvCpus`— Definisce il numero massimo di v CPUs che possono essere lanciati.
- `minvCpus`— Speciifica il numero minimo di v CPUs da mantenere.

Per gli ambienti di calcolo Fargate, puoi anche modificare queste impostazioni per scalare gli aggiornamenti:

- `securityGroupIds`— Gruppo di sicurezza IDs per l'ambiente di calcolo.
- `subnets`— Sottoreti per l'ambiente di calcolo.

> **Note**
>
> Si consiglia di non `desiredvCpus` utilizzarlo per avviare un aggiornamento di scalabilità poiché si adatterà dinamicamente. AWS Batch `desiredvCpus` Dovresti invece effettuare l'aggiornamento. `minvCpus`
> Durante l'aggiornamento`desiredvCpus`, il valore deve essere compreso tra `minvCpus` e`maxvCpus`. Il nuovo valore deve essere maggiore o uguale a quello corrente`desiredvCpus`. Per ulteriori informazioni, consulta the section called “Messaggio di errore quando si aggiorna l'`desiredvCpus`impostazione”.

> **Important**
>
> Se modifichi una di queste impostazioni di ridimensionamento insieme ad altre impostazioni dell'ambiente di calcolo (come tipi di istanze IDs, AMI o modelli di avvio), AWS Batch esegue un aggiornamento dell'infrastruttura anziché un aggiornamento di scalabilità. Gli aggiornamenti dell'infrastruttura richiedono più tempo e possono sostituire le istanze esistenti.

Performing scaling updates using the Console di gestione AWS

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel pannello di navigazione, scegli Ambienti, quindi la scheda Ambienti di calcolo.
3. Seleziona l'ambiente di calcolo da aggiornare.
4. Scegli Azioni e poi Modifica.
5. Modifica una o più impostazioni che supportano gli aggiornamenti di ridimensionamento. Ad esempio:
   - Per Minimo v CPUs, inserisci il numero minimo di vCPUs.
   - Per Desired v CPUs, inserire il numero desiderato di vCPUs.
   - Per Maximum v CPUs, inserire il numero massimo di vCPUs.
6. Scegli Save changes (Salva modifiche).
7. Monitora lo stato dell'ambiente di elaborazione. L'aggiornamento dovrebbe essere completato rapidamente poiché prevede solo operazioni di scalabilità.

Performing scaling updates using the AWS CLI

Utilizzate il update-compute-environment comando per eseguire gli aggiornamenti di ridimensionamento. I due esempi seguenti illustrano le operazioni di ridimensionamento comuni. È possibile modificare una o più delle seguenti impostazioni che supportano gli aggiornamenti di ridimensionamento

- Questo esempio aggiorna il valore v desiderato, minimo e massimo: CPUs

```
aws batch update-compute-environment \
    --compute-environment your-compute-environment-name \
    --compute-resources minvCpus=2,maxvCpus=8
```

Monitoraggio degli aggiornamenti di scalabilità

Monitora gli aggiornamenti di scalabilità utilizzando la AWS Batch console per visualizzare lo stato dell'ambiente di calcolo e controllare il numero di istanze e le metriche della vCPU. È inoltre possibile utilizzare il describe-compute-environments comando AWS CLI with the per controllare lo stato e monitorare il numero di istanze e i valori della vCPU.

Esegui aggiornamenti dell'infrastruttura

Gli aggiornamenti dell'infrastruttura sostituiscono le istanze dell'ambiente di calcolo con nuove istanze con impostazioni aggiornate. Questa strategia di aggiornamento richiede più tempo rispetto alla scalabilità degli aggiornamenti e richiede impostazioni specifiche del ruolo di servizio e della strategia di allocazione. Gli aggiornamenti dell'infrastruttura forniscono un modo per modificare le configurazioni fondamentali dell'ambiente di calcolo mantenendo al contempo la disponibilità del servizio.

> **Important**
>
> Gli aggiornamenti dell'infrastruttura richiedono il ruolo AWSServiceRoleForBatchcollegato al servizio e una strategia di allocazione di, o. `BEST_FIT_PROGRESSIVE` `SPOT_CAPACITY_OPTIMIZED` `SPOT_PRICE_CAPACITY_OPTIMIZED` Se il tuo ambiente non soddisfa questi requisiti, utilizza blue/green invece gli aggiornamenti.

Modifiche che attivano gli aggiornamenti dell'infrastruttura

Quando si modifica una delle seguenti impostazioni, AWS Batch esegue un aggiornamento dell'infrastruttura. Gli aggiornamenti dell'infrastruttura si verificano anche quando si modificano queste impostazioni insieme al ridimensionamento delle impostazioni di aggiornamento.

Le seguenti impostazioni attivano gli aggiornamenti dell'infrastruttura:

Configurazione di calcolo

- `allocationStrategy`— Determina il modo in cui AWS Batch vengono selezionati i tipi di istanza.
- `instanceTypes`— Speciifica i tipi di EC2 istanza da utilizzare.
- `bidPercentage`— Percentuale massima del prezzo on demand per le istanze Spot.
- `type`— Tipo di ambiente di calcolo (`EC2`o). `SPOT`

AMI e configurazione di avvio

- `imageId`— AMI specifica da utilizzare per le istanze.
- `ec2Configuration`— EC2 configurazione inclusa`imageIdOverride`.
- `launchTemplate`— EC2 avvia le impostazioni del modello.
- `ec2KeyPair`— Coppia di key pair SSH per l'accesso ad esempio.
- `updateToLatestImageVersion`— Impostazione degli aggiornamenti AMI automatici.

Rete e sicurezza

- `subnets`— Sottoreti VPC in cui vengono lanciate le istanze (per ambienti di elaborazione). EC2
- `securityGroupIds`— Gruppi di sicurezza per istanze (per ambienti di elaborazione). EC2
- `placementGroup`— configurazione del gruppo di EC2 collocamento.

Altre impostazioni

- `instanceRole`— Ruolo IAM per le EC2 istanze.
- `tags`— Tag applicati alle EC2 istanze.

> **Important**
>
> Se si modificano le impostazioni di aggiornamento dell'infrastruttura insieme al ridimensionamento delle impostazioni di aggiornamento (ad esempio `desiredvCpus`, `maxvCpus`, o`minvCpus`), AWS Batch esegue un aggiornamento dell'infrastruttura. Gli aggiornamenti dell'infrastruttura richiedono più tempo rispetto agli aggiornamenti di scalabilità.

Selezione dell'AMI durante gli aggiornamenti dell'infrastruttura

Durante un aggiornamento dell'infrastruttura, l'ID AMI dell'ambiente di calcolo potrebbe cambiare, a seconda che sia AMIs specificato in una di queste tre impostazioni. AMIs sono specificati nel modello `imageId` (in`computeResources`), `imageIdOverride` (in`ec2Configuration`) o nel modello di avvio specificato in`launchTemplate`. Supponiamo che nessun AMI IDs sia specificato in nessuna di queste impostazioni e che l'`updateToLatestImageVersion`impostazione sia`true`. Quindi,

l'ultima AMI ottimizzata per Amazon ECS supportata da AWS Batch viene utilizzata per qualsiasi aggiornamento dell'infrastruttura.

Se in almeno una di queste impostazioni è specificato un ID AMI, l'aggiornamento dipende dall'impostazione che ha fornito l'ID AMI utilizzato prima dell'aggiornamento. Quando crei un ambiente di calcolo, la priorità per la selezione di un ID AMI è prima il modello di avvio, poi l'`imageId`impostazione e infine l'`imageId0verride`impostazione. Tuttavia, se l'ID AMI utilizzato proviene dal modello di avvio, l'aggiornamento `imageId0verride` delle impostazioni `imageId` o non aggiorna l'ID AMI. L'unico modo per aggiornare un ID AMI selezionato dal modello di avvio è aggiornare il modello di avvio. Se il parametro della versione del modello di lancio è `$Default` o`$Latest`, viene valutata la versione predefinita o più recente del modello di lancio specificato. Se per impostazione predefinita è selezionato un ID AMI diverso o è selezionata la versione più recente del modello di avvio, tale ID AMI viene utilizzato nell'aggiornamento.

Se il modello di avvio non è stato utilizzato per selezionare l'ID AMI, viene utilizzato l'ID AMI specificato nei `imageId0verride` parametri `imageId` o. Se vengono specificati entrambi, viene utilizzato l'ID AMI specificato nel `imageId0verride` parametro.

Supponiamo che l'ambiente di calcolo utilizzi un ID AMI specificato dai `launchTemplate` parametri `imageIdimageId0verride`, o e che desideri utilizzare l'AMI ottimizzata Amazon ECS più recente supportata da. AWS Batch Quindi, l'aggiornamento deve rimuovere le impostazioni che hanno fornito l'AMI IDs. Ciò richiede infatti la specificazione di una stringa vuota per quel parametro. `imageId` Perché`imageId0verride`, ciò richiede la specificazione di una stringa vuota per il `ec2Configuration` parametro.

Se l'ID AMI proviene dal modello di avvio, puoi passare alla più recente AMI ottimizzata per Amazon ECS AWS Batch supportata in uno dei seguenti modi:

- Rimuovi il modello di avvio specificando una stringa vuota per il parametro `launchTemplateId` or`launchTemplateName`. Ciò rimuove l'intero modello di avvio, anziché il solo ID AMI.
- Se la versione aggiornata del modello di avvio non specifica un ID AMI, il `updateToLatestImageVersion` parametro deve essere impostato su`true`.

Gestione dei job durante gli aggiornamenti

Configura come vengono gestiti i lavori in esecuzione durante un aggiornamento dell'infrastruttura utilizzando la politica di aggiornamento. Quando si impostaterminateJobs0nUpdate=`true`, i processi in esecuzione vengono interrotti

immediatamente, l'`jobExecutionTimeoutMinutes`impostazione viene ignorata e l'aggiornamento procede non appena le istanze possono essere sostituite. Quando si imposta`terminateJobsOnUpdate=false`, i lavori in esecuzione continuano per il periodo di timeout specificato con un timeout predefinito di 30 minuti e i lavori vengono interrotti se superano il timeout.

> **Note**
>
> Per riprovare i lavori che vengono interrotti durante un aggiornamento, configura una strategia di nuovo tentativo. Per ulteriori informazioni, consulta the section called “Ritentativi di lavoro automatizzati”.

Performing infrastructure updates using the Console di gestione AWS

1. Apri la console all' AWS Batch indirizzo. https://console.aws.amazon.com/batch/
2. Nel pannello di navigazione, scegli Ambienti, quindi la scheda Ambienti di calcolo.
3. Seleziona l'ambiente di calcolo da aggiornare.
4. Scegli Azioni e poi Modifica.
5. Nella sezione Aggiorna comportamento, configura la modalità di gestione dei processi in esecuzione:
   - Scegli Aggiorna AMI alla versione più recente per aggiornare l'AMI alla versione più recente.
   - Scegli Termina i lavori immediatamente all'aggiornamento per terminare i lavori quando viene eseguito il processo di aggiornamento.
   - Per Job execution timeout, immettete il numero di minuti da attendere prima di iniziare il processo di aggiornamento.
6. Modifica una o più impostazioni che richiedono aggiornamenti dell'infrastruttura. Ad esempio:
   - Ruolo dell'istanza
   - Usa le istanze EC2 Spot
   - Tipi di istanze consentiti
   - Gruppo di collocamento
   - EC2 coppia di chiavi
   - EC2 configurazione

- Modelli di lancio
- Sottoreti
- Gruppi di sicurezza

7. Scegli Save changes (Salva modifiche).
8. Monitora lo stato dell'ambiente di calcolo. L'ambiente verrà visualizzato `UPDATING` durante il processo di aggiornamento.

Performing infrastructure updates using the AWS CLI

Utilizza il update-compute-environment comando per modificare una o più impostazioni che richiedono aggiornamenti dell'infrastruttura. I tre esempi seguenti sono operazioni infrastrutturali comuni.

- Questo esempio aggiorna i tipi di istanza e configura la politica di aggiornamento:

```
aws batch update-compute-environment \
    --compute-environment your-compute-environment-name \
    --compute-resources instanceTypes=default_x86_64 \
    --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=30
```

- Questo esempio aggiorna le sottoreti e i gruppi di sicurezza VPC:

```
aws batch update-compute-environment \
    --compute-environment your-compute-environment-name \
    --compute-resources subnets=subnet-abcd1234,subnet-efgh5678
 securityGroupIds=sg-abcd1234 \
    --update-policy terminateJobsOnUpdate=true
```

- Questo esempio abilita gli aggiornamenti automatici all'ultima AMI ottimizzata per Amazon ECS:

```
aws batch update-compute-environment \
    --compute-environment your-compute-environment-name \
    --compute-resources updateToLatestImageVersion=true \
    --update-policy terminateJobsOnUpdate=false,jobExecutionTimeoutMinutes=60
```

Monitoraggio degli aggiornamenti dell'infrastruttura

Monitora gli aggiornamenti dell'infrastruttura utilizzando la AWS Batch console per osservare la modifica dello stato dell'ambiente di calcoloUPDATING, monitorare l'avanzamento della sostituzione delle istanze e verificare la presenza di eventuali aggiornamenti non riusciti. L'aggiornamento ha esito positivo una volta raggiunto lo stato dell'ambiente di calcolo. VAILD È inoltre possibile utilizzarlo CloudWatch per tenere traccia degli eventi di chiusura dell'istanza e monitorare lo stato del processo durante l'aggiornamento. Con AWS CLI, utilizzate il describe-compute-environments comando per controllare lo stato e monitorare gli eventi del ciclo di vita dell'istanza.

Esegui blue/green aggiornamenti per gli ambienti di elaborazione

Un blue/green aggiornamento è una strategia di aggiornamento che riduce i tempi di inattività e i rischi creando un nuovo ambiente di elaborazione (verde) accanto all'ambiente di elaborazione esistente (blu). Questo approccio consente di trasferire gradualmente i carichi di lavoro al nuovo ambiente mantenendo operativo l'ambiente esistente. Blue/green gli aggiornamenti forniscono il percorso di aggiornamento più sicuro e funzionano con qualsiasi tipo di ruolo di servizio o strategia di allocazione.

Panoramica

Gli aggiornamenti blu/verdi offrono diversi vantaggi che li rendono ideali per gli ambienti di produzione. Garantiscono zero tempi di inattività mantenendo i carichi di lavoro in esecuzione ininterrottamente durante il processo di aggiornamento. L'approccio consente semplici funzionalità di rollback, che consentono di ripristinare rapidamente l'ambiente originale in caso di problemi. È possibile implementare una strategia di transizione graduale, verificando le prestazioni del nuovo ambiente prima di passare completamente ai carichi di lavoro di produzione. Questo metodo offre anche un'eccellente mitigazione del rischio, poiché l'ambiente originale rimane invariato e operativo fino a quando non si sceglie di rimuoverlo.

Quando sono necessari blue/green aggiornamenti

È necessario utilizzare blue/green gli aggiornamenti nelle seguenti situazioni:

- Quando l'ambiente di elaborazione utilizza la strategia di BEST_FIT allocazione (non supporta gli aggiornamenti dell'infrastruttura)
- Quando l'ambiente di elaborazione non utilizza il ruolo collegato ai servizi AWSServiceRoleForBatch

- Quando è necessario passare da un tipo di ruolo di servizio all'altro

Quando sono consigliati blue/green gli aggiornamenti

Blue/green updates are particularly recommended for production environments where zero downtime is critical for your workloads. This approach works well when you need to test new configurations before transitioning production workloads, ensuring that changes meet your performance and reliability requirements. Choose blue/greenaggiornamenti quando la funzionalità di rollback rapido è importante per le tue operazioni, specialmente se esegui aggiornamenti personalizzati AMIs con modifiche significative. Questo metodo è ideale anche quando si desidera convalidare le caratteristiche e il comportamento delle prestazioni prima di impegnarsi completamente nelle modifiche, in modo da avere fiducia nel processo di aggiornamento.

Prerequisiti

Prima di eseguire un blue/green aggiornamento, assicuratevi di avere:

- Autorizzazioni IAM appropriate per creare e gestire ambienti di elaborazione
- Accesso per visualizzare e modificare le impostazioni della coda di lavoro
- Strategie di rielaborazione dei lavori configurate per le definizioni dei lavori per gestire potenziali errori durante la transizione. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.
- L'ID AMI per il nuovo ambiente di calcolo. Questo può essere uno dei seguenti:
  - Una versione recente e approvata dell'AMI ottimizzata Amazon ECS (utilizzata per impostazione predefinita)
  - Un'AMI personalizzata che soddisfa le specifiche AMI dell'istanza di container Amazon ECS. Quando si utilizza un'AMI personalizzata, è possibile specificarla in uno dei seguenti modi:
    - Utilizzo del campo Image ID override nella configurazione EC2
    - Specificandolo in un modello di lancio

    Per ulteriori informazioni sulla creazione di contenuti personalizzati AMIs, consultaTutorial: Creare un'AMI per risorse di calcolo.

Prima di creare il nuovo ambiente, è necessario registrare la configurazione dell'ambiente di elaborazione esistente. È possibile eseguire questa operazione utilizzando il Console di gestione AWS o il AWS CLI.

> **Note**
>
> Le seguenti procedure descrivono in dettaglio come eseguire un blue/green aggiornamento che modifichi solo l'AMI. È possibile aggiornare altre impostazioni per il nuovo ambiente.

> **Important**
>
> Quando rimuovi il vecchio ambiente di calcolo (blu), tutti i processi attualmente in esecuzione su tali istanze falliranno perché le istanze verranno terminate. Configura le strategie di ripetizione dei job nelle definizioni dei job per gestire automaticamente questi errori. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.
> Una volta acquisita fiducia nel nuovo ambiente:
>
> 1. Modifica la coda dei lavori per rimuovere il vecchio ambiente di calcolo.
> 2. Attendi il completamento di tutti i processi in esecuzione nel vecchio ambiente.
> 3. Eliminare il vecchio ambiente di calcolo.

Performing blue/green updates using the Console di gestione AWS

1. Clona il tuo ambiente di elaborazione attuale
   a. Apri la AWS Batch console all'indirizzo. https://console.aws.amazon.com/batch/
   b. Seleziona l'ambiente di calcolo esistente.
   c. Scegli Azioni e poi Clona.
   d. In Nome, inserisci un nome univoco per il tuo nuovo ambiente di calcolo.
   e. Scegli Next (Successivo).
   f. Nella sezione Configurazione dell'istanza, aggiorna le impostazioni AMI:
      i. Espandere Additional configuration (Configurazione aggiuntiva).
      ii. Per la EC2 configurazione, specifica il nuovo tipo di AMI in Tipo di immagine e ID AMI nel campo Image ID override.
   g. Scegli Next (Successivo).
   h. Per la configurazione di rete, scegli Avanti.

i. Controlla le altre impostazioni che vengono copiate automaticamente dall'ambiente esistente.

j. Scegli Crea ambiente di calcolo.

k. Attendi che diventi lo stato del nuovo ambiente di calcolo. `VALID`

2. Modifica l'ordine della coda dei lavori

a. Nel riquadro di navigazione, scegli Job queues.

b. Seleziona la coda di lavoro associata all'ambiente di calcolo esistente.

c. Scegli Modifica.

d. In Ambiente Connected Compute, aggiungi il nuovo ambiente di calcolo:

- Aggiungi il nuovo ambiente di elaborazione con un numero d'ordine più alto rispetto all'ambiente esistente per trasferire il carico di lavoro.
- Dopo aver verificato che il nuovo ambiente funzioni correttamente, puoi renderlo l'ambiente principale assegnandogli un numero d'ordine inferiore.

e. Scegli Aggiorna coda lavori.

3. Eliminazione

a. Monitora l'esecuzione dei lavori nel nuovo ambiente per assicurarti che tutto funzioni come previsto.

b. Una volta acquisita fiducia nel nuovo ambiente:

1. Modifica la coda dei lavori per rimuovere il vecchio ambiente di calcolo.
2. Attendi il completamento di tutti i processi in esecuzione nel vecchio ambiente.
3. Eliminare il vecchio ambiente di calcolo.

Performing blue/green updates using the AWS CLI

1. Per ottenere la configurazione utilizzando il AWS CLI, utilizzate il seguente comando:

```
aws batch describe-compute-environments \
  --compute-environments your-compute-environment-name
```

Salvate l'output come riferimento durante la creazione del nuovo ambiente.

2. Crea un nuovo ambiente di calcolo utilizzando la configurazione dell'ambiente esistente, ma con la nuova AMI. Ecco un esempio di struttura di comando:

   Sostituisci i valori di esempio con la configurazione effettiva del passaggio precedente:

```
cat <<EOF > ./blue-green-compute-environment.json
{
  "computeEnvironmentName": "your-new-compute-environment-name",
  "type": "MANAGED",
  "state": "ENABLED",
  "computeResources": {
    "instanceRole": "arn:aws:iam::012345678901:instance-profile/
ecsInstanceRole",
    "type": "EC2",
    "minvCpus": 2,
    "desiredvCpus": 2,
    "maxvCpus": 256,
    "instanceTypes": [
      "optimal"
    ],
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "ec2Configuration": [
      {
        "imageType": "ECS_AL2023",
        "imageIdOverride": "ami-0abcdef1234567890"
      }
    ],
    "subnets": [,
      "subnet-0abcdef1234567890"
    ],
    "securityGroupIds": [
      "sg-0abcdef1234567890"
    ]
  }
}
EOF
```

```
$ aws batch create-compute-environment --cli-input-json file://./blue-green-
compute-environment.json
```

3. Attendi che il nuovo ambiente diventi disponibile:

```
aws batch describe-compute-environments \
  --compute-environments your-new-compute-environment-name \
  --query 'computeEnvironments[].status'
```

4. Aggiungi il nuovo ambiente di elaborazione alla tua coda di lavoro:

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-existing-
environment \
  order=2,computeEnvironment=your-new-compute-environment-name
```

5. Una volta verificato, esegui nuovamente l'aggiornamento per rendere principale il nuovo ambiente:

```
aws batch update-job-queue \
  --job-queue your-job-queue \
  --compute-environment-order order=1,computeEnvironment=your-new-compute-
environment-name
```

Una volta completati tutti i job nel vecchio ambiente, disattivatelo e poi cancellatelo:

```
aws batch update-compute-environment \
    --compute-environment your-existing-environment \
    --state DISABLED
```

```
aws batch delete-compute-environment \
  --compute-environment your-existing-environment
```

Risorsa di calcolo AMIs

Per impostazione predefinita, gli ambienti di elaborazione AWS Batch gestiti utilizzano una versione recente e approvata dell'AMI ottimizzata Amazon ECS per le risorse di calcolo. Tuttavia, potresti voler creare la tua AMI da utilizzare per i tuoi ambienti di elaborazione gestiti e non gestiti. Se hai bisogno di uno dei seguenti elementi, ti consigliamo di creare la tua AMI personale:

- Aumento delle dimensioni di archiviazione della radice o dei volumi di dati dell'AMI
- Aggiungere volumi di storage delle istanze per i tipi di EC2 istanze Amazon supportati

- Personalizzazione dell'agente container Amazon ECS
- Personalizzazione di Docker
- Configurazione di un'AMI per carichi di lavoro GPU per consentire ai container di accedere all'hardware GPU sui tipi di istanze Amazon supportati EC2

> **Note**
>
> Dopo aver creato un ambiente di calcolo, AWS Batch non lo aggiorna nell'ambiente di calcolo. AMIs AWS Batch inoltre non aggiorna il file AMIs nel tuo ambiente di calcolo quando è disponibile una versione più recente dell'AMI ottimizzata per Amazon ECS. Sei responsabile della gestione del sistema operativo guest. Ciò include eventuali aggiornamenti e patch di sicurezza. Sei inoltre responsabile di qualsiasi software applicativo o utilità aggiuntivo che installi sulle risorse di elaborazione. Per utilizzare una nuova AMI per i tuoi AWS Batch lavori, procedi come segue:
>
> 1. Creare un nuovo ambiente di calcolo con la nuova AMI.
> 2. Aggiungere l'ambiente di calcolo a una coda di processi esistente.
> 3. Rimuovere il precedente ambiente di calcolo dalla coda di processi.
> 4. Eliminare l'ambiente di calcolo precedente.
>
> Nell'aprile 2022, è AWS Batch stato aggiunto un supporto avanzato per l'aggiornamento degli ambienti di elaborazione. Per ulteriori informazioni, consulta Aggiornare un ambiente di calcolo in AWS Batch. Per utilizzare l'aggiornamento avanzato degli ambienti di elaborazione per l'aggiornamento AMIs, segui queste regole:
>
> - Non impostate il parametro service role (`serviceRole`) o impostatelo sul ruolo collegato al AWSServiceRoleForBatchservizio.
> - Imposta il parametro allocation strategy (`allocationStrategy`) su`BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED` o. `SPOT_PRICE_CAPACITY_OPTIMIZED`
> - Imposta il parametro di aggiornamento all'ultima versione dell'immagine (`updateToLatestImageVersion`) su`true`.
> - Non specificare un ID AMI in `imageId`, `imageIdOverride`(in `ec2Configuration`) o nel modello di avvio (`launchTemplate`). Quando non specifichi un ID AMI, AWS Batch seleziona l'AMI ottimizzata Amazon ECS più recente che AWS Batch supporta al

momento dell'aggiornamento dell'infrastruttura. In alternativa, puoi specificare l'ID AMI nei `imageIdOverride` parametri `imageId` o. In alternativa, è possibile specificare il modello di avvio identificato dalle `LaunchTemplate` proprietà. La modifica di una di queste proprietà avvia un aggiornamento dell'infrastruttura. Se l'ID AMI è specificato nel modello di avvio, l'ID AMI non può essere sostituito specificando un ID AMI nei `imageIdOverride` parametri `imageId` o. L'ID AMI può essere sostituito solo specificando un modello di avvio diverso. Se la versione del modello di avvio è impostata su `$Default` o`$Latest`, l'ID AMI può essere sostituito impostando una nuova versione predefinita per il modello di avvio (if`$Default`) o aggiungendo una nuova versione al modello di avvio (if`$Latest`).

Se vengono seguite queste regole, qualsiasi aggiornamento che avvia un aggiornamento dell'infrastruttura causa la riseleziona l'ID AMI. Se l'`version`impostazione nel modello di avvio (`launchTemplate`) è impostata su `$Latest` o`$Default`, la versione più recente o predefinita del modello di lancio viene valutata al momento dell'aggiornamento dell'infrastruttura, anche se non `launchTemplate`era aggiornata.

Argomenti

- Specifiche dell'AMI delle risorse di calcolo
- Tutorial: Creare un'AMI per risorse di calcolo
- Usa un'AMI per carichi di lavoro GPU
- Deprecazione di Amazon Linux
- Deprecazione delle AMI Amazon EKS Amazon Linux 2
- Deprecazione delle AMI Amazon ECS Amazon Linux 2

Specifiche dell'AMI delle risorse di calcolo

La specifica AMI delle risorse di AWS Batch calcolo di base è composta da quanto segue:

Campo obbligatorio

- Una distribuzione Linux moderna che esegue almeno la versione 3.10 del kernel Linux su un'AMI di tipo di virtualizzazione HVM. I contenitori Windows non sono supportati.

> **Important**
>
> I lavori paralleli multinodo possono essere eseguiti solo su risorse di calcolo lanciate su un'istanza Amazon Linux con il `ecs-init` pacchetto installato. Ti consigliamo di utilizzare l'AMI ottimizzata Amazon ECS predefinita quando crei il tuo ambiente di elaborazione. È possibile farlo non specificando un AMI personalizzato. Per ulteriori informazioni, consulta Lavori paralleli multinodo.

- L'agente container Amazon ECS. Consigliamo di utilizzare la versione più recente. Per ulteriori informazioni, consulta Installazione di Amazon ECS Container Agent nella Amazon Elastic Container Service Developer Guide.
- Il driver di `awslogs` registro deve essere specificato come driver di registro disponibile con la variabile di `ECS_AVAILABLE_LOGGING_DRIVERS` ambiente all'avvio dell'agente contenitore Amazon ECS. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Container Service.
- Un demone Docker che esegue almeno la versione 1.9 e tutte le dipendenze di runtime di Docker. Per ulteriori informazioni, consulta Check runtime dependencies (Controlla dipendenze di runtime) nella documentazione di Docker.

> **Note**
>
> Ti consigliamo la versione Docker fornita con e testata con la corrispondente versione dell'agente Amazon ECS che stai utilizzando. Amazon ECS fornisce un changelog per la variante Linux dell'AMI ottimizzata per Amazon ECS su. GitHub Per ulteriori informazioni, consulta Changelog.

Consigliato

- Un processo di inizializzazione e nanny per eseguire e monitorare l'agente Amazon ECS. L'AMI ottimizzata per Amazon ECS utilizza il processo `ecs-init` upstart e altri sistemi operativi potrebbero utilizzarlo. `systemd` Per ulteriori informazioni ed esempi, consulta Example Container Instance User Data Configuration Scripts nella Amazon Elastic Container Service Developer Guide. Per ulteriori informazioni su`ecs-init`, consulta il `ecs-init`progetto su GitHub. Come minimo, gli ambienti di elaborazione gestiti richiedono che l'agente Amazon ECS si avvii all'avvio. Se l'agente

Amazon ECS non è in esecuzione sulla tua risorsa di elaborazione, non può accettare lavori da. AWS Batch

L'AMI ottimizzata per Amazon ECS è preconfigurata con questi requisiti e raccomandazioni. Ti consigliamo di utilizzare l'AMI ottimizzata Amazon ECS o un'AMI Amazon Linux con il `ecs-init` pacchetto installato per le tue risorse di calcolo. Scegli un'altra AMI se la tua applicazione richiede un sistema operativo specifico o una versione Docker non ancora disponibile in questi AMIs. Per ulteriori informazioni, consulta l'AMI ottimizzata per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Tutorial: Creare un'AMI per risorse di calcolo

Puoi creare la tua AMI di risorse di calcolo personalizzata da utilizzare per i tuoi ambienti di elaborazione gestiti e non gestiti. Per istruzioni, consulta il Specifiche dell'AMI delle risorse di calcolo. Quindi, dopo aver creato un'AMI personalizzata, puoi creare un ambiente di calcolo che utilizza quell'AMI a cui puoi associare una coda di lavoro. Infine, inizia a inviare i lavori a quella coda.

Per creare una risorsa di calcolo AMI personalizzata

1. Scegli un AMI di base da cui iniziare. L'AMI di base deve utilizzare la virtualizzazione HVM. L'AMI di base non può essere un'AMI Windows.

   > **Note**
   >
   > L'AMI scelta per un ambiente di elaborazione deve corrispondere all'architettura dei tipi di istanza che intendi utilizzare per quell'ambiente di elaborazione. Ad esempio, se l'ambiente di elaborazione utilizza tipi di istanza A1, l'AMI della risorsa di calcolo scelta deve supportare le istanze ARM. Amazon ECS offre versioni x86 e ARM dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta l'AMI Amazon Linux 2 ottimizzata per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

   L'AMI Amazon Linux 2 ottimizzata per Amazon ECS è l'AMI predefinita per le risorse di elaborazione in ambienti di elaborazione gestiti. L'AMI Amazon Linux 2 ottimizzata per Amazon ECS è preconfigurata e testata AWS Batch dagli AWS ingegneri. È un'AMI minima con cui iniziare e con cui puoi far funzionare AWS rapidamente le tue risorse di calcolo. Per ulteriori

informazioni, consulta Amazon ECS Optimized AMI nella Amazon Elastic Container Service Developer Guide.

In alternativa, puoi scegliere un'altra variante di Amazon Linux 2 e installare il `ecs-init` pacchetto con i seguenti comandi. Per ulteriori informazioni, consulta Installazione dell'agente container Amazon ECS su un' EC2 istanza Amazon Linux 2 nella Amazon Elastic Container Service Developer Guide:

```
$ sudo amazon-linux-extras disable docker
$ sudo amazon-linux-extras install ecs-init
```

Ad esempio, se desideri eseguire carichi di lavoro GPU sulle tue risorse di AWS Batch calcolo, puoi iniziare con l'AMI Amazon Linux Deep Learning. Quindi, configura l'AMI per eseguire i AWS Batch lavori. Per ulteriori informazioni, consulta Usa un'AMI per carichi di lavoro GPU.

> **Important**
>
> Puoi scegliere un'AMI di base che non supporti il `ecs-init` pacchetto. Tuttavia, se lo fai, devi configurare un modo per avviare l'agente Amazon ECS all'avvio e mantenerlo in esecuzione. Puoi anche visualizzare diversi esempi di script di configurazione dei dati utente che vengono utilizzati `systemd` per avviare e monitorare l'agente container Amazon ECS. Per ulteriori informazioni, consulta Esempi di script di configurazione dei dati utente dell'istanza del contenitore nella Amazon Elastic Container Service Developer Guide.

2. Avvia un'istanza dall'AMI di base selezionata con le opzioni di archiviazione appropriate per l'AMI. Puoi configurare la dimensione e il numero di volumi Amazon EBS collegati o i volumi di storage delle istanze se il tipo di istanza selezionato li supporta. Per ulteriori informazioni, consulta Launching an Instance e Amazon EC2 Instance Store nella Amazon EC2 User Guide.
3. Connect alla tua istanza SSH ed esegui tutte le attività di configurazione necessarie. Ciò potrebbe includere alcuni o tutti i seguenti passaggi:

   - Installazione dell'agente container Amazon ECS. Per ulteriori informazioni, consulta Installazione di Amazon ECS Container Agent nella Amazon Elastic Container Service Developer Guide.
   - Configurazione di uno script per la formattazione di volumi instance store.

- Aggiungere il volume dell'instance store o i file system Amazon EFS al `/etc/fstab` file in modo che vengano montati all'avvio.
- Configurazione delle opzioni Docker, come l'abilitazione del debug o la regolazione delle dimensioni dell'immagine di base.
- Installazione di pacchetti o copia di file.

Per ulteriori informazioni, consulta Connessione all'istanza Linux tramite SSH nella Amazon EC2 User Guide.

4. Se hai avviato l'agente container Amazon ECS sulla tua istanza, devi interromperlo e rimuovere tutti i file di checkpoint persistenti dei dati prima di creare l'AMI. Altrimenti, se non lo fai, l'agente non si avvia sulle istanze lanciate dalla tua AMI.

   a. Arresta l'agente del container di Amazon ECS.

      - AMI Amazon Linux 2 ottimizzata per Amazon ECS:

        ```
        sudo systemctl stop ecs
        ```

      - AMI Amazon Linux ottimizzata per Amazon ECS:

        ```
        sudo stop ecs
        ```

   b. Rimuovi i file persistenti del checkpoint dei dati. Per impostazione predefinita, questi file si trovano nella `/var/lib/ecs/data/` directory. Usa il seguente comando per rimuovere questi file, se ce ne sono.

      ```
      sudo rm -rf /var/lib/ecs/data/*
      ```

5. Crea una nuova AMI dall'istanza in esecuzione. Per ulteriori informazioni, consulta Creazione di un'AMI Linux supportata da Amazon EBS nella Amazon EC2 User Guide.

Per usare la tua nuova AMI con AWS Batch

1. Dopo aver creato la nuova AMI, crea un ambiente di calcolo con la nuova AMI. Per fare ciò, scegli il tipo di immagine e inserisci l'ID AMI personalizzato nella casella Image ID override quando crei l'ambiente di AWS Batch calcolo. Per ulteriori informazioni, consulta the section called "Tutorial: crea un ambiente di elaborazione gestito utilizzando le risorse Amazon EC2 ".

> **Note**
>
> L'AMI scelta per un ambiente di elaborazione deve corrispondere all'architettura dei tipi di istanza che intendi utilizzare per quell'ambiente di elaborazione. Ad esempio, se l'ambiente di elaborazione utilizza tipi di istanza A1, l'AMI della risorsa di calcolo scelta deve supportare le istanze ARM. Amazon ECS offre versioni x86 e ARM dell'AMI Amazon Linux 2 ottimizzata per Amazon ECS. Per ulteriori informazioni, consulta l'AMI Amazon Linux 2 ottimizzata per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

2. Crea una coda dei processi e associa il nuovo ambiente di calcolo. Per ulteriori informazioni, consulta Creare una coda di lavoro.

> **Note**
>
> Tutti gli ambienti di calcolo associati a una coda di processi devono condividere la stessa architettura, poiché AWS Batch non supporta diversi tipi di architettura in una singola coda.

3. (Facoltativo) Invia un processo di esempio alla nuova coda di processi. Per ulteriori informazioni, consultare Esempi di definizione di Job, Creare una definizione di processo a nodo singolo e Tutorial: invia un lavoro.

Usa un'AMI per carichi di lavoro GPU

Per eseguire carichi di lavoro GPU sulle risorse di AWS Batch elaborazione, devi utilizzare un'AMI con supporto GPU. Per ulteriori informazioni, consulta Working with GPUs on Amazon ECS e Amazon ECS Optimized nella AMIs Amazon Elastic Container Service Developer Guide.

Negli ambienti di elaborazione gestiti, se l'ambiente di calcolo specifica tipi o famiglie di istanze `p3p4,p5,p6,g3`, `g3s` `g4g5`, o tipi di `g6` istanze, utilizza AWS Batch un'AMI ottimizzata per GPU Amazon ECS.

In ambienti di elaborazione non gestiti, è consigliata un'AMI Amazon ECS ottimizzata per GPU. Puoi utilizzare AWS Systems Manager Parameter Store GetParameter e GetParametersByPathoperations per recuperare GetParametersi metadati per la versione ottimizzata per GPU Amazon ECS consigliata. AWS Command Line Interface AMIs

> **Note**
>
> La famiglia di p5 istanze è supportata solo su versioni uguali o successive all'AMI ottimizzata per GPU 20230912 di Amazon ECS e sono incompatibili con p2 tutti i tipi di istanze. g2 Se devi usare p5 istanze, assicurati che il tuo ambiente di calcolo non contenga g2 istanze p2 o e utilizzi l'ultima AMI Batch predefinita. La creazione di un nuovo ambiente di calcolo utilizzerà l'AMI più recente, ma se stai aggiornando l'ambiente di calcolo per includerep5, puoi assicurarti di utilizzare l'AMI più recente impostando `updateToLatestImageVersion`su `true` nelle `ComputeResource` proprietà. Per ulteriori informazioni sulla compatibilità delle AMI con le istanze GPU, consulta Working with on Amazon ECS nella GPUs Amazon Elastic Container Service Developer Guide.

Gli esempi seguenti mostrano come usare il comando. GetParameter

AWS CLI

```
$ aws ssm get-parameter --name /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended \
                        --region us-east-2 --output json
```

L'output include le informazioni AMI nel `Value` parametro.

```
{
    "Parameter": {
        "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended",
        "LastModifiedDate": 1555434128.664,
        "Value": "{\"schema_version\":1,\"image_name\":\"amzn2-ami-ecs-gpu-
hvm-2.0.20190402-x86_64-ebs\",\"image_id\":\"ami-083c800fe4211192f\",\"os\":\"Amazon
 Linux 2\",\"ecs_runtime_version\":\"Docker version 18.06.1-ce\",\"ecs_agent_version
\":\"1.27.0\"}",
        "Version": 9,
        "Type": "String",
        "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended"
    }
}
```

Python

```
from __future__ import print_function

import json
import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameter(Name='/aws/service/ecs/optimized-ami/amazon-linux-2/
gpu/recommended')
jsonVal = json.loads(response['Parameter']['Value'])
print("image_id   = " + jsonVal['image_id'])
print("image_name = " + jsonVal['image_name'])
```

L'output include solo il nome e l'ID dell'AMI:

```
image_id   = ami-083c800fe4211192f
image_name = amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Gli esempi seguenti illustrano l'uso di GetParameters.

AWS CLI

```
$ aws ssm get-parameters --names  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_name \
                                  /aws/service/ecs/optimized-ami/amazon-linux-2/gpu/
recommended/image_id \
                          --region us-east-2 --output json
```

L'output include i metadati completi per ciascuno dei parametri:

```
{
    "InvalidParameters": [],
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
```

```
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters(
            Names=['/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name',
                   '/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id'])
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

L'output include l'ID AMI e il nome AMI, utilizzando il percorso completo per i nomi.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
```

Gli esempi seguenti mostrano come utilizzare il GetParametersByPathcomando.

AWS CLI

```
$ aws ssm get-parameters-by-path --path /aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended \
                                --region us-east-2 --output json
```

L'output include i metadati completi per tutti i parametri nel percorso specificato.

```
{
    "Parameters": [
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_agent_version",
            "LastModifiedDate": 1555434128.801,
            "Value": "1.27.0",
            "Version": 8,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_agent_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
ecs_runtime_version",
            "LastModifiedDate": 1548368308.213,
            "Value": "Docker version 18.06.1-ce",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/ecs_runtime_version"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_id",
            "LastModifiedDate": 1555434128.749,
            "Value": "ami-083c800fe4211192f",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_id"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
image_name",
```

```
            "LastModifiedDate": 1555434128.712,
            "Value": "amzn2-ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs",
            "Version": 9,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/image_name"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
os",
            "LastModifiedDate": 1548368308.143,
            "Value": "Amazon Linux 2",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/os"
        },
        {
            "Name": "/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/
schema_version",
            "LastModifiedDate": 1548368307.914,
            "Value": "1",
            "Version": 1,
            "Type": "String",
            "ARN": "arn:aws:ssm:us-east-2::parameter/aws/service/ecs/optimized-ami/
amazon-linux-2/gpu/recommended/schema_version"
        }
    ]
}
```

Python

```
from __future__ import print_function

import boto3

ssm = boto3.client('ssm', 'us-east-2')

response = ssm.get_parameters_by_path(Path='/aws/service/ecs/optimized-ami/amazon-
linux-2/gpu/recommended')
for parameter in response['Parameters']:
    print(parameter['Name'] + " = " + parameter['Value'])
```

L'output include i valori di tutti i nomi dei parametri nel percorso specificato, utilizzando il percorso completo per i nomi.

```
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_agent_version =
 1.27.0
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/ecs_runtime_version =
 Docker version 18.06.1-ce
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_id =
 ami-083c800fe4211192f
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/image_name = amzn2-
ami-ecs-gpu-hvm-2.0.20190402-x86_64-ebs
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/os = Amazon Linux 2
/aws/service/ecs/optimized-ami/amazon-linux-2/gpu/recommended/schema_version = 1
```

Per ulteriori informazioni, consulta Recupero dei metadati AMI ottimizzati per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Deprecazione di Amazon Linux

L'AMI Amazon Linux (chiamata anche Amazon Linux 1) ha raggiunto la fine del suo ciclo di vita il 31 dicembre 2023. AWS Batch ha interrotto il supporto per l'AMI Amazon Linux in quanto non riceverà aggiornamenti di sicurezza o correzioni di bug a partire dal 1° gennaio 2024. Per ulteriori informazioni su Amazon Linux end-of-life, consulta le domande frequenti su AL.

Ti consigliamo di aggiornare gli ambienti di elaborazione esistenti basati su Amazon Linux ad Amazon Linux 2023 per evitare interruzioni impreviste del carico di lavoro e continuare a ricevere aggiornamenti di sicurezza e di altro tipo.

I tuoi ambienti di elaborazione che utilizzano l'AMI Amazon Linux potrebbero continuare a funzionare oltre la end-of-life data del 31 dicembre 2023. Tuttavia, questi ambienti di elaborazione non riceveranno più nuovi aggiornamenti software, patch di sicurezza o correzioni di bug da. AWS Successivamente end-of-life, è tua responsabilità mantenere questi ambienti di calcolo sull'AMI Amazon Linux. Consigliamo di migrare gli ambienti di AWS Batch elaborazione su Amazon Linux 2023 o Amazon Linux 2 per mantenere prestazioni e sicurezza ottimali.

Per assistenza nella migrazione AWS Batch dall'AMI Amazon Linux ad Amazon Linux 2023 o Amazon Linux 2, vedi Aggiornamento degli ambienti di calcolo -. AWS Batch

Deprecazione delle AMI Amazon EKS Amazon Linux 2

AWS terminerà il supporto per Amazon Linux 2 ottimizzato per Amazon EKS AMIs, a partire dal 26/11/25. Consigliamo di migrare gli ambienti di calcolo AWS Batch Amazon EKS su Amazon Linux 2023 prima del 26/11/25 per mantenere prestazioni e sicurezza ottimali.

Sebbene tu possa continuare a utilizzare Amazon EKS ottimizzato per Amazon EKS 2 AMIs sui tuoi ambienti di calcolo Amazon EKS oltre la end-of-support data del 26/11/25, questi ambienti di elaborazione non riceveranno più nuovi aggiornamenti software, patch di sicurezza o correzioni di bug. AWS Successivamente end-of-life, è tua responsabilità mantenere questi ambienti di calcolo sull'AMI Amazon Linux 2 ottimizzata per Amazon EKS.

Per ulteriori informazioni su Amazon EKS AL2 end-of-life, consulta la deprecazione delle AMI di Amazon EKS nella Guida per FAQs l'utente di Amazon EKS.

Per informazioni sulla migrazione degli ambienti di calcolo AWS Batch Amazon EKS da Amazon Linux 2 ad Amazon Linux 2023, consulta. Come eseguire l'aggiornamento da EKS a EKS 023 AL2 AL2

Deprecazione delle AMI Amazon ECS Amazon Linux 2

AWS terminerà il supporto per Amazon Linux 2. A partire da gennaio 2026, AWS Batch cambierà l'AMI predefinita per i nuovi ambienti di calcolo Amazon ECS da Amazon Linux 2 ad Amazon Linux 2023. Consigliamo di migrare gli ambienti di calcolo AWS Batch Amazon ECS su Amazon Linux 2023, per mantenere prestazioni e sicurezza ottimali.

Per ulteriori informazioni su Amazon Linux 2 end-of-life, consulta Amazon Linux 2 FAQs.

Per informazioni sulle differenze tra Amazon Linux 2 e Amazon Linux 2023, consulta Confronta Amazon Linux 2023 e Amazon Linux 2 nella Guida per l'utente di Amazon Linux 2023.

Per informazioni sulle modifiche apportate ad Amazon Linux 2023 per l'AMI ottimizzata per Amazon ECS, consulta Migrazione da un'AMI ottimizzata per Amazon Linux 2023 per Amazon ECS nella Guida per l'utente di Amazon ECS.

Per informazioni sulla migrazione degli ambienti di calcolo AWS Batch Amazon ECS da Amazon Linux 2 ad Amazon Linux 2023, consulta. Come migrare da ECS AL2 a ECS 023 AL2

Usa i modelli di EC2 lancio di Amazon con AWS Batch

AWS Batch supporta l'utilizzo di modelli di EC2 lancio di Amazon con i tuoi ambienti di EC2 elaborazione. Con i modelli di avvio, puoi modificare la configurazione predefinita delle tue risorse di AWS Batch calcolo senza doverne creare di personalizzate. AMIs

> **Note**
>
> I modelli di avvio non sono supportati nelle risorse AWS Fargate.

È necessario creare un modello di avvio prima di associarlo a un ambiente di calcolo. Puoi creare un modello di lancio nella EC2 console Amazon. In alternativa, puoi utilizzare AWS CLI o un AWS SDK. Ad esempio, il seguente file JSON rappresenta un modello di avvio che ridimensiona il volume di dati Docker per l'AMI della risorsa di AWS Batch calcolo predefinita e lo imposta anche per la crittografia.

```
{
    "LaunchTemplateName": "increase-container-volume-encrypt",
    "LaunchTemplateData": {
        "BlockDeviceMappings": [
            {
                "DeviceName": "/dev/xvda",
                "Ebs": {
                    "Encrypted": true,
                    "VolumeSize": 100,
                    "VolumeType": "gp2"
                }
            }
        ]
    }
}
```

È possibile creare il modello di avvio precedente salvando il codice JSON in un file chiamato `lt-data.json` ed eseguendo il comando seguente. AWS CLI

```
aws ec2 --region <region> create-launch-template --cli-input-json file://lt-data.json
```

Per ulteriori informazioni sui modelli di lancio, consulta Launching an Instance from a Launch Template nella Amazon EC2 User Guide.

Se si utilizza un modello di avvio per creare il tuo ambiente di calcolo, è possibile spostare i seguenti parametri di ambiente di calcolo esistenti sul modello di avvio:

> **Note**
>
> Supponiamo che uno qualsiasi di questi parametri (ad eccezione dei EC2 tag Amazon) sia specificato sia nel modello di avvio che nella configurazione dell'ambiente di calcolo. Quindi, i parametri dell'ambiente di calcolo hanno la precedenza. I EC2 tag Amazon vengono uniti tra il modello di lancio e la configurazione dell'ambiente di calcolo. In caso di collisione sulla chiave del tag, il valore nella configurazione dell'ambiente di calcolo ha la precedenza.

- Coppia di EC2 chiavi Amazon
- ID EC2 AMI Amazon
- Gruppo di sicurezza IDs
- EC2 Tag Amazon

I seguenti parametri del modello di avvio vengono ignorati da AWS Batch:

- Tipo di istanza (specificare i tipi di istanza desiderati al momento della creazione dell'ambiente di calcolo)
- Ruolo istanza (specificare il ruolo di istanza desiderato al momento della creazione dell'ambiente di calcolo)
- Sottoreti dell'interfaccia di rete (specificare i tipi di sottoreti desiderati al momento della creazione dell'ambiente di calcolo)
- Opzioni di mercato delle istanze (AWS Batch deve controllare la configurazione dell'istanza Spot)
- Disattiva la terminazione dell'API (AWS Batch deve controllare il ciclo di vita dell'istanza)

AWS Batch aggiorna il modello di lancio solo con una nuova versione del modello di avvio durante gli aggiornamenti dell'infrastruttura. Per ulteriori informazioni, consulta Aggiornare un ambiente di calcolo in AWS Batch.

Modelli di avvio predefiniti e sostituiti

È possibile definire un modello di avvio predefinito per l'ambiente di calcolo e un modello di avvio sostitutivo per tipi e famiglie di istanze specifici. Questo può esserti utile in modo che il modello predefinito venga utilizzato per la maggior parte dei tipi di istanze negli ambienti di calcolo.

Le variabili `$Default` di sostituzione `$Latest` possono essere utilizzate invece di denominare una versione specifica. Se non si fornisce un modello di avvio sostitutivo, viene applicato automaticamente il modello di avvio predefinito.

Se si utilizza la `$Latest` variabile `$Default` or, AWS Batch applicherà le informazioni correnti al momento della creazione dell'ambiente di calcolo. Se la versione predefinita o più recente verrà modificata in futuro, è necessario aggiornare le informazioni tramite UpdateComputeEnvironmento tramite il pulsante Console di gestione AWS - AWS Batch.

Per fornire ulteriore flessibilità, puoi definire che i modelli di avvio di override vengano applicati a tipi o famiglie di istanze di calcolo specifici.

> **Note**
>
> È possibile specificare fino a dieci (10) modelli di avvio override per ambiente di calcolo.

Utilizzate il `targetInstanceTypes` parametro per selezionare il tipo o la famiglia di istanza che deve utilizzare questo modello di avvio di override. Il tipo o la famiglia di istanze devono essere prima identificati dal `instanceTypes`parametro.

Se definisci le sostituzioni dei modelli di avvio e decidi di rimuoverle in un secondo momento, puoi passare un array vuoto per annullare l'impostazione del `overrides`parametro nell'`UpdateComputeEnvironment`operazione API. Puoi anche scegliere di non includere il `overrides` parametro quando invii l'operazione API. `UpdateComputeEnvironment` Per ulteriori informazioni, consulta, `LaunchTemplateSpecification.overrides`

Per ulteriori informazioni, `LaunchTemplateSpecificationOverride.targetInstanceTypes`consulta la guida di riferimento delle AWS Batch API.

Dati EC2 degli utenti Amazon nei modelli di lancio

Puoi fornire i dati EC2 utente di Amazon nel tuo modello di lancio eseguito da cloud-init all'avvio delle istanze. I tuoi dati utente possono eseguire scenari di configurazione comuni, tra cui, a titolo esemplificativo ma non esaustivo, i seguenti:

- Inclusione di utenti o gruppi.
- Installazione di pacchetti
- Creazione di partizioni e file system

I dati EC2 degli utenti Amazon nei modelli di lancio devono essere in formato di archivio multiparte MIME. Questo perché i dati utente vengono uniti ad altri dati AWS Batch utente necessari per configurare le risorse di elaborazione. È possibile unire più blocchi di dati utente in un unico blocco, detto file MIME in più parti. Ad esempio, potresti voler combinare un cloud boothook che configura il daemon Docker con uno script di user data shell che scrive informazioni di configurazione per l'agente container Amazon ECS.

Se lo utilizzi AWS CloudFormation, il AWS::CloudFormation::Inittipo può essere utilizzato con lo script di supporto cfn-init per eseguire scenari di configurazione comuni.

Un file MIME in più parti è composto dai seguenti elementi:

- Il tipo di contenuto e la dichiarazione di delimitazione della parte: `Content-Type: multipart/mixed; boundary="==BOUNDARY=="`
- La dichiarazione della versione MIME: `MIME-Version: 1.0`
- Uno o più blocchi di dati utente che contengono i seguenti componenti:
  - Il limite di apertura che segnala l'inizio di un blocco di dati utente:`--==BOUNDARY==`. È necessario mantenere vuota la linea prima di questo limite.
  - La dichiarazione del tipo di contenuto per il blocco: `Content-Type: text/cloud-config; charset="us-ascii"`. Per ulteriori informazioni sui tipi di contenuto, consulta la documentazione di cloud-init. È necessario lasciare vuota la riga dopo la dichiarazione del tipo di contenuto.
  - Il contenuto dei dati utente, ad esempio un elenco di comandi o `cloud-init` direttive di shell.
- Il limite di chiusura che segnala la fine del file multiparte MIME:. `--==BOUNDARY==--` È necessario mantenere vuota la linea prima del limite di chiusura.

> **Note**
>
> Se aggiungi dati utente a un modello di lancio nella EC2 console Amazon, puoi incollarli come testo semplice. Oppure puoi caricarli da un file. Se utilizzi AWS CLI o un AWS SDK, devi prima `base64` codificare i dati utente e inviare quella stringa come valore del `UserData` parametro quando chiami CreateLaunchTemplate, come mostrato in questo file JSON.
>
> ```
> {
>     "LaunchTemplateName": "base64-user-data",
>     "LaunchTemplateData": {
>         "UserData":
>  "ewogICAgIkxhdW5jaFRlbXBsYXRlTmFtZSI6ICJpbmNyZWFzZS1jb250YWluZXItdm9sdW..."
>     }
> }
> ```

Argomenti

- Riferimento: esempi di modelli di EC2 lancio di Amazon

Riferimento: esempi di modelli di EC2 lancio di Amazon

Di seguito sono riportati alcuni esempi di file MIME multiparte che puoi usare per creare modelli personalizzati.

Esempi

- Esempio: monta un file system Amazon EFS esistente
- Esempio: sovrascrivi la configurazione predefinita dell'agente container Amazon ECS
- Esempio: monta un file system Amazon FSx for Lustre esistente

Esempio: monta un file system Amazon EFS esistente

Example

Questo esempio di file multiparte MIME configura la risorsa di calcolo per installare il `amazon-efs-utils` pacchetto e montare un file system Amazon EFS esistente su. `/mnt/efs`

```
MIME-Version: 1.0
```

```
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs tls,_netdev" >> /etc/fstab
- mount -a -t efs defaults

--==MYBOUNDARY==--
```

Esempio: sovrascrivi la configurazione predefinita dell'agente container Amazon ECS

Example

Questo esempio di file MIME in più parti sostituisce le impostazioni predefinite per la pulizia di un'immagine Docker per una risorsa di calcolo.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/x-shellscript; charset="us-ascii"

#!/bin/bash
echo ECS_IMAGE_CLEANUP_INTERVAL=60m >> /etc/ecs/ecs.config
echo ECS_IMAGE_MINIMUM_CLEANUP_AGE=60m >> /etc/ecs/ecs.config

--==MYBOUNDARY==--
```

Esempio: monta un file system Amazon FSx for Lustre esistente

Example

Questo esempio di file MIME multiparte configura la risorsa di calcolo per installare il `lustre2.10` pacchetto dalla libreria Extras e montare un file system FSx for Lustre esistente su e un nome di

montaggio di. `/scratch` `fsx` Questo esempio è per Amazon Linux 2. Per istruzioni di installazione per altre distribuzioni Linux, consulta Installazione del client Lustre nella Amazon FSx for Lustre User Guide. Per ulteriori informazioni, consulta Mounting your Amazon FSx file system automaticamente nella Amazon FSx for Lustre User Guide.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

runcmd:
- file_system_id_01=fs-0abcdef1234567890
- region=us-east-2
- fsx_directory=/scratch
- amazon-linux-extras install -y lustre2.10
- mkdir -p ${fsx_directory}
- mount -t lustre ${file_system_id_01}.fsx.${region}.amazonaws.com@tcp:fsx
 ${fsx_directory}

--==MYBOUNDARY==--
```

Nei volumi e nei membri MountPoints delle proprietà del contenitore, i punti di montaggio devono essere mappati nel contenitore.

```
{
    "volumes": [
        {
            "host": {
                "sourcePath": "/scratch"
            },
            "name": "Scratch"
        }
    ],
    "mountPoints": [
        {
            "containerPath": "/scratch",
            "sourceVolume": "Scratch"
        }
    ],
}
```

Configurazione del servizio IMDS (Instance Metadata Service)

L'Instance Metadata Service (IMDS) fornisce metadati sulle EC2 istanze alle applicazioni in esecuzione su tali istanze. Utilizzali IMDSv2 per tutti i nuovi carichi di lavoro e migra i carichi di lavoro esistenti da a per una maggiore sicurezza. IMDSv1 IMDSv2 Per ulteriori informazioni su IMDS e sulla configurazione di IMDS, consulta Usare i metadati dell'istanza per gestire l'istanza e Configurare le EC2 opzioni dei metadati dell'istanza per le nuove istanze nella Amazon User Guide. EC2

Scenari di configurazione

Scegli il metodo di configurazione appropriato in base alla configurazione del tuo ambiente di calcolo:

AMI predefinita senza modello di avvio

Quando utilizzi l' AWS Batch AMI predefinita e non specifichi un modello di avvio, scegli una di queste opzioni:

1. Usa l'AMI predefinita di Amazon Linux 2023: Amazon Linux 2023 richiede IMDSv2 per impostazione predefinita. Quando crei il tuo ambiente di calcolo, seleziona Amazon Linux 2023 come tipo di immagine.
2. Imposta la IMDSv2 configurazione a livello di account: configura il tuo AWS account in modo che sia necessario IMDSv2 per tutte le nuove istanze. Questa impostazione influisce su tutte le nuove istanze che avvii nell'account. Per istruzioni, consulta Imposta IMDSv2 come predefinito per l'account nella Amazon EC2 User Guide.

> **Note**
>
> La configurazione IMDS a livello di account può essere sostituita dal modello di avvio o dalla configurazione AMI. Le impostazioni del modello di avvio hanno la precedenza sulle impostazioni a livello di account.

AMI personalizzata senza modello di avvio

Quando utilizzi un'AMI personalizzata senza un modello di avvio, scegli una di queste opzioni:

1. Usa Amazon Linux 2023 come base: crea la tua AMI personalizzata utilizzando Amazon Linux 2023 come immagine di base. Per informazioni sulla creazione di contenuti personalizzati AMIs per Batch, vedereTutorial: Creare un'AMI per risorse di calcolo.

2. Configura IMDSv2 nell'AMI personalizzata: quando crei l'AMI personalizzata, configurala in base alle esigenze IMDSv2. Per istruzioni, consulta Configurare le opzioni dei metadati delle istanze per le AMI personalizzate nella Amazon EC2 User Guide.
3. Imposta la IMDSv2 configurazione a livello di account: configura il tuo AWS account in modo che sia necessario IMDSv2 per tutte le nuove istanze. Questa impostazione influisce su tutte le nuove istanze che avvii nell'account. Per istruzioni, consulta Imposta IMDSv2 come predefinito per l'account nella Amazon EC2 User Guide.

> **Note**
>
> La configurazione IMDS a livello di account può essere sostituita dal modello di avvio o dalla configurazione AMI. Le impostazioni del modello di avvio hanno la precedenza sulle impostazioni a livello di account.

Utilizzo dei modelli di avvio

Quando utilizzi i modelli di avvio nel tuo ambiente di calcolo, aggiungi le opzioni di metadati al modello di lancio in base alle tue esigenze. IMDSv2 Per ulteriori informazioni sull'utilizzo dei modelli di avvio con Batch, consultaUsa i modelli di EC2 lancio di Amazon con AWS Batch.

```
{
    "LaunchTemplateName": "batch-imdsv2-template",
    "VersionDescription": "IMDSv2 only template for Batch",
    "LaunchTemplateData": {
        "MetadataOptions": {
            "HttpTokens": "required"
        }
    }
}
```

Crea il modello di lancio utilizzando la AWS CLI:

```
aws ec2 create-launch-template --cli-input-json file://imds-template.json
```

EC2 configurazioni

AWS Batch utilizza Amazon ECS ottimizzato AMIs per ambienti EC2 di calcolo EC2 Spot. L'impostazione predefinita è Amazon Linux 2 (`ECS_AL2`). A partire da gennaio 2026, l'impostazione predefinita cambierà in AL2023 ()`ECS_AL2023`.

AWS terminerà il supporto per Amazon Linux 2. Consigliamo di migrare gli ambienti di calcolo AWS Batch Amazon ECS su Amazon Linux 2023 per mantenere prestazioni e sicurezza ottimali. Per ulteriori informazioni, consulta Deprecazione delle AMI Amazon ECS Amazon Linux 2.

Ti consigliamo di aggiornare gli ambienti di elaborazione esistenti basati su Amazon Linux ad Amazon Linux 2023 per evitare interruzioni impreviste del carico di lavoro e continuare a ricevere aggiornamenti di sicurezza e di altro tipo.

Per informazioni sulla migrazione AWS Batch dall'AMI Amazon Linux ad Amazon Linux 2023, consulta Come migrare da ECS AL2 a ECS 023 AL2

Argomenti

Come migrare da ECS AL2 a ECS 023 AL2

AL2023 è un sistema operativo basato su Linux progettato per fornire un ambiente sicuro, stabile e ad alte prestazioni per le applicazioni cloud. Per ulteriori informazioni sulle differenze tra AL2 e AL2 023, consulta Confronta Amazon Linux 2023 e Amazon Linux 2 nella Amazon Linux 2023 User Guide.

A partire da gennaio 2026, AWS Batch cambierà l'AMI predefinita per i nuovi ambienti di calcolo Amazon ECS da Amazon Linux 2 ad Amazon Linux 2023 perché AWS terminerà il supporto per Amazon Linux 2. L'AMI predefinito viene utilizzato quando non si specifica un valore per il campo ImageType.ec2Configuration durante la creazione di un nuovo ambiente di calcolo. Consigliamo di migrare gli ambienti di calcolo AWS Batch Amazon ECS su Amazon Linux 2023 per mantenere prestazioni e sicurezza ottimali.

A seconda di come è configurato l'ambiente di calcolo, puoi utilizzare uno dei seguenti percorsi di aggiornamento da 0 a 023. AL2 AL2

Effettua l'aggiornamento utilizzando Ec2Configuration. ImageType

- Se non stai utilizzando un modello di avvio o sostituzioni di modelli di avvio, modifica Ec2Configuration. ImageTypea `ECS_AL2023` (o `ECS_AL2023_NVIDIA` quando si usano istanze GPU) e poi esegui. UpdateComputeEnvironment
- Se si specifica una configurazione Ec2. ImageIdOverridequindi Ec2Configuration. ImageTypedeve corrispondere al tipo di AMI specificato in Ec2Configuration. ImageIdOverride.

  In caso di mancata corrispondenza `ImageIdOverrideImageType`, l'ambiente di calcolo potrebbe non funzionare correttamente.

Esegui l'aggiornamento utilizzando i modelli di avvio

- Se utilizzi un modello di lancio che specifica un'AMI basata su`ECS_AL2023`, assicurati che il modello di lancio sia compatibile con Amazon Linux 2023. Per informazioni sulle modifiche apportate ad Amazon Linux 2023 per l'AMI ottimizzata per Amazon ECS, consulta Migrazione da un'AMI ottimizzata per Amazon Linux 2023 per Amazon ECS nella Guida per l'utente di Amazon ECS.
- Per AL2 023 AMIs, verifica che tutti i dati utente o gli script di inizializzazione personalizzati siano compatibili con l'ambiente 023 e il sistema di gestione dei pacchetti. AL2

Esegui l'aggiornamento utilizzando CloudFormation

- Se lo utilizzi CloudFormation per gestire i tuoi ambienti di elaborazione, aggiorna il modello per modificare la `ImageType` proprietà in `Ec2Configuration` from `ECS_AL2` to `ECS_AL2023` (o `ECS_AL2023_NVIDIA` quando usi istanze GPU):

  ```
  ComputeEnvironment:
    Type: AWS::Batch::ComputeEnvironment
    Properties:
      ComputeResources:
        Ec2Configuration:
          - ImageType: ECS_AL2023
  ```

  Quindi aggiorna lo CloudFormation stack per applicare le modifiche.

- Se il CloudFormation modello specifica l'utilizzo di un'AMI personalizzata`ImageId0verride`, assicurati che l'ID AMI corrisponda a un AMI AL2 basato su 023 e `ImageType` corrisponda all'impostazione.

Considerazioni sulla migrazione

Durante la migrazione da Amazon Linux 2 ad Amazon Linux 2023, considera quanto segue:

- Gestione dei pacchetti: Amazon Linux 2023 utilizza `dnf` invece che `yum` per la gestione dei pacchetti.
- Servizi di sistema: alcuni servizi di sistema e le relative configurazioni possono differire tra AL2 0 AL2 e 023.
- Container runtime: entrambi AL2 e AL2 023 supportano Docker, ma AL2 023 può avere configurazioni predefinite diverse.
- Sicurezza: AL2 023 include funzionalità di sicurezza avanzate e potrebbe richiedere aggiornamenti alle configurazioni relative alla sicurezza.
- Instance Metadata Service Version 2 (IMDSv2): IMDSv2 è un servizio orientato alla sessione che richiede l'autenticazione basata su token per accedere ai metadati dell'istanza, garantendo una maggiore sicurezza. EC2 Per ulteriori informazioni su IMDS, consulta la sezione How Instance Metadata Service Version 2 nella Amazon EC2 User Guide.

Per un elenco completo di modifiche e considerazioni sulla migrazione, consulta Migrazione da Amazon Linux 2 a un'AMI ottimizzata per Amazon ECS 2023 nella Amazon ECS User Guide.

Strategie di allocazione del tipo di istanza per AWS Batch

Quando viene creato un ambiente di elaborazione gestito, AWS Batch seleziona tra quelli `instanceTypes` specificati i tipi di istanza più adatti alle esigenze dei lavori. La strategia di allocazione definisce il comportamento quando AWS Batch richiede capacità aggiuntiva. Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate. Non specificare questo parametro.

`BEST_FIT` (predefinito)

AWS Batch seleziona il tipo di istanza più adatto alle esigenze dei job con una preferenza per il tipo di istanza con il costo più basso. Se non sono disponibili istanze aggiuntive del tipo di istanza selezionato, AWS Batch attende che siano disponibili. Se non ci sono abbastanza istanze

disponibili o se l'utente sta raggiungendo le quote di EC2 servizio Amazon, i lavori aggiuntivi non vengono eseguiti fino al completamento dei processi attualmente in esecuzione. Questa strategia di allocazione riduce i costi, ma può limitare il ridimensionamento. Se utilizzi Spot Fleets con`BEST_FIT`, devi specificare il ruolo IAM di Spot Fleet. `BEST_FIT`non è supportato durante l'aggiornamento degli ambienti di elaborazione. Per ulteriori informazioni, consulta Aggiornare un ambiente di calcolo in AWS Batch.

> **Note**
>
> AWS Batch gestisce AWS le risorse del tuo account. Gli ambienti di calcolo con la strategia di allocazione BEST_FIT utilizzavano originariamente le configurazioni di avvio per impostazione predefinita. Tuttavia, l'uso delle configurazioni di avvio con nuovi AWS account sarà limitato nel tempo. Pertanto, a partire dalla fine di aprile 2024, gli ambienti di calcolo BEST_FIT appena creati utilizzeranno per impostazione predefinita i modelli di avvio. Se il tuo ruolo di servizio non dispone delle autorizzazioni per gestire i modelli di lancio, AWS Batch puoi continuare a utilizzare le configurazioni di avvio. Gli ambienti di elaborazione esistenti continueranno a utilizzare le configurazioni di avvio.

`BEST_FIT_PROGRESSIVE`

AWS Batch seleziona tipi di istanze aggiuntivi sufficientemente grandi da soddisfare i requisiti dei lavori in coda. Sono preferiti i tipi di istanza con un costo inferiore per ogni unità vCPU. Se non sono disponibili istanze aggiuntive dei tipi di istanza selezionati in precedenza, AWS Batch seleziona nuovi tipi di istanza.

> **Note**
>
> Per i lavori paralleli a più nodi, AWS Batch sceglie il tipo di istanza ottimale disponibile. Se il tipo di istanza non è disponibile a causa di una capacità insufficiente, gli altri tipi di istanza all'interno della famiglia non vengono avviati.

`SPOT_CAPACITY_OPTIMIZED`

AWS Batch seleziona uno o più tipi di istanze sufficientemente grandi da soddisfare i requisiti dei lavori in coda. Sono preferiti i tipi di istanza che hanno meno probabilità di essere interrotte. Questa strategia di allocazione è disponibile solo per le risorse di calcolo di istanze Spot.

`SPOT_PRICE_CAPACITY_OPTIMIZED`

La strategia di allocazione ottimizzata per prezzo e capacità esamina sia il prezzo che la capacità per selezionare i pool di istanze spot che hanno il prezzo più basso possibile e meno probabilità di interruzioni. Questa strategia di allocazione è disponibile solo per le risorse di calcolo di istanze Spot.

> **Note**
>
> Si consiglia di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` piuttosto che `SPOT_CAPACITY_OPTIMIZED` nella maggior parte dei casi.

Le `BEST_FIT` strategie `BEST_FIT_PROGRESSIVE` and utilizzano istanze On-Demand o Spot, mentre le `SPOT_PRICE_CAPACITY_OPTIMIZED` strategie `SPOT_CAPACITY_OPTIMIZED` and utilizzano istanze Spot. Tuttavia, AWS Batch potrebbe essere necessario superare il limite per `maxvCpus` soddisfare i requisiti di capacità. In questo caso, AWS Batch non supera mai di più `maxvCpus` di una singola istanza.

Gestione della memoria delle risorse di calcolo

Quando l'agente container Amazon ECS registra una risorsa di calcolo in un ambiente di elaborazione, deve determinare la quantità di memoria disponibile per la risorsa di elaborazione da riservare per i tuoi lavori. A causa del sovraccarico di memoria della piattaforma e della memoria occupata dal kernel di sistema, questo numero è diverso dalla quantità di memoria installata per le istanze Amazon EC2 . Ad esempio, un'istanza `m4.large` dispone di 8 GiB di memoria installata. Tuttavia, ciò non sempre si traduce in esattamente 8192 MiB di memoria disponibili per i lavori quando la risorsa di elaborazione viene registrata.

Si supponga di specificare 8192 MiB per il job e che nessuna delle risorse di elaborazione disponga di 8192 MiB o più di memoria disponibile per soddisfare questo requisito. Quindi, il lavoro non può essere inserito nel tuo ambiente di elaborazione. Se utilizzi un ambiente di elaborazione gestito, AWS Batch devi avviare un tipo di istanza più grande per soddisfare la richiesta.

L'AMI di risorse di AWS Batch calcolo predefinita riserva inoltre 32 MiB di memoria per l'agente container Amazon ECS e altri processi di sistema critici. Questa memoria non è disponibile per l'allocazione dei lavori. Per ulteriori informazioni, consulta Memoria di sistema di riserva.

L'agente del container di Amazon ECS utilizza la funzione Docker `ReadMemInfo()` per eseguire una query sulla memoria totale disponibile per il sistema operativo. Linux fornisce utilità da riga di comando per determinare la memoria totale.

Example- Determinare la memoria totale in Linux

Il free comando restituisce la memoria totale riconosciuta dal sistema operativo.

```
$ free -b
```

Di seguito è riportato un esempio di output per un'`m4.large`istanza 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. Ciò significa che sono disponibili 7985 MiB per le attività.

Argomenti

- Memoria di sistema di riserva
- Tutorial: Visualizza la memoria delle risorse di calcolo
- Considerazioni su AWS Batch memoria e vCPU per Amazon EKS

Memoria di sistema di riserva

Se si occupa tutta la memoria di una risorsa di elaborazione con i propri lavori, è possibile che tali processi abbiano a che fare con processi di sistema critici per la memoria e che possano causare un errore di sistema. L'agente container Amazon ECS fornisce una variabile di configurazione `ECS_RESERVED_MEMORY` denominata. Puoi usare questa variabile di configurazione per rimuovere un numero specifico di MiB di memoria dal pool allocato ai tuoi lavori. In questo modo si riserva la memoria per i processi di sistema critici.

L'AMI della risorsa di AWS Batch calcolo predefinita riserva 32 MiB di memoria per l'agente container Amazon ECS e altri processi di sistema critici. Consigliamo di riservare un buffer di memoria del 5% per l'agente container Amazon ECS e altri processi di sistema critici.

Tutorial: Visualizza la memoria delle risorse di calcolo

Puoi visualizzare la quantità di memoria con cui viene registrata una risorsa di calcolo nella console Amazon ECS o con il funzionamento dell'DescribeContainerInstancesAPI. Se stai cercando di massimizzare l'utilizzo delle risorse fornendo ai tuoi job quanta più memoria possibile per un particolare tipo di istanza, puoi osservare la memoria disponibile per quella risorsa di elaborazione e quindi assegnare ai job quella quantità di memoria.

Per visualizzare la memoria delle risorse di calcolo

1. Apri la console alla https://console.aws.amazon.com/ecs/v2.
2. Scegli Cluster, quindi scegli il cluster che ospita le risorse di elaborazione da visualizzare.

   Il nome del cluster dell'ambiente di calcolo inizia con il nome dell'ambiente di calcolo.
3. Scegli Infrastruttura.
4. In Istanze di container, scegli l'istanza del contenitore.
5. La sezione Risorse e reti mostra la memoria registrata e disponibile per la risorsa di calcolo.

   Il valore di memoria della capacità totale è quello che la risorsa di elaborazione ha registrato con Amazon ECS al momento del primo avvio, mentre il valore di memoria disponibile è quello che non è già stato allocato ai lavori.

Considerazioni su AWS Batch memoria e vCPU per Amazon EKS

In AWS Batch Amazon EKS, puoi specificare le risorse messe a disposizione di un contenitore. Ad esempio, è possibile specificare `requests` o `limits` valori per vCPU e risorse di memoria.

Di seguito sono riportati i vincoli per specificare le risorse vCPU:

- È necessario specificare almeno una vCPU `requests` o `limits` un valore.
- Un'unità vCPU equivale a un core fisico o virtuale.
- Il valore vCPU deve essere immesso in numeri interi o in incrementi di 0,25.
- Il valore vCPU più piccolo valido è 0,25.
- Se vengono specificati entrambi, il `requests` valore deve essere inferiore o uguale al `limits` valore. In questo modo, è possibile configurare configurazioni vCPU sia soft che hard.
- I valori vCPU non possono essere specificati nel formato MilliCPU. Ad esempio, `100m` non è un valore valido.

- AWS Batch utilizza il `requests` valore per ridimensionare le decisioni. Se non `requests` viene specificato un valore, il `limits` valore viene copiato nel `requests` valore.

Di seguito sono riportati i vincoli per specificare le risorse di memoria:

- È necessario specificare almeno una memoria `requests` o un `limits` valore.
- I valori di memoria devono essere in mebibytes (MiBs).
- Se vengono specificati entrambi, il `requests` valore deve essere uguale al `limits` valore.
- AWS Batch utilizza il `requests` valore per ridimensionare le decisioni. Se non viene specificato un `requests` valore, il `limits` valore viene copiato nel `requests` valore.

Di seguito sono riportati i vincoli per specificare le risorse GPU:

- Se vengono specificati entrambi, il `requests` valore deve essere uguale al valore. `limits`
- AWS Batch utilizza il `requests` valore per ridimensionare le decisioni. Se non `requests` viene specificato un valore, il `limits` valore viene copiato nel `requests` valore.

Esempio: definizioni delle mansioni

Quanto segue AWS Batch su Amazon EKS Job Definition configura le condivisioni soft vCPU. Ciò consente ad AWS Batch Amazon EKS di utilizzare tutta la capacità vCPU per il tipo di istanza. Tuttavia, se ci sono altri job in esecuzione, al job viene assegnato un massimo di 2 v. CPUs La memoria è limitata a 2 GB.

```
{
   "jobDefinitionName": "MyJobOnEks_Sleep",
   "type": "container",
   "eksProperties": {
       "podProperties": {
           "containers": [
               {
                   "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                   "command": ["sleep", "60"],
                   "resources": {
                       "requests": {
                           "cpu": "2",
                           "memory": "2048Mi"
                       }
```

```
                    }
                }
            ]
        }
    }
}
```

La seguente definizione di processo AWS Batch su Amazon EKS ha un `request` valore 1 e assegna un massimo di 4 v CPUs al lavoro.

```
{
   "jobDefinitionName": "MyJobOnEks_Sleep",
   "type": "container",
   "eksProperties": {
       "podProperties": {
           "containers": [
               {
                   "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                   "command": ["sleep", "60"],
                   "resources": {
                       "requests": {
                           "cpu": "1"
                       },
                       "limits": {
                           "cpu": "4",
                           "memory": "2048Mi"
                       }
                   }
               }
           ]
       }
   }
}
```

La seguente definizione del processo AWS Batch su Amazon EKS imposta un valore vCPU 1 e un `limits` valore di memoria `limits` di 1 GB.

```
{
   "jobDefinitionName": "MyJobOnEks_Sleep",
   "type": "container",
   "eksProperties": {
       "podProperties": {
```

```
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["sleep", "60"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
                        }
                    }
                }
            ]
        }
    }
}
```

Quando AWS Batch traduce un lavoro AWS Batch su Amazon EKS in un pod Amazon EKS, AWS Batch copia il `limits` valore nel `requests` valore. Questo è se non viene specificato un `requests` valore. Quando inviate la definizione di lavoro di esempio precedente, il contenitore `spec` è il seguente.

```
apiVersion: v1
kind: Pod
...
spec:
  ...
  containers:
    - command:
        - sleep
        - 60
      image: public.ecr.aws/amazonlinux/amazonlinux:2
      resources:
        limits:
          cpu: 1
          memory: 1024Mi
        requests:
          cpu: 1
          memory: 1024Mi
      ...
```

Nodo, CPU e prenotazioni di memoria

AWS Batch si basa sulla logica predefinita del `bootstrap.sh` file per le prenotazioni di vCPU e memoria. Per ulteriori informazioni sul `bootstrap.sh` file, vedere bootstrap.sh. Per quanto riguarda le dimensioni della vCPU e delle risorse di memoria, considera gli esempi seguenti.

> **Note**
>
> Se nessuna istanza è in esecuzione, le prenotazioni di vCPU e memoria possono inizialmente AWS Batch influire sulla logica di scalabilità e sul processo decisionale. Dopo l'esecuzione delle istanze, AWS Batch regola le allocazioni iniziali.

Esempio: prenotazione della CPU del nodo

Il valore di prenotazione della CPU viene calcolato in millicore utilizzando il numero totale di v CPUs disponibili per l'istanza.

Numero vCPU	Percentuale riservata
1	6%
2	1%
3-4	0,5%
4 e versioni successive	0,25%

Utilizzando i valori precedenti, è vero quanto segue:

- Il valore di prenotazione della CPU per un'`c5.large`istanza con 2 v CPUs è 70 m. Viene calcolato nel modo seguente: (1*60) + (1*10) = 70 m.
- Il valore di riserva della CPU per un'`c5.24xlarge`istanza con 96 v CPUs è 310 m. Viene calcolato nel modo seguente: (1*60) + (1*10) + (2*5) + (92*2,5) = 310 m.

In questo esempio, sono disponibili 1930 unità vCPU millicore (calcolate 2000-70) per eseguire processi su un'istanza. `c5.large` Supponiamo che il processo richieda 2 (2*1000 m) unità vCPU, il

processo non si adatta a una singola istanza. `c5.large` Tuttavia, un lavoro che richiede unità `1.75` vCPU è adatto.

Esempio: prenotazione della memoria del nodo

Il valore di prenotazione della memoria viene calcolato in mebibyte utilizzando quanto segue:

- La capacità dell'istanza in mebibyte. Ad esempio, un'istanza da 8 GB è pari a 7.748. MiB
- Il `kubeReserved` valore. Il `kubeReserved` valore è la quantità di memoria da riservare ai demoni di sistema. Il `kubeReserved` valore viene calcolato nel modo seguente: ((11 * numero massimo di pod supportato dal tipo di istanza) + 255). Per informazioni sul numero massimo di pod supportato da un tipo di istanza, consulta .txt eni-max-pods
- Il valore. `HardEvictionLimit` Quando la memoria disponibile scende al di sotto del `HardEvictionLimit` valore, l'istanza tenta di eliminare i pod.

La formula per calcolare la memoria allocabile è la seguente: (*`instance_capacity_in_MiB`*) - (11 * ()) - 255 - (*`maximum_number_of_pods`*)). *`HardEvictionLimit value.`*

Un'`c5.large`istanza supporta fino a 29 pod. Per un'`c5.large`istanza da 8 GB con un `HardEvictionLimit` valore di 100 MiB, la memoria allocabile è 7074. MiB Viene calcolato nel modo seguente: (7748 - (11 * 29) -255 -100) = 7074 MiB. In questo esempio, un MiB job di 8.192 non rientra in questa istanza anche se si tratta di un'istanza 8 (). gibibyte GiB

DaemonSets

Quando si utilizzaDaemonSets, si consideri quanto segue:

- Se nessuna istanza di AWS Batch Amazon EKS è in esecuzione, DaemonSets può inizialmente influire sulla logica di AWS Batch scalabilità e sul processo decisionale. AWS Batch inizialmente alloca 0,5 unità vCPU e 500 MiB come previsto. DaemonSets Dopo l'esecuzione delle istanze, AWS Batch regola le allocazioni iniziali.
- Se a DaemonSet definisce limiti di vCPU o memoria, AWS Batch su Amazon EKS i job hanno meno risorse. Ti consigliamo di mantenere il numero DaemonSets di AWS Batch lavori assegnati al più basso possibile.

Ambienti di elaborazione Fargate

Fargate è una tecnologia che puoi utilizzare AWS Batch per eseguire container senza dover gestire server o cluster di istanze Amazon. EC2 Con 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 si eseguono lavori con le risorse Fargate, si impacchetta l'applicazione in contenitori, si specificano i requisiti di CPU e memoria, si definiscono le politiche di rete e IAM e si avvia l'applicazione. Ogni job Fargate ha il proprio limite di isolamento e non condivide il kernel sottostante, le risorse della CPU, le risorse di memoria o l'interfaccia elastica di rete con un altro lavoro.

Argomenti

Quando usare Fargate

Consigliamo di utilizzare Fargate nella maggior parte degli scenari. Fargate avvia e ridimensiona l'elaborazione per soddisfare al meglio i requisiti di risorse specificati per il contenitore. Con Fargate, non è necessario fornire troppo o pagare server aggiuntivi. Inoltre, non è necessario preoccuparsi delle specifiche dei parametri relativi all'infrastruttura, come il tipo di istanza. Quando l'ambiente di elaborazione deve essere ampliato, i lavori eseguiti con risorse Fargate possono iniziare più rapidamente. In genere, sono necessari alcuni minuti per avviare una nuova EC2 istanza Amazon. Tuttavia, è possibile eseguire il provisioning dei lavori eseguiti su Fargate in circa 30 secondi. Il tempo esatto richiesto dipende da diversi fattori, tra cui la dimensione dell'immagine del contenitore e il numero di lavori.

Tuttavia, ti consigliamo di utilizzare Amazon EC2 se i tuoi lavori richiedono uno dei seguenti requisiti:

- Più di 16 v CPUs
- Più di 120 gibibyte (GiB) di memoria
- UNA GPU

- Un'Amazon Machine Image (AMI) personalizzata
- Qualsiasi parametro LinuxParameters

Se hai un numero elevato di lavori, ti consigliamo di utilizzare l' EC2 infrastruttura Amazon. Ad esempio, se il numero di lavori in esecuzione contemporaneamente supera i limiti di limitazione di Fargate. Questo perché, con EC2, i posti di lavoro possono essere assegnati alle risorse a un ritmo più elevato rispetto alle EC2 risorse di Fargate. Inoltre, è possibile eseguire più lavori contemporaneamente quando si utilizza. EC2 Per ulteriori informazioni, consulta le quote dei servizi Fargate nella Amazon Elastic Container Service Developer Guide.

Definizioni di lavoro su Fargate

AWS Batch jobs on AWS Fargate non supporta tutti i parametri di definizione dei processi disponibili. Alcuni parametri non sono affatto supportati e altri si comportano in modo diverso per i lavori di Fargate.

L'elenco seguente descrive i parametri di definizione dei processi che non sono validi o altrimenti limitati nei lavori Fargate.

`platformCapabilities`

Deve essere specificato come`FARGATE`.

```
"platformCapabilities": [ "FARGATE" ]
```

`type`

Deve essere specificato come`container`.

```
"type": "container"
```

Parametri in `containerProperties`

`executionRoleArn`

Deve essere specificato per i lavori eseguiti su risorse Fargate. Per ulteriori informazioni, consulta Ruoli IAM per le attività nella Guida per sviluppatori di Amazon Elastic Container Service.

```
"executionRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole"
```

`fargatePlatformConfiguration`

(Facoltativo, solo per le definizioni dei job di Fargate). Specificate la versione della piattaforma Fargate `LATEST` o una versione recente della piattaforma. I valori possibili per `platformVersion` sono `1.3.0` `1.4.0`, e `LATEST` (impostazione predefinita).

```
"fargatePlatformConfiguration": { "platformVersion": "1.4.0" }
```

`instanceType`, `ulimits`

Non applicabile ai lavori eseguiti su risorse Fargate.

`memory`, `vcpus`

Queste impostazioni devono essere specificate in `resourceRequirements`

`privileged`

O non specificate questo parametro oppure specificatelo`false`.

```
"privileged": false
```

`resourceRequirements`

I requisiti di memoria e vCPU devono essere specificati utilizzando i valori supportati. Le risorse GPU non sono supportate per i job eseguiti su risorse Fargate.

Se si utilizza GuardDuty Runtime Monitoring, si verifica un leggero sovraccarico di memoria per il GuardDuty security agent. Pertanto, il limite di memoria deve includere la dimensione del GuardDuty security agent. Per informazioni sui limiti di memoria del GuardDuty Security Agent, vedere Limiti di CPU e memoria nella Guida per l'GuardDuty utente. Per informazioni sulle best practice, consulta Come posso rimediare agli errori di memoria esaurita nelle mie attività di Fargate dopo aver abilitato il monitoraggio del runtime nella Amazon ECS Developer Guide.

```
"resourceRequirements": [
  {"type": "MEMORY", "value": "512"},
  {"type": "VCPU",   "value": "0.25"}
]
```

Parametri in `linuxParameters`

`devices`, `maxSwap`, `sharedMemorySize`, `swappiness`, `tmpfs`

Non applicabile ai lavori eseguiti su risorse Fargate.

Parametri in `logConfiguration`

`logDriver`

Solo `awslogs` e `splunk` sono supportati. Per ulteriori informazioni, consulta Usa il driver di registro awslogs.

Membri in `networkConfiguration`

`assignPublicIp`

Se la sottorete privata non dispone di un gateway NAT collegato per inviare traffico a Internet, `assignPublicIp` deve essere "»ENABLED. Per ulteriori informazioni, consulta AWS Batch Ruolo di esecuzione IAM.

Job in coda a Fargate

AWS Batch le code di lavoro sono sostanzialmente AWS Fargate invariate. L'unica restrizione è che gli ambienti di elaborazione elencati `computeEnvironmentOrder` devono essere tutti ambienti di calcolo Fargate (o). `FARGATE` `FARGATE_SPOT` EC2 e gli ambienti di elaborazione Fargate non possono essere combinati.

Ambienti di calcolo su Fargate

AWS Batch gli ambienti di calcolo attivi AWS Fargate non supportano tutti i parametri dell'ambiente di calcolo disponibili. Alcuni parametri non sono affatto supportati. Altri hanno requisiti specifici per Fargate.

L'elenco seguente descrive i parametri dell'ambiente di calcolo che non sono validi o altrimenti limitati nei lavori Fargate.

`type`

Questo parametro deve essere. `MANAGED`

```
"type": "MANAGED"
```

Parametri nell'`computeResources`oggetto

`allocationStrategy`, `bidPercentage`, `desiredvCpus`, `imageId`, `instanceTypes`, `ec2Configuration`, `ec2KeyPair`, `instanceRole`, `launchTemplate`, `minvCpus`, `placementGroup`, `spotIamFleetRole`

Questi non sono applicabili agli ambienti di calcolo Fargate e non possono essere forniti.

`subnets`

Se alle sottoreti elencate in questo parametro non sono collegati gateway NAT, il `assignPublicIp` parametro nella definizione del processo deve essere impostato su. `ENABLED`

`tags`

Questo non è applicabile agli ambienti di calcolo Fargate e non può essere fornito. Per specificare i tag per gli ambienti di calcolo Fargate, utilizzate il `tags` parametro che non si trova nell'oggetto. `computeResources`

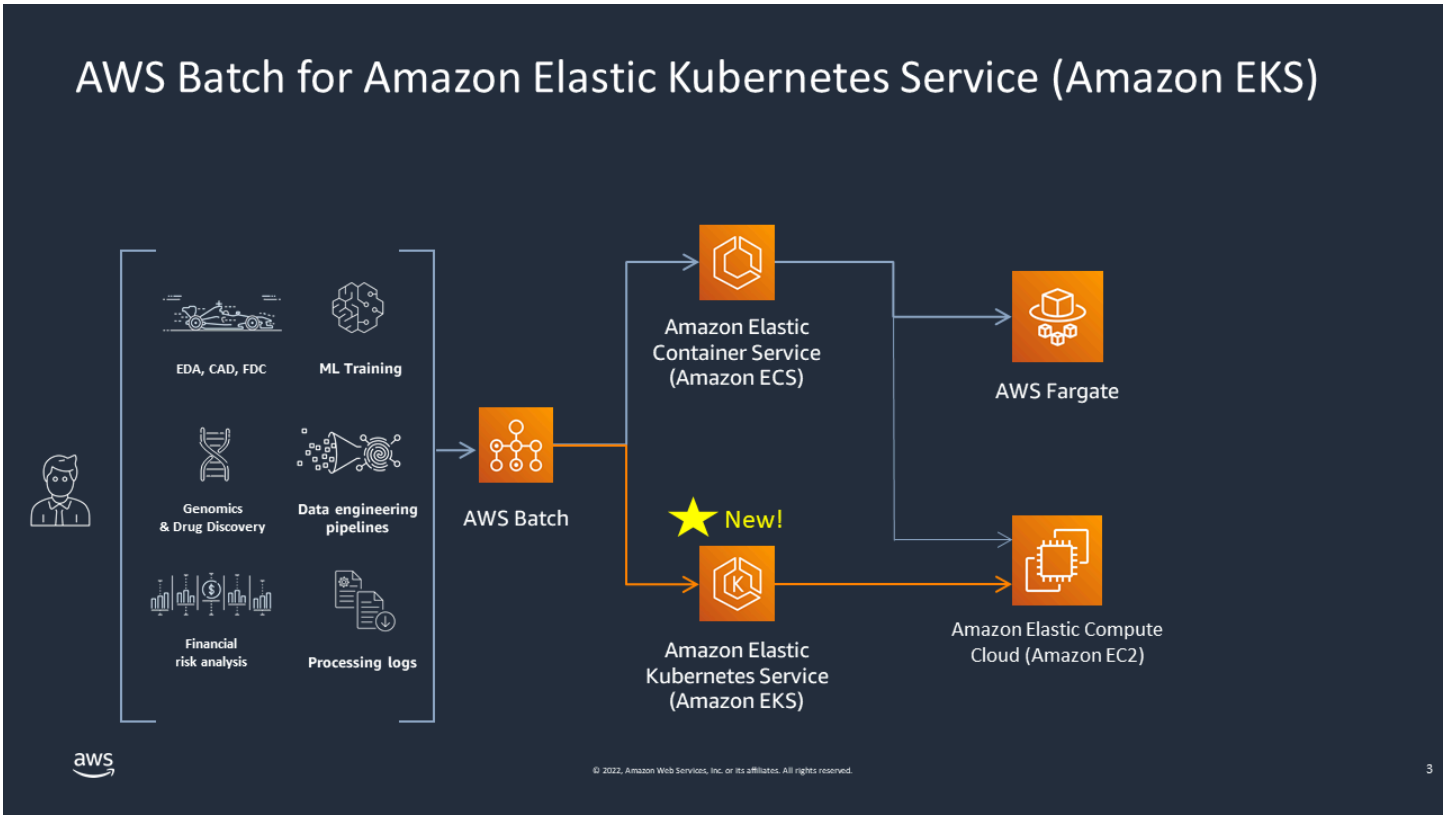
`type`

Questo deve essere `FARGATE` o `FARGATE_SPOT`.

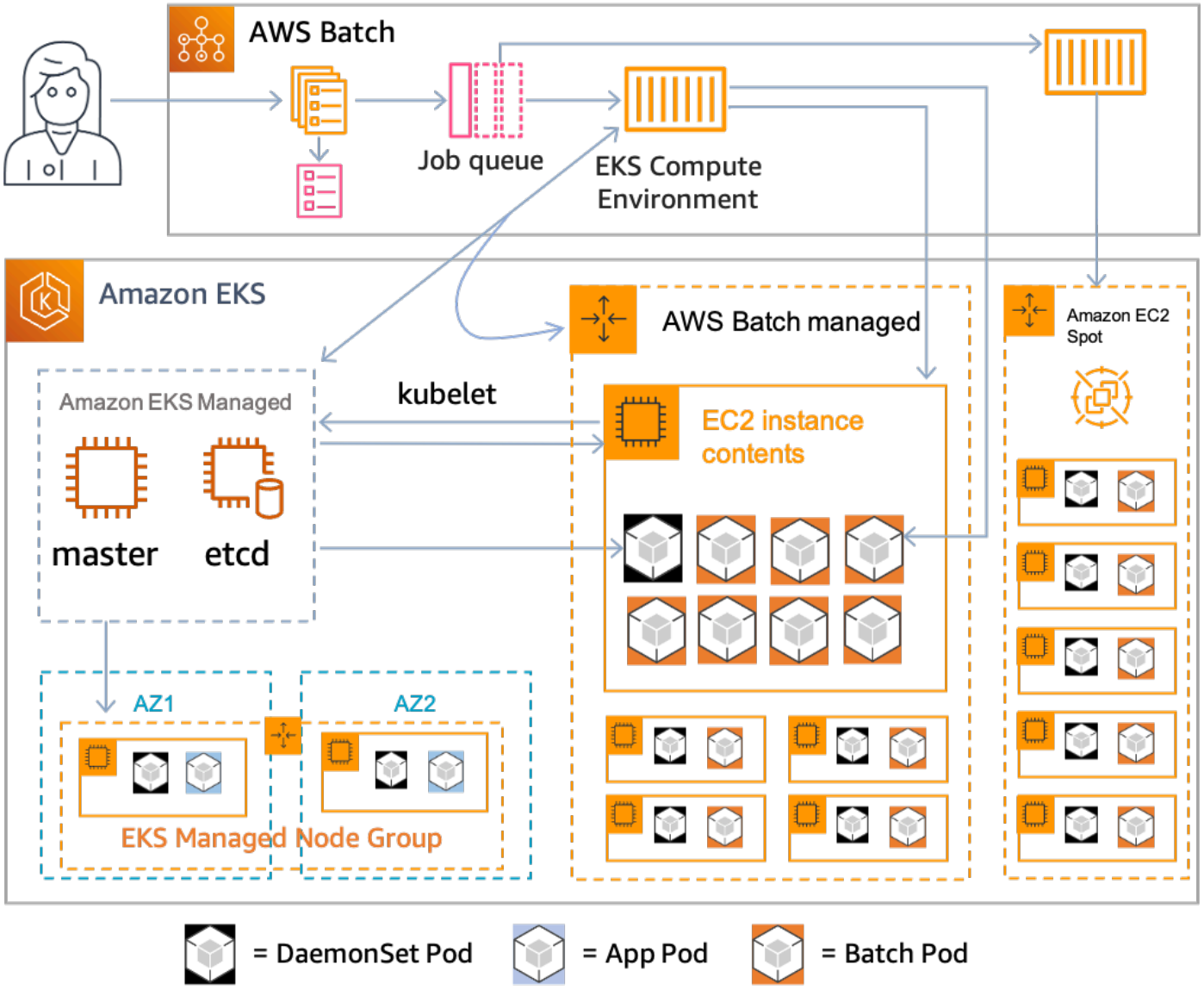
```
"type": "FARGATE_SPOT"
```

Ambienti di elaborazione Amazon EKS

Guida introduttiva ad AWS Batch Amazon EKSfornisce una breve guida alla creazione di ambienti di calcolo EKS. Questa sezione fornisce maggiori dettagli sugli ambienti di calcolo Amazon EKS.



AWS Batch semplifica i carichi di lavoro in batch sui cluster Amazon EKS fornendo funzionalità batch gestite. Ciò include l'accodamento, il monitoraggio delle dipendenze, la gestione delle priorità e dei nuovi tentativi di lavoro, la gestione dei pod e la scalabilità dei nodi. AWS Batch può gestire più zone di disponibilità e più tipi e dimensioni di EC2 istanze Amazon. AWS Batch integra diverse best practice di Amazon EC2 Spot per eseguire i carichi di lavoro in modo tollerante ai guasti, consentendo un minor numero di interruzioni. Puoi utilizzarlo AWS Batch per eseguire una manciata di lavori durante la notte o milioni di lavori cruciali con sicurezza.



AWS Batch è un servizio gestito che orchestra carichi di lavoro in batch nei Kubernetes cluster gestiti da Amazon Elastic Kubernetes Service (Amazon EKS). AWS Batch esegue questa orchestrazione all'esterno dei cluster utilizzando un modello «overlay». Poiché AWS Batch si tratta di un servizio gestito, non ci sono Kubernetes componenti (ad esempio, operatori o risorse personalizzate) da installare o gestire nel cluster. AWS Batch richiede solo che il cluster sia configurato con Role-Based Access Controls (RBAC) che consentono di comunicare con AWS Batch il server API. Kubernetes AWS Batch chiamate Kubernetes APIs per creare, monitorare ed eliminare Kubernetes pod e nodi.

AWS Batch dispone di una logica di scalabilità integrata per scalare Kubernetes i nodi in base al carico della coda di lavoro con ottimizzazioni in termini di allocazione della capacità lavorativa. Quando la coda dei processi è vuota, riduce AWS Batch i nodi alla capacità minima impostata, che per impostazione predefinita è zero. AWS Batch gestisce l'intero ciclo di vita di questi nodi e decora i

nodi con etichette e macchie. In questo modo, altri Kubernetes carichi di lavoro non vengono collocati sui nodi gestiti da. AWS Batch L'eccezione è che possono indirizzare AWS Batch i nodi per fornire il monitoraggio e altre funzionalità necessarie per la corretta esecuzione dei lavori. `DaemonSets` Inoltre, AWS Batch non esegue job, in particolare pod, su nodi del cluster che non gestisce. In questo modo, è possibile utilizzare logiche e servizi di scalabilità separati per altre applicazioni del cluster.

Per inviare lavori AWS Batch, interagisci direttamente con l' AWS Batch API. AWS Batch traduce i lavori `podspecs` e quindi crea le richieste per posizionare i pod sui nodi gestiti da AWS Batch nel tuo cluster Amazon EKS. Puoi utilizzare strumenti come visualizzare i pod e `kubectl` i nodi in esecuzione. Quando un pod ha completato la sua esecuzione, AWS Batch elimina il pod che ha creato per mantenere un carico inferiore sul Kubernetes sistema.

Puoi iniziare collegando un cluster Amazon EKS valido con AWS Batch. Quindi allega una coda di AWS Batch lavoro e registra una definizione di processo Amazon EKS utilizzando attributi `podspec` equivalenti. Infine, invia i lavori utilizzando l'operazione SubmitJobAPI che fa riferimento alla definizione del lavoro. Per ulteriori informazioni, consulta Guida introduttiva ad AWS Batch Amazon EKS.

Amazon EKS

Argomenti

- AMI Amazon EKS predefinita
- Ambienti AMI misti
- Versioni di Kubernetes supportate
- Aggiorna la Kubernetes versione dell'ambiente di calcolo
- Responsabilità condivisa dei nodi Kubernetes
- Esegui un DaemonSet su nodi AWS Batch gestiti
- Personalizza i modelli di lancio di Amazon EKS
- Come eseguire l'aggiornamento da EKS a EKS 023 AL2 AL2

AMI Amazon EKS predefinita

Quando crei un ambiente di calcolo Amazon EKS, non è necessario specificare un'Amazon Machine Image (AMI). AWS Batch seleziona un'AMI ottimizzata per Amazon EKS in base alla Kubernetes versione e ai tipi di istanza specificati nella CreateComputeEnvironmentrichiesta. In generale, ti consigliamo di utilizzare la selezione AMI predefinita. Per ulteriori informazioni su Amazon EKS

optimized AMIs, consulta Amazon Linux ottimizzato per Amazon EKS AMIs nella Guida per l'utente di Amazon EKS.

> **Important**
>
> Amazon Linux 2023 AMIs è l'impostazione predefinita AWS Batch per Amazon EKS. AWS terminerà il supporto per Amazon EKS AL2 ottimizzato e AL2 accelerato AMIs a partire dal 26/11/25. Puoi continuare a utilizzare Amazon Linux 2 ottimizzato per Amazon EKS AWS Batch fornito AMIs sui tuoi ambienti di calcolo Amazon EKS oltre la end-of-support data del 26/11/25, tuttavia questi ambienti di elaborazione non riceveranno più nuovi aggiornamenti software, patch di sicurezza o correzioni di bug. AWSPer ulteriori informazioni sull'aggiornamento da 023 a 023, consulta la Guida per l'utente. AL2 AL2 Come eseguire l'aggiornamento da EKS a EKS 023 AL2 AL2AWS Batch

Esegui il comando seguente per vedere quale tipo di AMI è AWS Batch stato selezionato per il tuo ambiente di calcolo Amazon EKS. L'esempio seguente è un tipo di istanza non GPU.

```
# compute CE example: indicates Batch has chosen the AL2 x86 or ARM EKS 1.32 AMI,
 depending on instance types
    $ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
        | jq '.computeEnvironments[].computeResources.ec2Configuration'
    [
      {
        "imageType": "EKS_AL2",
        "imageKubernetesVersion": "1.32"
      }
    ]
```

L'esempio seguente è un tipo di istanza GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.32 AMI
    $ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
        | jq '.computeEnvironments[].computeResources.ec2Configuration'
    [
      {
        "imageType": "EKS_AL2_NVIDIA",
        "imageKubernetesVersion": "1.32"
      }
    ]
```

Ambienti AMI misti

Puoi utilizzare le sostituzioni dei modelli di avvio per creare ambienti di calcolo con Amazon Linux 2 () AL2 e Amazon Linux 2023 (AL2023). AMIs Ciò è utile per utilizzare architetture diverse AMIs per diverse o durante i periodi di migrazione durante la transizione dalla versione 023. AL2 AL2

> **Note**
>
> AWS terminerà il supporto per Amazon EKS AL2 ottimizzato e AL2 accelerato AMIs a partire dal 26/11/25. Sebbene tu possa continuare a utilizzare Amazon Linux 2 ottimizzato per Amazon EKS AWS Batch fornito AMIs sui tuoi ambienti di calcolo Amazon EKS oltre la end-of-support data del 26/11/25, questi ambienti di elaborazione non riceveranno più nuovi aggiornamenti software, patch di sicurezza o correzioni di bug. AWS Gli ambienti AMI misti possono essere utili durante il periodo di transizione, consentendoti di migrare gradualmente i carichi di lavoro alla versione AL2 023 mantenendo la compatibilità con i carichi di lavoro basati esistenti AL2.

Esempio di configurazione che utilizza entrambi i tipi di AMI:

```
{
  "computeResources": {
    "launchTemplate": {
      "launchTemplateId": "TemplateId",
      "version": "1",
      "userdataType": "EKS_BOOTSTRAP_SH",
      "overrides": [
        {
          "instanceType": "c5.large",
          "imageId": "ami-al2-custom",
          "userdataType": "EKS_BOOTSTRAP_SH"
        },
        {
          "instanceType": "c6a.large",
          "imageId": "ami-al2023-custom",
          "userdataType": "EKS_NODEADM"
        }
      ]
    },
    "instanceTypes": ["c5.large", "c6a.large"]
  }
```

```
}
```

Versioni di Kubernetes supportate

AWS Batch su Amazon EKS attualmente supporta le seguenti Kubernetes versioni:

- `1.33`
- `1.32`
- `1.31`
- `1.30`
- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`

Potresti visualizzare un messaggio di errore simile al seguente quando utilizzi l'operazione `CreateComputeEnvironment` API o l'operazione `UpdateComputeEnvironment` API per creare o aggiornare un ambiente di calcolo. Questo problema si verifica se si specifica una versione non supportata Kubernetes in. `EC2Configuration`

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Per risolvere questo problema, elimina l'ambiente di calcolo e quindi ricrealo con una versione supportata. Kubernetes

Puoi eseguire un aggiornamento di versione minore sul tuo cluster Amazon EKS. Ad esempio, puoi aggiornare il cluster da `1.xx` a `1.yy` anche se la versione secondaria non è supportata.

Tuttavia, lo stato dell'ambiente di calcolo potrebbe cambiare `INVALID` dopo un aggiornamento della versione principale. Ad esempio, se si esegue un aggiornamento della versione principale da `1.xx` a`2.yy`. Se la versione principale non è supportata da AWS Batch, viene visualizzato un messaggio di errore analogo al seguente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Aggiorna la Kubernetes versione dell'ambiente di calcolo

Con AWS Batch, puoi aggiornare la Kubernetes versione di un ambiente di calcolo per supportare gli upgrade dei cluster Amazon EKS. La Kubernetes versione di un ambiente di calcolo è la versione AMI di Amazon EKS per i Kubernetes nodi che vengono AWS Batch avviati per eseguire processi. Puoi eseguire un upgrade di Kubernetes versione sui loro nodi Amazon EKS prima o dopo aver aggiornato la versione del piano di controllo del cluster Amazon EKS. Ti consigliamo di aggiornare i nodi dopo aver aggiornato il piano di controllo. Per ulteriori informazioni, consulta Aggiornamento di una Kubernetes versione del cluster Amazon EKS nella Guida per l'utente di Amazon EKS.

Per aggiornare la Kubernetes versione di un ambiente di calcolo, utilizza l'operazione UpdateComputeEnvironmentAPI.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Responsabilità condivisa dei nodi Kubernetes

La manutenzione degli ambienti di elaborazione è una responsabilità condivisa.

- Non modificare o rimuovere AWS Batch nodi, etichette, tinte, namespace, modelli di avvio o gruppi con ridimensionamento automatico. Non aggiungere macchie ai nodi gestiti. AWS Batch Se apporti una di queste modifiche, l'ambiente di calcolo non può essere supportato e si verificano errori, comprese le istanze inattive.
- Non indirizzate i pod verso nodi gestiti. AWS Batch Se indirizzate i pod verso i nodi gestiti, si verificheranno problemi di scalabilità e code di lavoro bloccate. Esegui carichi di lavoro che non vengono utilizzati AWS Batch su nodi autogestiti o gruppi di nodi gestiti. Per ulteriori informazioni, consulta Gruppi di nodi gestiti nella Guida per l'utente di Amazon EKS.
- Puoi scegliere come target DaemonSet a da eseguire su nodi AWS Batch gestiti. Per ulteriori informazioni, consulta Esegui un DaemonSet su nodi AWS Batch gestiti.

AWS Batch non aggiorna automaticamente l'ambiente di calcolo. AMIs È tua responsabilità aggiornarli. Esegui il seguente comando per aggiornare il tuo AMI AMIs alla versione più recente.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
```

```
    --compute-resources 'updateToLatestImageVersion=true'
```

AWS Batch non aggiorna automaticamente la Kubernetes versione. Esegui il comando seguente per aggiornare la Kubernetes versione del tuo ambiente informatico a*1.32*.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.32}]'
```

Quando si esegue l'aggiornamento a un'AMI o alla Kubernetes versione più recente, è possibile specificare se terminare i lavori quando vengono aggiornati (`terminateJobsOnUpdate`) e per quanto tempo attendere prima che un'istanza venga sostituita se i lavori in esecuzione non terminano (`jobExecutionTimeoutMinutes`.) Per ulteriori informazioni, consulta Aggiornare un ambiente di calcolo in AWS Batch e la policy di aggiornamento dell'infrastruttura (UpdatePolicy) impostata nell'operazione UpdateComputeEnvironmentAPI.

Esegui un DaemonSet su nodi AWS Batch gestiti

AWS Batch imposta le macchie sui nodi AWS Batch gestitiKubernetes. Puoi scegliere come target DaemonSet a da eseguire su nodi AWS Batch gestiti con quanto segue`tolerations`.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
```

Un altro modo per eseguire questa operazione è il seguente`tolerations`.

```
tolerations:
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoSchedule"
  - key: "batch.amazonaws.com/batch-node"
    operator: "Exists"
    effect: "NoExecute"
```

Personalizza i modelli di lancio di Amazon EKS

AWS Batch su Amazon EKS supporta i modelli di lancio. Esistono dei vincoli su ciò che il modello di lancio può fare.

Important

- Per EKS AL2 AMIs, corre AWS Batch . `/etc/eks/bootstrap.sh` Non eseguirlo `/etc/eks/bootstrap.sh` nel modello o negli cloud-init user-data script di lancio. È possibile aggiungere parametri aggiuntivi oltre al `--kubelet-extra-args` parametro di bootstrap.sh. A tale scopo, impostate la `AWS_BATCH_KUBELET_EXTRA_ARGS` variabile nel `/etc/aws-batch/batch.config` file. Vedi l'esempio seguente per i dettagli.
- Per EKS AL2 023, AWS Batch utilizza il NodeConfigSpecformato EKS per inserire le istanze nel cluster EKS. AWS Batch vengono ClusterDetailscompilati NodeConfigSpecper il cluster EKS e non è necessario specificarli.

Note

Ti consigliamo di non impostare nessuna delle seguenti NodeConfigSpecimpostazioni nel modello di lancio poiché AWS Batch sostituiranno i tuoi valori. Per ulteriori informazioni, consulta Responsabilità condivisa dei nodi Kubernetes.

- `Taints`
- `Cluster Name`
- `apiServerEndpoint`
- `certificatAuthority`
- `CIDR`
- Non creare etichette con il prefisso `batch.amazonaws.com/`

Note

Se il modello di avvio viene modificato dopo la CreateComputeEnvironmentchiamata, UpdateComputeEnvironmentdeve essere chiamato per valutare la versione del modello di avvio da sostituire.

Argomenti

- Aggiungi argomenti kubelet aggiuntivi

- Configura il runtime del contenitore
- Montare un volume Amazon EFS
- IPv6 supporto

Aggiungi argomenti **kubelet** aggiuntivi

AWS Batch supporta l'aggiunta di argomenti aggiuntivi al `kubelet` comando. Per l'elenco dei parametri supportati, kubeletconsultate la Kubernetesdocumentazione. Nell'esempio seguente per EKS AL2 AMIs, *`--node-labels mylabel=helloworld`* viene aggiunto alla `kubelet` riga di comando.

```
MIME-Version: 1.0
      Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

      --==MYBOUNDARY==
      Content-Type: text/x-shellscript; charset="us-ascii"

      #!/bin/bash
      mkdir -p /etc/aws-batch

      echo AWS_BATCH_KUBELET_EXTRA_ARGS=\"--node-labels mylabel=helloworld\" >> /etc/
aws-batch/batch.config

      --==MYBOUNDARY==--
```

Per EKS AL2 023 AMIs il formato del file è YAML. Per l'elenco dei parametri supportati, NodeConfigSpecconsultate la documentazione. Kubernetes Nell'esempio seguente per EKS AL2 023 AMIs, *`--node-labels mylabel=helloworld`* viene aggiunto alla `kubelet` riga di comando.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: application/node.eks.aws

apiVersion: node.eks.aws/v1alpha1
kind: NodeConfig
spec:
  kubelet:
    flags:
```

```
    - --node-labels=mylabel=helloworld

--==MYBOUNDARY==--
```

Configura il runtime del contenitore

È possibile utilizzare la variabile di AWS Batch `CONTAINER_RUNTIME` ambiente per configurare il runtime del contenitore su un nodo gestito. L'esempio seguente imposta il runtime del contenitore su `containerd` when `bootstrap.sh` run. Per ulteriori informazioni, `containerd`consultate la Kubernetesdocumentazione.

Se si utilizza un'`EKS_AL2023_NVIDIA`AMI ottimizzata `EKS_AL2023` o non è necessario specificare il runtime del contenitore poiché è supportato solo containerd.

> **Note**
>
> La variabile di `CONTAINER_RUNTIME` ambiente è equivalente all'`--container-runtime`opzione di. `bootstrap.sh` Per ulteriori informazioni, Optionsconsulta la Kubernetesdocumentazione.

```
MIME-Version: 1.0
      Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

      --==MYBOUNDARY==
      Content-Type: text/x-shellscript; charset="us-ascii"

      #!/bin/bash
      mkdir -p /etc/aws-batch

      echo CONTAINER_RUNTIME=containerd >> /etc/aws-batch/batch.config

      --==MYBOUNDARY==--
```

Montare un volume Amazon EFS

Puoi utilizzare i modelli di avvio per montare volumi sul nodo. Nell'esempio seguente, vengono utilizzate `runcmd` le impostazioni `cloud-config` `packages` e. Per ulteriori informazioni, consulta gli esempi di configurazione di Cloud nella cloud-initdocumentazione.

```
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="==MYBOUNDARY=="

--==MYBOUNDARY==
Content-Type: text/cloud-config; charset="us-ascii"

packages:
- amazon-efs-utils

runcmd:
- file_system_id_01=fs-abcdef123
- efs_directory=/mnt/efs

- mkdir -p ${efs_directory}
- echo "${file_system_id_01}:/ ${efs_directory} efs _netdev,noresvport,tls,iam 0 0"
 >> /etc/fstab
- mount -t efs -o tls ${file_system_id_01}:/ ${efs_directory}

--==MYBOUNDARY==--
```

Per utilizzare questo volume nel job, è necessario aggiungerlo nel parametro EksProperties a. RegisterJobDefinition L'esempio seguente è una parte importante della definizione del processo.

```
{
    "jobDefinitionName": "MyJobOnEks_EFS",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "containers": [
                {
                    "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                    "command": ["ls", "-la", "/efs"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi"
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "efs-volume",
                            "mountPath": "/efs"
                        }
```

```
                    ]
                }
            ],
            "volumes": [
                {
                    "name": "efs-volume",
                    "hostPath": {
                        "path": "/mnt/efs"
                    }
                }
            ]
        }
    }
}
```

Nel nodo, il volume Amazon EFS è montato nella `/mnt/efs` directory. Nel contenitore per il job Amazon EKS, il volume viene montato nella `/efs` directory.

IPv6 supporto

AWS Batch supporta i cluster Amazon EKS con IPv6 indirizzi. Non sono richieste personalizzazioni per AWS Batch il supporto. Tuttavia, prima di iniziare, ti consigliamo di rivedere le considerazioni e le condizioni descritte nella sezione Assegnazione di IPv6 indirizzi a pod e servizi nella Amazon EKS User Guide.

Come eseguire l'aggiornamento da EKS a EKS 023 AL2 AL2

Gli Amazon EKS ottimizzati AMIs sono disponibili in due famiglie basate su Amazon Linux 2 (AL2) e Amazon Linux 2023 (AL2023). AL2023 è un sistema operativo basato su Linux progettato per fornire un ambiente sicuro, stabile e ad alte prestazioni per le tue applicazioni cloud. Per ulteriori informazioni sulle differenze tra AL2 e AL2 023, consulta Upgrade da Amazon Linux 2 ad Amazon Linux 2023 nella Amazon EKS User Guide.

> **Important**
>
> AWS terminerà il supporto per Amazon EKS AL2 ottimizzato e AL2 accelerato AMIs a partire dal 26/11/25. Consigliamo di migrare gli ambienti di calcolo AWS Batch Amazon EKS su Amazon Linux 2023 prima del 26/11/25 per mantenere prestazioni e sicurezza ottimali. Sebbene tu possa continuare a utilizzare Amazon Linux 2 ottimizzato per Amazon EKS AWS Batch fornito AMIs sui tuoi ambienti di calcolo Amazon EKS oltre la end-of-support data del

26/11/25, questi ambienti di elaborazione non riceveranno più nuovi aggiornamenti software, patch di sicurezza o correzioni di bug. AWS Successivamente end-of-life, è tua responsabilità mantenere questi ambienti di calcolo sull'AMI Amazon Linux 2 ottimizzata per Amazon EKS.

A seconda di come è configurato l'ambiente di calcolo, puoi utilizzare uno dei seguenti percorsi di aggiornamento da 0 AL2 a AL2 023.

Effettua l'aggiornamento utilizzando Ec2Configuration. ImageType

- Se non stai utilizzando un modello di avvio o sostituzioni di modelli di avvio, modifica Ec2Configuraiton. ImageTypea o e poi esegui. `EKS_AL2023` `EKS_AL2023_NVIDIA` UpdateComputeEnvironment
- Se si specifica una configurazione Ec2. ImageIdOverridequindi Ec2Configuration. ImageTypedeve corrispondere al tipo di AMI specificato in Ec2Configuration. ImageIdOverride.

  In caso di mancata corrispondenza `ImageIdOverrideImageType`, il nodo non entrerà a far parte del cluster.

Esegui l'aggiornamento utilizzando i modelli di avvio

- Se sono stati definiti argomenti `kubelet` aggiuntivi in un modello di avvio o in un modello di avvio sovrascritto, è necessario aggiornarli al nuovo formato degli argomenti `kubelet` aggiuntivi.

  Se il formato degli argomenti `kubelet` aggiuntivi non corrisponde, gli argomenti aggiuntivi non vengono applicati.
- Per AL2 023 AMIs, containerd è l'unico runtime di container supportato. Non è necessario specificare il runtime del contenitore `EKS_AL2023` nel modello di avvio.

  Non è possibile specificare un runtime del contenitore personalizzato con`EKS_AL2023`.
- Se utilizzi un modello di avvio o un override del modello di avvio che specifica un AMI basato su, `EKS_AL2023` devi impostare UserDataType su. `EKS_NODEADM`

  Se non combini con l'AMI `userdataType` e, il nodo non entrerà a far parte del cluster EKS.

Ambienti di servizio per AWS Batch

Gli ambienti di servizio consentono l'integrazione con AWS Batch SageMaker l'intelligenza artificiale. Un ambiente di servizio contiene i parametri di configurazione specifici dell' SageMaker IA necessari AWS Batch per inviare e gestire i lavori di SageMaker formazione, fornendo al contempo funzionalità AWS Batch di coda, pianificazione e gestione delle priorità.

Grazie agli ambienti di servizio, i data scientist e gli ingegneri ML possono inviare lavori di SageMaker formazione con priorità alle code di lavoro di assistenza. Questa integrazione elimina la necessità di coordinare manualmente i carichi di lavoro ML, previene spese eccessive accidentali e migliora l'utilizzo delle risorse nei flussi di lavoro di machine learning dell'organizzazione.

Argomenti

In cosa si trovano gli ambienti di servizio AWS Batch

Un ambiente di servizio è una AWS Batch risorsa che contiene i parametri di configurazione necessari per l'integrazione AWS Batch con l' SageMaker intelligenza artificiale. Gli ambienti di servizio AWS Batch consentono di inviare e gestire i lavori AWS Batch di SageMaker formazione fornendo al contempo funzionalità di coda, pianificazione e gestione delle priorità.

Gli ambienti di servizio risolvono le sfide comuni che i team di data science devono affrontare nella gestione dei carichi di lavoro di machine learning. Le organizzazioni spesso limitano il numero di istanze disponibili per i modelli di formazione per evitare spese eccessive accidentali, rispettare i vincoli di budget, risparmiare sui costi con le istanze riservate o utilizzare tipi di istanze specifici per i carichi di lavoro. Tuttavia, i data scientist potrebbero voler eseguire più carichi di lavoro contemporaneamente di quanto sia possibile con le istanze loro allocate, il che richiede un coordinamento manuale per decidere quali carichi di lavoro eseguire e quando.

Questa sfida di coordinamento ha un impatto sulle organizzazioni di tutte le dimensioni, dai team con pochi data scientist alle operazioni su larga scala. Man mano che le organizzazioni crescono,

la complessità aumenta, richiedendo più tempo per gestire il coordinamento dei carichi di lavoro e spesso richiedendo il coinvolgimento degli amministratori dell'infrastruttura. Queste attività manuali fanno perdere tempo e riducono l'efficienza delle istanze, con conseguenti costi reali per i clienti.

Grazie agli ambienti di servizio, i data scientist e gli ingegneri ML possono inviare lavori di SageMaker formazione con priorità a code configurabili, garantendo l'esecuzione automatica dei carichi di lavoro senza intervento non appena le risorse sono disponibili. Questa integrazione sfrutta le ampie funzionalità AWS Batch di coda e pianificazione di cui dispone, consentendo ai clienti di personalizzare le proprie politiche di coda e pianificazione per soddisfare gli obiettivi della propria organizzazione.

In che modo gli ambienti di servizio interagiscono con altri componenti AWS Batch

Gli ambienti di servizio si integrano con altri AWS Batch componenti per consentire l'accodamento dei lavori di SageMaker formazione:

- Job queues: gli ambienti di servizio sono associati alle code di lavoro per consentire alla coda di elaborare i lavori di servizio per il job Training SageMaker
- Lavori di servizio: quando si invia un lavoro di servizio a una coda associata a un ambiente di servizio, AWS Batch utilizza la configurazione dell'ambiente per inviare il lavoro di formazione corrispondente SageMaker
- Politiche di pianificazione: gli ambienti di servizio utilizzano politiche di AWS Batch pianificazione per stabilire le priorità e gestire l'ordine di esecuzione dei lavori di formazione SageMaker

Questa integrazione consente di sfruttare le funzionalità AWS Batch di coda e pianificazione già esistenti, mantenendo al contempo la piena funzionalità e flessibilità dei lavori di formazione. SageMaker

Le migliori pratiche per gli ambienti di servizio

Gli ambienti di servizio forniscono funzionalità per la gestione dei lavori di SageMaker formazione su larga scala. Seguire queste best practice consente di ottimizzare costi, prestazioni ed efficienza operativa evitando al contempo problemi di configurazione comuni che possono influire sui flussi di lavoro di machine learning.

Quando pianificate la capacità dell'ambiente di servizio, prendete in considerazione le quote e i limiti specifici che si applicano alle richieste di lavoro relative alla SageMaker formazione. Ogni ambiente

di servizio ha un limite di capacità massima espresso in numero di istanze, che controlla direttamente il numero di job di SageMaker formazione che possono essere eseguiti contemporaneamente. La comprensione di questi limiti aiuta a prevenire la contesa delle risorse e garantisce tempi di esecuzione dei lavori prevedibili.

Le prestazioni ottimali dell'ambiente di servizio dipendono dalla comprensione delle caratteristiche uniche della pianificazione dei lavori di SageMaker formazione. A differenza dei tradizionali lavori containerizzati, i lavori di assistenza passano da uno `SCHEDULED` stato all'altro mentre l' SageMaker IA acquisisce e fornisce le istanze di formazione necessarie. Ciò significa che gli orari di inizio dei lavori possono variare in modo significativo in base alla disponibilità delle istanze e alla capacità regionale.

> **Important**
>
> Gli ambienti di servizio hanno quote specifiche che possono influire sulla capacità di scalare i carichi di lavoro di SageMaker formazione. È possibile creare fino a 50 ambienti di servizio per account, con ogni coda di lavoro che supporta solo un ambiente di servizio associato. Inoltre, il payload di Service Request per i singoli job è limitato a 10 KiB e l'`SubmitServiceJob`API è limitata a 5 transazioni al secondo per account. La comprensione di questi limiti durante la pianificazione della capacità previene vincoli di scalabilità imprevisti.

Un monitoraggio efficace degli ambienti di servizio richiede attenzione sia AWS Batch alle metriche dei servizi di SageMaker intelligenza artificiale. Le transizioni dello stato del lavoro forniscono informazioni preziose sulle prestazioni del sistema, in particolare sul tempo trascorso in `SCHEDULED` stato, che indica i modelli di disponibilità della capacità. Gli ambienti di servizio mantengono i propri stati del ciclo di vita simili agli ambienti di elaborazione, in fase di transizione e agli `CREATING` `DELETING` stati che devono essere `INVALID` monitorati per verificarne lo stato operativo. `VALID` Organizations con pratiche di monitoraggio consolidate in genere tengono traccia della profondità delle code, dei tassi di completamento dei lavori e dei modelli di utilizzo delle istanze per ottimizzare le configurazioni degli ambienti di servizio nel tempo.

Stati e ciclo di vita dell'ambiente di servizio in AWS Batch

Gli ambienti di servizio mantengono stati del ciclo di vita che indicano lo stato operativo attuale e la disponibilità a elaborare i lavori di formazione. SageMaker La comprensione di questi stati consente

di monitorare lo stato dell'ambiente di servizio, risolvere i problemi di configurazione e garantire un'elaborazione affidabile dei lavori. Il sistema di gestione statale segue gli schemi prestabiliti degli ambienti di elaborazione, soddisfacendo al contempo i requisiti unici dell'integrazione dei ruoli di SageMaker formazione.

Gli stati dell'ambiente di servizio vengono gestiti automaticamente AWS Batch in base alla convalida della configurazione, alla disponibilità delle risorse e ai controlli dello stato operativo. A differenza degli ambienti di elaborazione che gestiscono l'infrastruttura fisica, gli ambienti di servizio si concentrano sulla convalida della configurazione e sulla predisposizione all'integrazione con i servizi di intelligenza artificiale. SageMaker Le transizioni di stato consentono di verificare se l'ambiente di servizio è in grado di inviare e gestire correttamente i lavori di formazione. SageMaker

Definizioni dello stato dell'ambiente di servizio

Gli ambienti di servizio possono trovarsi in uno dei quattro stati possibili che indicano lo stato operativo corrente e la disponibilità a elaborare i lavori di SageMaker formazione. Ogni stato rappresenta una fase specifica del ciclo di vita dell'ambiente di servizio, dalla creazione iniziale alla disponibilità operativa fino all'eventuale eliminazione. La tabella seguente descrive ogni stato e il relativo significato:

Stato	Description
`CREATING`	Lo stato iniziale quando si crea un ambiente di servizio. Durante questo stato, AWS Batch convalida i parametri di configurazione e stabilisce l'integrazione con i servizi di SageMaker intelligenza artificiale. L'ambiente di servizio non è in grado di elaborare i lavori e le code di lavoro ad esso associate non accettera nno l'invio di lavori di servizio. Il processo di creazione viene in genere completato entro pochi secondi per ambienti di servizio configura ti correttamente.
`VALID`	Lo stato operativo indica che l'ambiente di servizio ha superato tutti i controlli di convalida della configurazione ed è pronto per l'elabora zione dei job di SageMaker formazione. Questo

Stato	Description
	stato indica che la configurazione dell'ambi ente di servizio è corretta, che tutte le autorizza zioni richieste sono disponibili e che è AWS Batch possibile inviare correttamente i lavori all' SageMaker IA per conto dell'utente. Gli ambienti di servizio trascorrono la maggior parte del loro ciclo di vita operativo in questo stato.
`INVALID`	Uno stato che indica che l'ambiente di servizio ha riscontrato un problema di configurazione o di autorizzazioni che ne impedisce l'elabora zione dei SageMaker lavori di formazione. Le code di lavoro associate ad ambienti di servizio non validi non possono elaborare nuovi invii di lavori di servizio finché i problemi sottostanti non vengono risolti.
`DELETING`	Lo stato che si verifica quando si richiede l'eliminazione di un ambiente di servizio. Durante questo stato, AWS Batch assicura che nessun lavoro di SageMaker formazione attivo sia associato all'ambiente ed esegue le operazioni di pulizia necessarie. Gli ambienti di servizio in questo stato non possono elaborare nuovi invii di lavoro e il processo di eliminazi one viene completato una volta che tutte le risorse associate sono state pulite correttam ente.

Transizioni di stato dell'ambiente di servizio

Le transizioni di stato dell'ambiente di servizio avvengono automaticamente in base alle modifiche alla configurazione, ai risultati della convalida e al monitoraggio dello stato operativo. Il AWS Batch servizio monitora continuamente lo stato dell'ambiente di servizio e aggiorna gli stati di conseguenza.

La comprensione di queste transizioni aiuta a prevedere quando entreranno in vigore le modifiche alla configurazione e a risolvere i problemi che causano stati non validi.

Dopo una corretta creazione e convalida, gli ambienti di servizio passano da a. `CREATING` `VALID` Questa transizione conferma che tutti i parametri di configurazione sono corretti, le autorizzazioni IAM richieste sono configurate correttamente e l'ambiente di servizio può integrarsi con successo con i servizi di SageMaker intelligenza artificiale. Una volta nello `VALID` stato, le code di lavoro associate possono iniziare a elaborare gli invii di lavori di servizio.

Gli ambienti di servizio passano `VALID` da `INVALID` quando la convalida della configurazione fallisce o quando le dipendenze non sono più disponibili. Ciò può verificarsi a causa di modifiche dei ruoli IAM, modifiche dei limiti di capacità che violano le quote o modifiche delle risorse esterne che influiscono sulla capacità di funzionamento dell'ambiente di servizio. Il campo status reason fornisce dettagli specifici sulla causa dello stato non valido.

Gli ambienti di servizio possono tornare `VALID` a `INVALID` una volta risolti i problemi sottostanti. Ciò potrebbe comportare l'aggiornamento delle autorizzazioni IAM, la correzione delle configurazioni di capacità o il ripristino dell'accesso alle risorse richieste. AWS La transizione avviene in genere automaticamente una volta AWS Batch rilevato che i problemi di configurazione sono stati risolti.

Crea un ambiente di servizio in AWS Batch

Prima di poter eseguire i lavori di SageMaker formazione in AWS Batch, è necessario creare un ambiente di servizio. Puoi creare un ambiente di servizio che contenga i parametri di configurazione necessari AWS Batch per l'integrazione con i servizi di SageMaker intelligenza artificiale e inviare lavori di SageMaker formazione per tuo conto.

Prerequisiti

Prima di creare un ambiente di servizio, assicurati di avere:

- Autorizzazioni IAM: autorizzazioni per creare e gestire ambienti di servizio. Per ulteriori informazioni, consulta AWS Batch Politiche, ruoli e autorizzazioni IAM.

Create a service environment (AWS Console)

Utilizza la AWS Batch console per creare un ambiente di servizio tramite l'interfaccia web.

Per creare un ambiente di servizio

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegli Crea ambiente, seleziona Ambiente di servizio.
4. Per la configurazione dell'ambiente di servizio scegli SageMaker AI.
5. In Nome, inserisci un nome univoco per il tuo ambiente di servizio. I caratteri validi sono a-z, A-Z, 0-9, trattini (-) e trattini bassi (_).
6. Per Numero massimo di istanze, inserisci il numero massimo di istanze di addestramento simultanee
7. (Facoltativo) Aggiungi tag selezionando Aggiungi tag e inserendo le coppie chiave-valore.
8. Scegli Next (Successivo).
9. Esamina i dettagli del nuovo ambiente di servizio e scegli Crea ambiente di servizio.

Create a service environment (AWS CLI)

Utilizzate il `create-service-environment` comando per creare un ambiente di servizio con la AWS CLI.

Per creare un ambiente di servizio

1. Creare un ambiente di servizio con i parametri di base richiesti:

```
aws batch create-service-environment \
    --service-environment-name my-sagemaker-service-env \
    --service-environment-type SAGEMAKER_TRAINING \
    --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10
```

2. (Facoltativo) Crea un ambiente di servizio con tag:

```
aws batch create-service-environment \
    --service-environment-name my-sagemaker-service-env \
    --service-environment-type SAGEMAKER_TRAINING \
    --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=10 \
    --tags team=data-science,project=ml-training
```

3. Verifica che l'ambiente di servizio sia stato creato correttamente:

```
aws batch describe-service-environments \
```

```
    --service-environment my-sagemaker-service-env
```

L'ambiente di servizio viene visualizzato nell'elenco Ambienti con uno `CREATING` stato. Una volta completata correttamente la creazione, lo stato cambia `VALID` e l'ambiente di servizio è pronto per l'aggiunta di una coda di lavori di assistenza in modo che l'ambiente di servizio possa iniziare a elaborare i lavori.

Aggiornare un ambiente di servizio in AWS Batch

È possibile aggiornare un ambiente di servizio per modificarne i limiti di capacità, modificarne lo stato operativo o aggiornare i tag delle risorse. Gli aggiornamenti dell'ambiente di servizio consentono di regolare la capacità in base ai requisiti del carico di lavoro di SageMaker formazione che cambiano o modificano le impostazioni operative senza ricreare l'ambiente. Prima di aggiornare un ambiente di servizio, è necessario comprendere quali parametri possono essere modificati e l'impatto delle modifiche sui processi in esecuzione.

È possibile modificare i limiti di capacità, lo stato o i tag di un ambiente di servizio.

Update a service environment (AWS Console)

Utilizza la AWS Batch console per aggiornare un ambiente di servizio tramite l'interfaccia web.

Per aggiornare un ambiente di servizio

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegli la scheda Ambiente di servizio.
4. Scegli l'ambiente di servizio da aggiornare.
5. Scegli Azioni, quindi scegli una delle seguenti opzioni:
   - Stato: scegli Abilita o Disabilita per modificare lo stato.
   - Limite di capacità: modifica il numero massimo di istanze
6. Scegli Salva modifiche per applicare le modifiche.

L'ambiente di servizio si aggiorna immediatamente. Controlla i dettagli dell'ambiente per confermare che le modifiche siano state applicate correttamente. Se hai disabilitato l'ambiente

di servizio, le code di lavoro associate interromperanno l'elaborazione di nuovi lavori di servizio finché non lo riabiliti.

Update a service environment (AWS CLI)

Utilizzate il `update-service-environment` comando per modificare un ambiente di servizio con la AWS CLI.

Per aggiornare i limiti di capacità dell'ambiente di servizio

1. Aggiornare il limite di capacità per un ambiente di servizio:

```
aws batch update-service-environment \
    --service-environment my-sagemaker-service-env \
    --capacity-limits capacityUnit=NUM_INSTANCES,maxCapacity=20
```

2. Verifica che l'aggiornamento sia stato applicato correttamente:

```
aws batch describe-service-environments \
    --service-environments my-sagemaker-service-env
```

Per aggiornare lo stato dell'ambiente di servizio

1. Disabilita un ambiente di servizio per interrompere l'elaborazione di nuovi lavori:

```
aws batch update-service-environment \
    --service-environment my-sagemaker-service-env \
    --state DISABLED
```

2. Riattiva un ambiente di servizio per riprendere l'elaborazione:

```
aws batch update-service-environment \
    --service-environment my-sagemaker-service-env \
    --state ENABLED
```

Gli aggiornamenti dell'ambiente di servizio hanno effetto immediato. Monitora lo stato dell'ambiente di servizio per garantire che gli aggiornamenti vengano completati correttamente prima di inviare nuovi lavori.

Eliminare un ambiente di servizio in AWS Batch

È possibile eliminare un ambiente di servizio quando non è più necessario per i lavori di SageMaker formazione. L'eliminazione di un ambiente di servizio rimuove la configurazione e impedisce l'invio di ulteriori lavori. Prima di eliminare un ambiente di servizio, assicuratevi che nessun lavoro di SageMaker formazione attivo dipenda da esso e che non vi siano code di lavoro associate all'ambiente di servizio.

Important

L'eliminazione dell'ambiente di servizio è irreversibile. Una volta eliminato, non è possibile ripristinare l'ambiente di servizio o la relativa configurazione. Se in futuro saranno necessarie funzionalità simili, sarà necessario creare un nuovo ambiente di servizio con le impostazioni richieste. Valuta la possibilità di disabilitare l'ambiente di servizio anziché eliminarlo se potrebbe essere necessario riattivarlo in un secondo momento.

Note

L'eliminazione di tutti gli ambienti di servizio nel tuo account non rimuove automaticamente il ruolo collegato ai servizi creato per l'integrazione con l'intelligenza artificiale. AWS Batch SageMaker Il ruolo collegato ai servizi rimane disponibile per la creazione di ambienti di servizio futuri. Se desideri rimuovere il ruolo collegato al servizio, devi eliminarlo separatamente utilizzando IAM dopo esserti assicurato che non esistano ambienti di servizio nel tuo account.

Prerequisiti per l'eliminazione

Prima di poter eliminare un ambiente di servizio, è necessario dissociare qualsiasi coda di lavoro di servizio e quindi disabilitare l'ambiente di servizio.

Prima di eliminare un ambiente di servizio:

- Controlla i lavori attivi: assicurati che nessun SageMaker processo di formazione sia attualmente in esecuzione nell'ambiente di servizio.
- Rivedi le code di lavoro: identifica le code di lavoro associate all'ambiente di servizio e associa la coda di lavoro a un ambiente di servizio diverso o disabilita ed elimina la coda dei lavori.

Gestione delle code di lavoro: le code di lavoro associate a un ambiente di servizio eliminato possono ancora esistere ma non possono elaborare i lavori di servizio. È necessario eliminare le code di lavoro non utilizzate o associarle a un ambiente di servizio diverso prima di eliminare l'ambiente di servizio originale.

Delete a service environment (AWS Console)

Utilizzate la AWS Batch console per eliminare un ambiente di servizio tramite l'interfaccia web.

Per eliminare un ambiente di servizio

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nel riquadro di navigazione, selezionare Compute environments (Ambienti di calcolo).
3. Scegli la scheda Ambiente di servizio, quindi scegli un ambiente di servizio.
4. Se l'ambiente di servizio è abilitato, scegli Azioni e poi Disabilita.
5. Una volta disabilitato l'ambiente di servizio, scegli Azioni, quindi Elimina.
6. Nella finestra di dialogo di conferma, scegli Conferma.

L'ambiente di servizio mostra uno `DELETING` stato durante l'eliminazione. Una volta completata l'eliminazione, l'ambiente di servizio scompare dall'elenco Ambienti.

Delete a service environment (AWS CLI)

Utilizzate il `delete-service-environment` comando per rimuovere un ambiente di servizio con la AWS CLI.

Per eliminare un ambiente di servizio

1. Verifica la presenza di code di lavoro associate all'ambiente di servizio:

```
aws batch describe-job-queues
```

Se sono presenti code di lavoro associate all'ambiente di servizio, è possibile dissociare la coda di lavoro dall'ambiente di servizio e associarla a un ambiente di servizio diverso oppure eliminare la coda dei lavori.

2. Disabilita l'ambiente di servizio:

```
aws batch update-service-environment \
    --service-environment my-sagemaker-service-env \
```

```
    --state DISABLED
```

3. Eliminare l'ambiente di servizio:

```
aws batch delete-service-environment \
    --service-environment my-sagemaker-service-env
```

4. Monitora il processo di eliminazione:

```
aws batch describe-service-environments \
    --service-environment my-sagemaker-service-env
```

L'ambiente di servizio passa allo `DELETING` stato durante il processo di eliminazione. Una volta completata l'eliminazione, l'ambiente di servizio non è più elencato nella descrizione delle operazioni. Le code di lavoro associate rimangono ma non possono elaborare i lavori di servizio finché non vengono associate a un ambiente di servizio diverso.

Job queues

I lavori vengono inviati a una coda di lavoro in cui risiedono fino a quando non è possibile programmarne l'esecuzione in un ambiente di elaborazione. Un AWS account può avere più code di lavoro. Ad esempio, puoi creare una coda che utilizza istanze Amazon EC2 On-Demand per lavori ad alta priorità e un'altra coda che utilizza istanze Amazon EC2 Spot per lavori a bassa priorità. Le code di lavoro hanno una priorità che viene utilizzata dallo scheduler per determinare quali lavori in quale coda devono essere valutati per primi per l'esecuzione.

Argomenti

Creare una coda di lavoro

Prima di poter inviare lavori AWS Batch, è necessario creare una coda di lavoro. Quando si crea una coda di lavoro, si associano uno o più ambienti di elaborazione alla coda e si assegna un ordine di preferenza.

È inoltre possibile impostare la priorità sulla coda dei lavori che determina l'ordine in cui lo scheduler AWS Batch colloca i lavori. Ciò significa che, se un ambiente di elaborazione è associato a più di una coda di lavoro, viene data la preferenza alla coda di lavoro con una priorità più alta.

Argomenti

Crea una coda di EC2 lavoro Amazon

Completa i seguenti passaggi per creare una coda di lavoro per Amazon Elastic Compute Cloud (Amazon EC2).

Per creare una coda di EC2 lavoro Amazon

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon). EC2
6. Per Nome, inserisci un nome univoco per la tua coda di lavoro. Il nome può contenere fino a 128 caratteri e contenere lettere maiuscole e minuscole, numeri e caratteri di sottolineatura (_).
7. Per Priorità, inserite un valore numerico intero per la priorità della coda di lavoro. Le code di lavoro con una priorità più alta vengono eseguite prima delle code di lavoro con priorità inferiore associate allo stesso ambiente di elaborazione. La priorità viene determinata in ordine decrescente, ad esempio a una coda di processi con valore di priorità 10 viene assegnata una preferenza di pianificazione rispetto a una coda di processi con valore di priorità 1.
8. (Facoltativo) Per la politica di pianificazione Amazon Resource Name (ARN), scegli una politica di pianificazione esistente.
9. Per gli ambienti di elaborazione connessi, seleziona uno o più ambienti di elaborazione dall'elenco da associare alla coda di lavoro. Seleziona gli ambienti di calcolo nell'ordine in cui desideri che la coda tenti di posizionarsi nella coda di lavoro. Il job scheduler utilizza l'ordine in cui vengono selezionati gli ambienti di calcolo per determinare in quale ambiente di calcolo viene avviato un determinato processo. Prima di poterli associare a una coda di lavoro, gli ambienti di calcolo devono trovarsi nello stato. `VALID` Puoi associare fino a tre ambienti di calcolo a una coda dei processi. Se non disponi di un ambiente di calcolo esistente, scegli Crea ambiente di calcolo

> **Note**
>
> Tutti gli ambienti di elaborazione associati a una coda di lavoro devono condividere lo stesso modello di provisioning. AWS Batch non supporta la combinazione di modelli di provisioning in un'unica coda di lavoro.

10. Per ordinare l'ambiente di calcolo, scegli le frecce su e giù per configurare l'ordine che desideri.
11. Scegli Crea coda lavori per terminare e crea la tua coda di lavoro.

Crea una coda di lavoro Fargate

Completa i seguenti passaggi per creare una coda di lavoro per. AWS Fargate

Per creare una coda di lavoro a Fargate

1. Apri la AWS Batch console all'indirizzo. https://console.aws.amazon.com/batch/
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Fargate.
6. In Nome, inserisci un nome univoco per la tua coda di lavoro. Il nome può contenere fino a 128 caratteri e contenere lettere maiuscole e minuscole, numeri e caratteri di sottolineatura (_).
7. Per Priorità, inserite un valore numerico intero per la priorità della coda di lavoro. Le code di lavoro con una priorità più alta vengono eseguite prima delle code di lavoro con priorità più bassa associate allo stesso ambiente di elaborazione. La priorità viene determinata in ordine decrescente, ad esempio a una coda di processi con valore di priorità 10 viene assegnata una preferenza di pianificazione rispetto a una coda di processi con valore di priorità 1.
8. (Facoltativo) Per la politica di pianificazione Amazon Resource Name (ARN), scegli una politica di pianificazione esistente.
9. Per gli ambienti di elaborazione connessi, seleziona uno o più ambienti di elaborazione dall'elenco da associare alla coda di lavoro. Seleziona gli ambienti di calcolo nell'ordine in cui desideri che la coda tenti di posizionarsi nella coda di lavoro. Il job scheduler utilizza l'ordine in cui vengono selezionati gli ambienti di calcolo per determinare in quale ambiente di calcolo viene avviato un determinato processo. Prima di poterli associare a una coda di lavoro, gli ambienti di calcolo devono trovarsi in tale stato. `VALID` Puoi associare fino a tre ambienti di calcolo a una coda dei processi.

> **Note**
>
> Tutti gli ambienti di elaborazione associati a una coda di lavoro devono condividere lo stesso modello di provisioning. AWS Batch non supporta la combinazione di modelli di provisioning in un'unica coda di lavoro.

10. Per ordinare l'ambiente di calcolo, scegli le frecce su e giù per configurare l'ordine che desideri.
11. Scegli Crea coda lavori per terminare e crea la tua coda di lavoro.

Crea una coda di lavoro Amazon EKS

Completa i seguenti passaggi per creare una coda di lavoro per Amazon Elastic Kubernetes Service (Amazon EKS).

Per creare una coda di lavoro Amazon EKS

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Amazon Elastic Kubernetes Service (Amazon EKS).
6. Per Nome, inserisci un nome univoco per la tua coda di lavoro. Il nome può contenere fino a 128 caratteri e contenere lettere maiuscole e minuscole, numeri e caratteri di sottolineatura (_).
7. Per Priority (Priorità) immetti un valore intero per la priorità della coda di processi. Le code di lavoro con una priorità più alta vengono eseguite prima delle code di lavoro con priorità più bassa associate allo stesso ambiente di elaborazione. La priorità viene determinata in ordine decrescente, ad esempio a una coda di processi con valore di priorità 10 viene assegnata una preferenza di pianificazione rispetto a una coda di processi con valore di priorità 1.
8. (Facoltativo) Per la politica di pianificazione Amazon Resource Name (ARN), scegli una politica di pianificazione esistente.
9. Per gli ambienti di elaborazione connessi, seleziona uno o più ambienti di elaborazione dall'elenco da associare alla coda di lavoro. Seleziona gli ambienti di calcolo nell'ordine in cui desideri che la coda tenti di posizionarsi nella coda di lavoro. Il job scheduler utilizza l'ordine in cui vengono selezionati gli ambienti di calcolo per determinare in quale ambiente di calcolo viene avviato un determinato processo. Prima di poterli associare a una coda di lavoro, gli ambienti

di calcolo devono trovarsi nello stato. `VALID` Puoi associare fino a tre ambienti di calcolo a una coda dei processi.

> **Note**
>
> Tutti gli ambienti di elaborazione associati a una coda di lavoro devono condividere lo stesso modello di provisioning. AWS Batch non supporta la combinazione di modelli di provisioning in un'unica coda di lavoro.

> **Note**
>
> Tutti gli ambienti di elaborazione associati a una coda di lavoro devono condividere la stessa architettura. AWS Batch non supporta la combinazione di tipi di architettura dell'ambiente di calcolo in un'unica coda di lavoro.

10. Per ordinare l'ambiente di calcolo, scegli le frecce su e giù per configurare l'ordine che desideri.
11. Scegli Crea coda lavori per terminare e crea la tua coda di lavoro.

Crea una coda SageMaker di lavoro di formazione in AWS Batch

SageMaker Le code di lavoro di formazione si integrano direttamente con il servizio di SageMaker intelligenza artificiale per fornire una pianificazione dei lavori senza server senza la necessità di gestire l'infrastruttura di elaborazione sottostante.

Prerequisiti

Prima di creare una coda per i lavori SageMaker di formazione, assicurati di avere:

- Ambiente di servizio: un ambiente di servizio che definisce i limiti di capacità. Per ulteriori informazioni, consulta Crea un ambiente di servizio in AWS Batch.
- Autorizzazioni IAM: autorizzazioni per creare e gestire code di AWS Batch lavoro e ambienti di servizio. Per ulteriori informazioni, consulta AWS Batch Politiche, ruoli e autorizzazioni IAM.

Create a SageMaker Training job queue (AWS Batch console)

1. Apri la console all' AWS Batch indirizzo. https://console.aws.amazon.com/batch/

2. Nel riquadro di navigazione, scegli Job queues e Create.
3. Per il tipo di orchestrazione, scegli Training. SageMaker
4. Per la configurazione della coda Job:

    a. Per Nome, immettere il nome della coda Job.
    b. per Priorità, immettete un valore compreso tra 0 e 1000. A una coda Job con una priorità più alta viene data la preferenza per gli ambienti di servizio.
    c. (Facoltativo) Per la politica di pianificazione Amazon Resource Name (ARN), scegli una politica di pianificazione esistente.
    d. Per gli ambienti di servizio connessi, seleziona un ambiente di servizio dall'elenco da associare alla coda di lavoro.
5. (Facoltativo) Per i limiti dello stato Job:

    a. Per Configurazione errata, scegliete `SERVICE_ENVIRONMENT_MAX_RESOURCE` e immettete il Tempo massimo di esecuzione (secondi).
    b. Per Capacità, scegli `INSUFFICIENT_INSTANCE_CAPACITY` e inserisci il Tempo di esecuzione massimo (secondi).
6. Scegli Crea coda di lavoro

Create a SageMaker Training job queue (AWS CLI)

Usa il `create-job-queue` comando per creare una coda di lavori SageMaker di formazione.

L'esempio seguente crea una coda di job SageMaker Training di base che utilizza un ambiente di servizio:

```
aws batch create-job-queue \
  --job-queue-name my-sm-training-fifo-jq \
  --job-queue-type SAGEMAKER_TRAINING \
  --priority 1 \
  --service-environment-order order=1,serviceEnvironment=ExampleServiceEnvironment
```

Sostituire *`ExampleServiceEnvironment`* con il nome dell'ambiente di servizio.

Il comando restituisce un output simile al seguente:

```
{
  "jobQueueName": "my-sm-training-fifo-jq",
```

```
  "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-jq"
}
```

Dopo aver creato la coda dei lavori, verifica che sia stata creata correttamente e che sia in uno stato valido.

Usa il `describe-job-queues` comando per visualizzare i dettagli sulla tua coda di lavoro:

```
aws batch describe-job-queues --job-queues my-sm-training-fifo-jq
```

Il comando restituisce un output simile al seguente:

```
{
  "jobQueues": [
    {
      "jobQueueName": "my-sm-training-fifo-jq",
      "jobQueueArn": "arn:aws:batch:region:account:job-queue/my-sm-training-fifo-
jq",
      "state": "ENABLED",
      "status": "VALID",
      "statusReason": "JobQueue Healthy",
      "priority": 1,
      "computeEnvironmentOrder": [],
      "serviceEnvironmentOrder": [
        {
          "order": 1,
          "serviceEnvironment": "arn:aws:batch:region:account:service-
environment/ExampleServiceEnvironment"
        }
      ],
      "jobQueueType": "SAGEMAKER_TRAINING",
      "tags": {},
      "jobStateTimeLimitActions": []
    }
  ]
}
```

Assicuratevi che:

- La `state` è `ENABLED`
- La `status` è `VALID`
- La `statusReason` è `JobQueue Healthy`

- La `jobQueueType` è `SAGEMAKER_TRAINING`
- I `serviceEnvironmentOrder` riferimenti al tuo ambiente di servizio

Modello Job queue

Di seguito è riportato un modello di coda di lavoro vuoto. È possibile utilizzare questo modello per creare la propria coda di lavoro. È quindi possibile salvare questa coda di lavoro in un file e utilizzarla con l' AWS CLI `--cli-input-json`opzione. Per ulteriori informazioni su questi parametri, consulta CreateJobQueuel'AWS Batch API Reference.

> **Note**
>
> È possibile generare un modello di coda di lavoro con il seguente AWS CLI comando.
>
> ```
> $ aws batch create-job-queue --generate-cli-skeleton
> ```

```
{
   "computeEnvironmentOrder": [
      {
         "computeEnvironment": "",
         "order": 0
      }
   ],
   "jobQueueName": "",
   "jobStateTimeLimitActions": [
      {
         "state": "RUNNABLE",
         "action": "CANCEL",
   "maxTimeSeconds": 0,
         "reason": ""

      }
   ],
   "priority": 0,
   "schedulingPolicyArn": "",
   "state": "ENABLED",
   "tags": {
      "KeyName": ""
   }
```

```
}
```

Visualizza lo stato della coda di lavoro

Dopo aver creato una coda di lavoro e aver inviato i lavori, è importante poterne monitorare l'avanzamento. Puoi utilizzare la pagina dei dettagli del lavoro per rivedere, gestire e monitorare la tua coda di lavoro.

Visualizza le informazioni sulla coda di lavoro

Dalla AWS Batch console, seleziona Job queues nel riquadro di navigazione e scegli la coda lavori desiderata per visualizzarne i dettagli. In questa pagina, puoi esaminare e gestire la tua coda di lavoro e visualizzare informazioni aggiuntive sulle operazioni della coda di lavoro, come l'istantanea della coda dei lavori, i limiti dello stato dei lavori, l'ordine dell'ambiente, i tag e il codice JSON della coda dei lavori.

Dettagli della coda di lavoro

Questa sezione fornisce una panoramica e le opzioni di manutenzione per la coda dei lavori. È importante notare che puoi trovare l'Amazon Resource Name (ARN) in questa sezione.

Per trovare queste informazioni tramite AWS Command Line Interface, utilizzare l'`DescribeJobQueues`operazione insieme al nome della coda di lavoro o all'ARN corrispondente.

Istantanea della coda dei lavori

Questa sezione fornisce un elenco statico dei primi 100 `RUNNABLE` lavori in coda. È possibile utilizzare il campo di ricerca per restringere l'elenco cercando informazioni da qualsiasi colonna della sezione dei risultati. I lavori nell'area dei risultati delle istantanee vengono ordinati in base alla strategia di esecuzione della coda dei lavori. Per le code di lavoro first-in-first-out (FIFO), l'ordine dei lavori si basa sull'orario di invio. Per una programmazione equa delle code di lavoro, l'ordine dei lavori si basa sulla priorità dei lavori e sulla condivisione dell'utilizzo.

Poiché i risultati sono un'istantanea della coda dei lavori, l'elenco dei risultati non viene aggiornato automaticamente. Per aggiornare l'elenco, scegli l'aggiornamento nella parte superiore della sezione. Scegli il collegamento ipertestuale del nome del lavoro per accedere ai dettagli del lavoro e visualizzare lo stato del lavoro e altre informazioni correlate.

Per trovare queste informazioni tramite AWS CLI, utilizzate l'`GetJobQueueSnapshot`operazione insieme al nome della coda di lavoro o all'ARN corrispondente.

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Limiti dello stato del lavoro

Utilizza questa scheda per esaminare le informazioni di configurazione relative al periodo di tempo in cui un lavoro può rimanere in uno `RUNNABLE` stato prima di essere annullato.

Per trovare queste informazioni tramite AWS CLI, utilizzate l'`DescribeJobQueues`operazione insieme al nome della coda di lavoro o all'ARN corrispondente.

Ordine ambientale

Se la coda dei lavori viene eseguita in più ambienti, questa scheda fornisce l'ordine e una panoramica.

Per trovare queste informazioni tramite AWS CLI, utilizzate l'`DescribeJobQueues`operazione insieme al nome della coda di lavoro o all'ARN corrispondente.

Tag

Utilizzate questa scheda per rivedere e gestire i tag associati a questa coda di lavori.

JSON

Utilizza questa scheda per copiare il codice JSON associato a questa coda di lavori. Puoi quindi riutilizzare il JSON per AWS CloudFormation modelli e script. AWS CLI

Monitora i lavori di assistenza

È possibile monitorare lo stato dei lavori di assistenza nella coda dei lavori utilizzando diversi AWS Batch comandi. I job di servizio sono processi eseguiti su AWS servizi come SageMaker Training, dove AWS Batch fornisce funzionalità di pianificazione e accodamento mentre il servizio di destinazione gestisce l'esecuzione dei lavori.

Elenca i lavori di assistenza per stato

Utilizza l'`ListServiceJobs`operazione per visualizzare i lavori di assistenza nella coda filtrati per stato. I lavori di assistenza possono avere i seguenti stati:

- `SUBMITTED`- Il lavoro è stato inviato ma non ancora elaborato

- `PENDING`- Job in sospeso e in attesa di risorse
- `RUNNABLE`- Job pronto per l'esecuzione e in attesa in coda
- `STARTING`- Job è in fase di avvio
- `RUNNING`- Job è attualmente in esecuzione
- `SCHEDULED`- Il lavoro è stato inviato al servizio di destinazione ma non è ancora in esecuzione
- `SUCCEEDED`- Job completato con successo
- `FAILED`- Job non completato

Visualizza i lavori in esecuzione nella tua coda:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status RUNNING
```

Visualizza i lavori in attesa in coda:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status RUNNABLE
```

Visualizza i lavori che sono stati inviati SageMaker ma non ancora in esecuzione:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status SCHEDULED
```

Visualizza tutti i lavori eseguiti con successo:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status SUCCEEDED
```

Visualizza i lavori non riusciti per la risoluzione dei problemi:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status FAILED
```

Filtra i lavori di assistenza

È possibile filtrare i lavori di assistenza per nome utilizzando il pattern matching. Se il valore di un filtro termina con un asterisco (*), corrisponde a qualsiasi nome di lavoro che inizia con la stringa che precede '*'.

Trovate offerte di lavoro con nome che inizia per «formazione»:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --filters name=JOB_NAME,values=training*
```

Trova lavori con nomi specifici:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --filters name=JOB_NAME,values=my-training-job-1,my-training-job-2
```

Combina filtri di stato e nome:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status RUNNING \
  --filters name=JOB_NAME,values=production*
```

Gestisci set di risultati di grandi dimensioni

Quando hai molti lavori di assistenza, utilizza l'impaginazione per gestire i risultati in modo efficace.

Limita il numero di risultati restituiti:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --max-results 10
```

Usa il token successivo per ottenere risultati aggiuntivi:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --max-results 10 \
  --next-token eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Ottieni informazioni dettagliate sulle mansioni di assistenza

Utilizzate l'DescribeServiceJoboperazione per ottenere informazioni complete su una specifica mansione di servizio, tra cui lo stato corrente, gli identificatori delle risorse di servizio e informazioni dettagliate sui tentativi.

Visualizza informazioni dettagliate su un lavoro specifico:

```
aws batch describe-service-job \
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d
```

Questo comando restituisce informazioni complete sul lavoro, tra cui:

- Job ARN e stato attuale
- Identificatori delle risorse del servizio (ad esempio SageMaker Training job ARN)
- Priorità di pianificazione e configurazione dei nuovi tentativi
- Payload della richiesta di servizio contenente i parametri di servizio originali
- Informazioni dettagliate sui tentativi con orari di inizio e fine
- Messaggi di stato dal servizio di destinazione

Monitora i lavori di SageMaker formazione

Quando si monitorano i lavori di SageMaker formazione tramite AWS Batch, è possibile accedere sia alle informazioni sulla AWS Batch mansione sia ai dettagli relativi ai lavori di SageMaker formazione sottostanti.

L'identificatore della risorsa di servizio nei dettagli del lavoro contiene l'ARN del lavoro di SageMaker formazione:

```
{
  "latestAttempt": {
    "serviceResourceId": {
      "name": "TrainingJobArn",
      "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/my-training-job"
    }
  }
}
```

Puoi utilizzare questo ARN per ottenere ulteriori dettagli direttamente da: SageMaker

```
aws sagemaker describe-training-job \
  --training-job-name my-training-job
```

Monitora l'avanzamento del lavoro controllando sia AWS Batch lo stato del lavoro che lo stato del lavoro di SageMaker formazione. Lo stato del AWS Batch lavoro mostra il ciclo di vita complessivo del lavoro, mentre lo stato del lavoro di SageMaker formazione fornisce dettagli specifici del servizio sul processo di formazione.

Interrompere i lavori di assistenza

Utilizzare l'`TerminateServiceJob`operazione per interrompere un processo di servizio in esecuzione.

Termina un lavoro di servizio specifico:

```
aws batch terminate-service-job \
  --job-id a4d6c728-8ee8-4c65-8e2a-9a5e8f4b7c3d \
  --reason "Job terminated by user request"
```

Quando si termina un lavoro di servizio, AWS Batch interrompe il lavoro e invia una notifica al servizio di destinazione. Per i lavori di SageMaker formazione, ciò interromperà anche il lavoro di formazione nell' SageMaker intelligenza artificiale.

Eliminare una coda di lavoro in AWS Batch

Quando non è più necessaria la coda dei lavori, è possibile disabilitarla ed eliminarla.

Delete a job queue (AWS Batch console)

1. Apri la AWS Batch console all'indirizzo. https://console.aws.amazon.com/batch/
2. Nel riquadro di navigazione, scegli Job queues, quindi scegli una job queue.
3. Scegli Azioni, quindi Disabilita.
4. Una volta che lo stato della coda dei lavori è disabilitato, scegli Azioni, quindi Elimina.
5. Nella finestra modale scegli Elimina coda lavori.

Delete a job queue (AWS CLI)

1. Disabilita la coda dei lavori per impedire l'invio di nuovi lavori:

```
aws batch update-job-queue \
  --job-queue my-sm-training-fifo-jq \
  --state DISABLED
```

2. Attendi il completamento di tutti i lavori in esecuzione, quindi elimina la coda dei lavori:

```
aws batch delete-job-queue \
  --job-queue my-sm-training-fifo-jq
```

Politiche di pianificazione con condivisione equa

Lo AWS Batch scheduler valuta quando, dove e come eseguire i lavori inviati a una coda di lavoro. Se non si specifica un criterio di pianificazione quando si crea una coda di lavoro, lo scheduler utilizza per impostazione predefinita una strategia FIFO (first-in, first-out). AWS Batch Una strategia FIFO potrebbe far sì che i lavori importanti rimangano «bloccati» rispetto ai lavori inviati in precedenza. Specificando una politica di pianificazione diversa, puoi allocare le risorse di elaborazione in base alle tue esigenze specifiche.

> **Note**
>
> Se desideri pianificare l'ordine specifico in cui vengono eseguiti i lavori, utilizza il `dependsOn` parametro in SubmitJobper specificare le dipendenze per ogni processo.

Se crei una politica di pianificazione e la alleghi a una coda di lavori, viene attivata la pianificazione con condivisione equa. Se la coda dei lavori ha una politica di pianificazione, la politica di pianificazione determina l'ordine in cui vengono eseguiti i lavori. Per ulteriori informazioni, consulta Utilizza politiche di pianificazione con condivisione equa per assegnare identificatori di condivisione.

Argomenti

- Utilizza gli identificatori di condivisione per identificare i carichi di lavoro
- Utilizza politiche di pianificazione con condivisione equa per assegnare identificatori di condivisione
- Usa la pianificazione con condivisione equa per aiutarti a pianificare i lavori
- Tutorial: Creare una politica di pianificazione
- Riferimento: modello di policy di pianificazione

Utilizza gli identificatori di condivisione per identificare i carichi di lavoro

Puoi utilizzare gli identificatori di condivisione per etichettare i lavori e distinguere tra utenti e carichi di lavoro. Lo AWS Batch scheduler tiene traccia dell'utilizzo per ogni identificatore di condivisione utilizzando la *(T * weightFactor)* formula, *T*dov'è l'utilizzo della vCPU nel tempo. Lo scheduler seleziona i lavori con l'utilizzo più basso dall'identificatore di condivisione. È possibile utilizzare un identificatore di condivisione senza sostituirlo.

> **Note**
>
> Gli identificatori di condivisione sono univoci all'interno di una coda di lavoro e non sono aggregati tra le code di lavoro.

È possibile impostare la priorità di pianificazione della condivisione equa per configurare l'ordine in cui i lavori vengono eseguiti su un identificatore condiviso. I lavori con una priorità di pianificazione più elevata vengono pianificati per primi. Se non si specifica una politica di pianificazione equa, tutti i lavori inviati alla coda dei lavori vengono pianificati in ordine FIFO. Quando invii un lavoro, non puoi specificare un identificatore di condivisione o una priorità di pianificazione equa.

> **Note**
>
> Le risorse di elaborazione allegate vengono allocate equamente tra tutti gli identificatori di condivisione, a meno che non vengano esplicitamente sovrascritte.

Utilizza politiche di pianificazione con condivisione equa per assegnare identificatori di condivisione

È possibile utilizzare le policy di pianificazione per configurare la modalità di allocazione delle risorse di calcolo in una coda di lavoro tra utenti o carichi di lavoro. Utilizzando politiche di pianificazione con condivisione equa, puoi assegnare identificatori di condivisione diversi a carichi di lavoro o utenti. AWS Batch assegna a ciascun identificatore di condivisione una percentuale delle risorse totali disponibili in un determinato periodo di tempo.

La percentuale di fair share viene calcolata utilizzando i valori and. `shareDecaySeconds` `shareDistribution` È possibile aggiungere tempo all'analisi del fair share assegnando un tempo di decadimento delle azioni alla politica. L'aggiunta di tempo dà più peso al tempo e meno

al peso definito. Puoi riservare le risorse di calcolo per gli identificatori di condivisione che non sono attivi specificando una prenotazione di elaborazione. Per ulteriori informazioni, consulta SchedulingPolicyDetail.

Usa la pianificazione con condivisione equa per aiutarti a pianificare i lavori

La pianificazione con condivisione equa fornisce una serie di controlli per aiutare a pianificare i lavori.

> **Note**
>
> Per ulteriori informazioni sui parametri delle politiche di pianificazione, vedere. SchedulingPolicyDetail

- Secondi di decadimento della condivisione: il periodo di tempo (in secondi) utilizzato dallo AWS Batch scheduler per calcolare una percentuale equa per ogni identificatore di condivisione. Un valore pari a zero indica che viene misurato solo l'utilizzo corrente. Un tempo di decadimento più lungo dà più peso al tempo.

  > **Note**
  >
  > Il periodo di decadimento è calcolato come segue: *shareDecaySeconds* `+` *OrderMinutes*`OrderMinutes` dov'è il tempo nell'ordine in minuti.

- Prenotazione di elaborazione: impedisce ai lavori in un unico identificatore di condivisione di utilizzare tutte le risorse allegate alla coda dei lavori. Il rapporto riservato indica *(computeReservation/100)^ActiveFairShares* dove si `ActiveFairShares` trova il numero di identificatori di condivisione attivi.

  > **Note**
  >
  > Se un identificatore di condivisione ha funzioni in uno `RUNNING` stato `SUBMITTEDPENDING`,, `RUNNABLESTARTING`, o, viene considerato un identificatore di condivisione attivo. Dopo la scadenza del periodo di decadimento, un identificatore di condivisione è considerato inattivo.

- Fattore di peso: il fattore di peso per l'identificatore azionario. Il valore predefinito è 1. Un valore più basso consente l'esecuzione dei job dall'identificatore di condivisione o fornisce un'autonomia aggiuntiva all'identificatore di condivisione. Ad esempio, ai lavori che utilizzano un identificatore di

condivisione con un fattore di peso di 0,125 (1/8) vengono assegnate otto volte le risorse di calcolo dei lavori che utilizzano un identificatore di condivisione con un fattore di peso pari a 1.

> **Note**
>
> È necessario definire questo attributo solo quando è necessario aggiornare il fattore di peso predefinito di 1.

Quando la coda dei lavori è attiva e sta elaborando i lavori, è possibile esaminare un elenco dei primi 100 `RUNNABLE` lavori tramite lo snapshot Job queue. Per ulteriori informazioni, consulta Visualizza lo stato della coda di lavoro.

Tutorial: Creare una politica di pianificazione

Prima di poter creare una coda di lavoro con una politica di pianificazione, è necessario creare una politica di pianificazione. Quando si crea una politica di pianificazione equa, si associano uno o più identificatori di condivisione o prefissi di identificatori condivisi ai pesi della coda e, facoltativamente, si assegna un periodo di decadimento e si calcola la prenotazione alla politica.

Per creare una politica di pianificazione

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel riquadro di navigazione, scegli Politiche di pianificazione, Crea.
4. In Nome, inserisci un nome univoco per la tua politica di pianificazione. Il nome può contenere un massimo di 128 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura.
5. (Facoltativo) Per Share decay seconds, inserite un valore intero per il tempo di decadimento della condivisione della policy di Fair-Share Scheduling. Un tempo di decadimento delle condivisioni più lungo comporterà un utilizzo notevolmente maggiore delle risorse di calcolo rispetto a un periodo di tempo più lungo durante la pianificazione dei lavori. Ciò può consentire ai job che utilizzano un identificatore di condivisione di utilizzare temporaneamente più risorse di elaborazione rispetto al peso che l'identificatore di condivisione consentirebbe se tale identificatore di condivisione non avesse utilizzato recentemente risorse di elaborazione.
6. (Facoltativo) Per Compute booking, inserisci un valore intero per la prenotazione di elaborazione della policy di pianificazione fair-share. La prenotazione di calcolo conterrà alcuni v come riserva da utilizzare per gli CPUs identificatori di condivisione che non sono attualmente attivi.

Il rapporto riservato indica `(computeReservation/100)^ActiveFairShares` dove si ActiveFairSharestrova il numero di identificatori azionari attivi.

Ad esempio, un `computeReservation` valore pari a 50 indica che è AWS Batch necessario riservare il 50% della VCPU massima disponibile se è presente un solo identificatore di condivisione, il 25% se sono presenti due identificatori di condivisione e il 12,5% se sono presenti tre identificatori di condivisione. Un `computeReservation` valore pari a 25 indica che è AWS Batch necessario riservare il 25% della VCPU massima disponibile se è presente un solo identificatore di condivisione, il 6,25% se sono presenti due identificatori di condivisione e l'1,56% se sono presenti tre identificatori di condivisione.

7. Nella sezione Attributi di condivisione, puoi specificare l'identificatore di condivisione e il peso per ogni identificatore di condivisione da associare alla politica di pianificazione della condivisione equa.

   a. Scegli Aggiungi identificatore di condivisione.

   b. Per Condividi identificatore, specifica l'identificatore di condivisione. Se la stringa termina con '*', diventa un prefisso identificativo di condivisione utilizzato per abbinare gli identificatori di condivisione per i lavori. Tutti gli identificatori di condivisione e i prefissi degli identificatori di condivisione in una politica di pianificazione devono essere unici e non possono sovrapporsi. Ad esempio, non è possibile inserire gli identificatori di condivisione come prefisso 'userA*' e l'identificatore di condivisione 'userA1' nella stessa politica di pianificazione a condivisione equa.

   c. Per il fattore di peso, specifica il peso relativo per l'identificatore della condivisione. Il valore predefinito è 1.0. Un valore inferiore ha una priorità più alta per le risorse di calcolo. Se viene utilizzato un prefisso identificativo di condivisione, i lavori con identificatori di condivisione che iniziano con il prefisso condivideranno il fattore di peso. Ciò aumenta efficacemente il fattore di peso per tali lavori, abbassandone la priorità individuale ma mantenendo lo stesso fattore di peso per il prefisso dell'identificatore di condivisione.

8. (Facoltativo) Nella sezione Tag, puoi specificare la chiave e il valore per ogni tag da associare alla politica di pianificazione. Per ulteriori informazioni, consulta Tagga le tue AWS Batch risorse.

9. Scegli Invia per completare e creare la tua politica di pianificazione.

Riferimento: modello di policy di pianificazione

Di seguito è riportato un modello di politica di pianificazione vuoto. È possibile utilizzare questo modello per creare la politica di pianificazione, che può quindi essere salvata in un file e utilizzata con l' AWS CLI `--cli-input-json`opzione. Per ulteriori informazioni su questi parametri, consulta CreateSchedulingPolicyl'AWS Batch API Reference.

> **Note**
>
> È possibile generare un modello di coda di lavoro con il seguente AWS CLI comando.
>
> ```
> $ aws batch create-scheduling-policy --generate-cli-skeleton
> ```

```
{
    "name": "",
    "fairsharePolicy": {
        "shareDecaySeconds": 0,
        "computeReservation": 0,
        "shareDistribution": [
            {
                "shareIdentifier": "",
                "weightFactor": 0.0
            }
        ]
    },
    "tags": {
        "KeyName": ""
    }
}
```

Pianificazione basata sulle risorse

AWS Batch pianifica i lavori in base alla vCPU, alla GPU e alla disponibilità di memoria nel Compute Environment (CE) associato a Job Queue (JQ). A volte, tuttavia, la semplice disponibilità di queste risorse CE non garantisce che il processo venga eseguito correttamente, poiché potrebbe dipendere da altre risorse necessarie, quindi tali lavori vengono annullati o terminati. Ciò si traduce in un uso inefficiente delle risorse di elaborazione. Per risolvere questo problema, la pianificazione basata sulle

risorse può verificare la disponibilità di risorse non CE dipendenti prima di pianificare l'esecuzione del lavoro su un CE.

AWS Batch La pianificazione basata sulle risorse consente di pianificare i lavori in base alle risorse consumabili necessarie per eseguire i lavori, ad esempio token di licenza di terze parti, larghezza di banda per l'accesso al database, necessità di limitare le chiamate a un'API di terze parti e così via. Si specificano le risorse consumabili necessarie per l'esecuzione di un processo e Batch tiene conto di queste dipendenze tra le risorse quando pianifica un lavoro. È possibile evitare interventi manuali per eliminare i fallimenti dei lavori e le lunghe attese causate dalla carenza di risorse consumabili. È possibile ridurre il sottoutilizzo delle risorse di elaborazione allocando solo i lavori che dispongono di tutte le risorse necessarie.

La pianificazione basata sulle risorse è disponibile per le politiche di pianificazione FIFO e Fair-share e può essere utilizzata con tutte le piattaforme di elaborazione supportate da Batch, tra cui EKS, ECS e Fargate. Può essere utilizzato con i job Array, i job Multi-node parallel (MNP) e con i normali job Batch.

Per configurare una pianificazione basata sulle risorse, è innanzitutto necessario specificare tutte le risorse consumabili necessarie per eseguire i processi, oltre al numero totale disponibile di ciascuna risorsa. Quindi, per ogni lavoro che richiede una risorsa consumabile, si specifica il nome e le quantità richieste di ogni risorsa necessaria. Batch tiene traccia di quante risorse consumabili sono disponibili per i lavori nelle code di lavoro e assicura che l'esecuzione di un processo sia pianificata solo quando tutte le risorse consumabili necessarie sono disponibili per il corretto funzionamento del lavoro.

Argomenti

Crea risorse consumabili

È innanzitutto necessario creare le risorse consumabili che rappresentano le risorse non CE che vengono consumate durante l'esecuzione di un lavoro e sono disponibili solo in quantità limitate. Ogni risorsa consumabile ha:

- nome della risorsa (`consumableResourceName`) che deve essere univoco a livello di account.
- (opzionale) tipo di risorsa (`resourceType`) che indica se la risorsa è disponibile per essere riutilizzata dopo il completamento di un lavoro. Può essere uno dei seguenti:
  - `REPLENISHABLE` (predefinito)
  - `NON_REPLENISHABLE`
- total quantity (`totalQuantity`) che specifica la quantità totale di risorse consumabili disponibili.

Il numero massimo di risorse consumabili per account è 50.000.

Console:

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.
2. Scegli Crea risorsa consumabile.
3. Inserisci un nome di risorsa univoco, la quantità totale della risorsa e seleziona se il tipo di risorsa è rifornibile o non rifornibile.
4. Scegli Crea risorsa consumabile.

API:

Usa l'`CreateConsumableResource`API per definire le risorse che desideri.

Specificate le risorse necessarie per eseguire un lavoro

Quando registri un lavoro, puoi specificare il nome di una o più risorse che hai creato (`consumableResource`) e la quantità di quella risorsa richiesta da ciascuna istanza del lavoro (`quantity`).

Batch tiene traccia delle unità disponibili di ogni risorsa in un dato momento. Per ogni processo nella coda dei lavori, lo scheduler Batch assicura che il processo venga eseguito solo quando sono disponibili le dipendenze delle risorse specificate.

Se una risorsa consumabile per il lavoro non è disponibile quando il lavoro raggiunge la testa della coda, il lavoro resterà in `RUNNABLE` stato attivo fino a quando tutte le risorse richieste non saranno disponibili o non verrà raggiunto il limite di tempo relativo allo stato del lavoro (vedi). Visualizza lo stato della coda di lavoro Una volta che Batch ha convalidato che tutte le risorse sono disponibili, il lavoro passa allo `STARTING` stato e poi allo stato. `RUNNING` Le risorse vengono bloccate una volta che il lavoro passa a `STARTING` o vengono quindi sbloccate quando il lavoro passa a o. `SUCCEEDED` `FAILED`

Puoi anche aggiornare la quantità di una risorsa necessaria per un lavoro specifico quando invii il lavoro.

Console:

Per specificare le risorse e le relative quantità necessarie quando si definisce un lavoro:

1. Definisci un lavoro utilizzando la procedura guidata di definizione del lavoro dalla AWS Batch console (Job definition -> Create).
2. Nella fase 4 della procedura guidata: Configurazione dei contenitori, in Risorsa consumabile, seleziona il nome di una risorsa richiesta dall'elenco. Nel campo Valore richiesto, inserisci la quantità di questa risorsa necessaria per un'istanza di questo lavoro, quindi scegli Aggiungi risorsa consumabile.
3. Ripeti il passaggio precedente per tutte le risorse consumabili richieste dal lavoro. È possibile specificare fino a 5 risorse per ogni lavoro definito.
4. Vedrai un elenco delle risorse consumabili che hai creato dopo aver completato la procedura guidata di definizione del lavoro ma prima di scegliere Crea definizione del lavoro.

Per aggiornare le quantità di risorse necessarie quando invii un lavoro:

1. Nel riquadro di navigazione a sinistra della AWS Batch console, scegli Lavori, quindi scegli Invia nuovo lavoro.
2. Nel Passaggio 2 della procedura guidata: Configurazione delle sostituzioni, in Sostituzioni di risorse consumabili, inserisci un nuovo valore richiesto per ogni risorsa consumabile di cui desideri sostituire la quantità necessaria per il lavoro.
3. Dopo aver completato tutte le sostituzioni che desideri apportare a questo lavoro, scegli Avanti per continuare con la revisione e l'invio.

API:

Quando registri un lavoro con l'`RegisterJobDefinitionAPI`, utilizza la `consumableResourceList` nella `consumableResourceProperties` parte della richiesta per specificare le risorse consumabili necessarie per eseguire un'istanza del processo e la quantità di ciascuna.

Quando invii un lavoro con l'`SubmitJobAPI`, puoi sovrascrivere l'elenco di risorse consumabili e la quantità di ciascuna utilizzando la `consumableResourcePropertiesOverride` parte della richiesta. Tieni presente che ciò sostituisce solo la quantità di risorsa necessaria per ogni istanza del lavoro, non la quantità totale disponibile.

Verifica quante risorse sono in uso e disponibili

Batch consente di interrogare il numero di risorse disponibili (`availableQuantity`), il numero di risorse in uso (`inUseQuantity`) e le risorse totali (`totalQuantity`) in un dato momento.

Una volta che un lavoro viene assegnato allo `STARTING` stato, le risorse consumate verranno sottratte dalla quantità disponibile di quella risorsa. Se la risorsa è`REPLENISHABLE`, il numero di risorse consumate verrà aggiunto nuovamente alla quantità disponibile non appena il lavoro passerà allo stato RIUSCITA o FALLITA e la quantità totale rimarrà la stessa. Se la risorsa è`NON_REPLENISHABLE`, il numero di risorse consumate viene sottratto sia dalla quantità totale che da quella disponibile e non verrà aggiunto nuovamente se il lavoro viene trasferito allo `SUCCEEDED` stato o. `FAILED`

> **Note**
>
> Queste informazioni possono subire ritardi fino a 30 secondi.

Console:

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.
2. Seleziona la scheda Rifornibile o Non rifornibile per visualizzare le risorse di quel tipo che hai creato.
3. Per ogni risorsa rifornibile, la console visualizza il nome, la quantità totale della risorsa, il numero attualmente in uso e quante sono ancora disponibili, insieme a un calcolo dell'utilizzo (il numero di risorse in uso diviso per la quantità totale di quella risorsa).

   Per ogni risorsa non rifornibile, la console visualizza il nome, il numero attualmente in uso e quante sono ancora disponibili.

È inoltre possibile visualizzare le informazioni correnti sulle risorse consumabili da una pagina dei dettagli del lavoro nella console.

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Lavori, quindi seleziona il nome di un lavoro per aprire la pagina dei dettagli del lavoro.
2. Le informazioni sulle risorse rifornibili e sulle risorse non rifornibili sono disponibili per visualizzare se il lavoro le richiede. Per entrambi i tipi, la console visualizza il nome della risorsa, la quantità richiesta per il lavoro, quante sono ancora disponibili, il numero attualmente in uso, la quantità totale della risorsa, insieme a un calcolo dell'utilizzo corrente (il numero di risorse in uso per il lavoro diviso per la quantità totale di quella risorsa).

API:

Utilizza l'`DescribeConsumableResource`API che restituisce le seguenti informazioni:

```
{
   "availableQuantity": number,
   "consumableResourceArn": "string",
   "consumableResourceName": "string",
   "createdAt": number,
   "inUseQuantity": number,
   "resourceType": "string",
   "tags": {
      "string" : "string"
   },
   "totalQuantity": number
}
```

L'`ListConsumableResources`API riporta anche il numero di risorse in uso (`inUseQuantity`) e il numero totale di risorse attualmente disponibili (`totalQuantity`) come parte dell'elenco di tutte le risorse consumabili che hai creato nel tuo account. Questa API consente inoltre di filtrare la query dell'elenco delle risorse consumabili in base al nome della risorsa consumabile.

Aggiorna la quantità di una risorsa mentre è utilizzata dai lavori

È possibile reimpostare la quantità totale di una risorsa su un nuovo valore, aggiungere alla quantità totale o sottrarla.

Se la nuova quantità totale specificata è maggiore rispetto a prima, Batch pianifica più lavori di conseguenza. Se la nuova quantità totale è inferiore a quella precedente e non ci sono unità di

questa risorsa in uso, Batch riduce semplicemente la quantità totale (o disponibile). Se ci sono unità in uso, Batch riduce immediatamente la quantità disponibile e, al termine dei lavori, Batch riduce la quantità totale (disponibile) in modo che alla fine arrivi al nuovo numero.

Console:

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.
2. Seleziona la scheda Rifornibile o Non rifornibile per visualizzare le risorse di quel tipo che hai creato.
3. Per le risorse rifornibili:

    1. Scegli la risorsa che desideri aggiornare, quindi seleziona Azioni e scegli Imposta risorse, Aggiungi risorse o Rimuovi risorse.
    2. Viene visualizzata una finestra pop-up in cui puoi impostare il valore totale, aggiungere risorse o rimuovere risorse a seconda dell'azione scelta nel passaggio precedente. Inserisci la quantità che desideri impostare come nuovo valore totale, che desideri aggiungere alla quantità totale o che desideri sottrarre dalla quantità totale, quindi seleziona Ok.

    Per le risorse non rifornibili:

    1. Scegli la risorsa che desideri aggiornare, quindi seleziona Azioni e scegli Imposta risorse, Aggiungi risorse o Rimuovi risorse.
    2. Viene visualizzata una finestra pop-up in cui puoi impostare il valore disponibile, aggiungere risorse o rimuovere risorse a seconda dell'azione scelta nel passaggio precedente. Inserisci la quantità che desideri impostare come nuovo valore disponibile, che desideri aggiungere alla quantità disponibile o che desideri sottrarre dalla quantità disponibile, quindi seleziona Ok.

API:

Utilizza l'`UpdateConsumableResource`API per impostare una nuova quantità totale per la risorsa o per aumentare o ridurre la quantità totale.

Trova i lavori che richiedono una risorsa consumabile specifica

Batch consente di recuperare un elenco di lavori che richiedono una risorsa consumabile specifica.

Console:

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.
2. Nell'elenco, seleziona il nome della risorsa consumabile. Viene visualizzata la pagina dei dettagli della risorsa.
3. In Cerca offerte di lavoro, inserisci i filtri che desideri applicare all'elenco delle offerte di lavoro. Puoi filtrare in base al nome del lavoro («uguale», «inizia con»), all'intervallo di date (quando il lavoro è stato creato) e in base a criteri aggiuntivi («coda lavoro», «definizione del lavoro», «identificatore del lavoro condiviso»). Per ogni tipo di filtro che desideri applicare, seleziona una delle opzioni disponibili nell'elenco a discesa e inserisci le informazioni aggiuntive richieste.

   Selezionare Search (Cerca).
4. Viene visualizzato un elenco (filtrato) di lavori che richiedono la risorsa consumabile, incluso il nome del lavoro, lo stato, il numero di unità richieste della risorsa consumabile, altre risorse consumabili necessarie e così via. Utilizzando questo elenco, è possibile selezionare uno o più lavori da annullare o terminare. Puoi anche selezionare il nome di un lavoro per aprire la pagina dei dettagli del lavoro.
5. In Cerca offerte di lavoro ora puoi aggiornare i risultati o cancellare la ricerca e ricominciare da capo.

API:

È possibile ottenere un elenco di lavori che utilizzano una risorsa consumabile specifica con l'`ListJobsByConsumableResource`API. Questa API consente inoltre di filtrare la query dell'elenco dei lavori utilizzando lo stato del lavoro o il nome del lavoro.

Eliminare una risorsa consumabile

È possibile eliminare una risorsa consumabile in qualsiasi momento, anche quando i lavori che richiedono la risorsa sono ancora in esecuzione. Una volta eliminata una risorsa consumabile, è possibile che si verifichi un intervallo tra il momento in cui il comando delete viene ricevuto e il job scheduler esegue l'eliminazione, pertanto è possibile che i lavori che consumano la risorsa vengano pianificati subito dopo la chiamata di eliminazione. Se la risorsa consumabile eliminata ha il tipo di risorsa (`resourceType`)`REPLENISHABLE`, questo verrà ignorato al termine dei lavori. Se si elimina una risorsa consumabile e la si ricrea con lo stesso nome, viene considerata la stessa risorsa e può essere utilizzata dai lavori. `RUNNABLE`

Console:

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.

2. Seleziona la scheda Rifornibile o Non rifornibile per visualizzare le risorse di quel tipo che hai creato.
3. Seleziona ogni risorsa che desideri eliminare, quindi scegli Elimina. Viene visualizzata una finestra pop-up Elimina risorsa consumabile. Per confermare l'eliminazione, scegliere Delete (Elimina).

Puoi anche eliminare una risorsa consumabile dalla relativa pagina dei dettagli nella console.

1. Nel pannello di navigazione a sinistra della AWS Batch console, scegli Risorse consumabili.
2. Seleziona la scheda Rifornibile o Non rifornibile per visualizzare le risorse di quel tipo che hai creato.
3. Scegli il nome della risorsa che desideri eliminare. Viene visualizzata la pagina dei dettagli della risorsa consumabile. Scegliere Delete (Elimina). Viene visualizzata una finestra pop-up Elimina risorsa consumabile. Per confermare l'eliminazione, scegliere Delete (Elimina).

API:

Utilizza l'`DeleteConsumableResource`API per eliminare una risorsa consumabile.

Definizioni del lavoro

AWS Batch le definizioni dei processi specificano come devono essere eseguiti i lavori. Sebbene ogni processo debba fare riferimento a una definizione di processo, molti dei parametri specificati nella definizione del processo possono essere sovrascritti in fase di esecuzione.

Alcuni degli attributi specificati in una definizione di processo includono:

- Quale immagine Docker utilizzare con il contenitore del lavoro.
- Quante v CPUs e quanta memoria usare con il contenitore.
- Il comando che il contenitore deve eseguire all'avvio.
- Quali variabili di ambiente (se ce ne sono) devono essere passate al contenitore all'avvio.
- Qualsiasi volume di dati da utilizzare con il contenitore.
- Quale ruolo IAM (se esiste) deve essere utilizzato dal tuo lavoro per AWS le autorizzazioni.

Indice

Creare una definizione di processo a nodo singolo

Prima di poter eseguire i lavori in AWS Batch, è necessario creare una definizione di processo. Questo processo varia leggermente tra processi paralleli a nodo singolo e multinodo. Questo argomento illustra in particolare come creare una definizione di processo per un AWS Batch processo che non è un processo parallelo a più nodi (noto anche come pianificazione di gruppo).

Puoi creare una definizione di processo parallelo multinodo sulle risorse di Amazon Elastic Container Service. Per ulteriori informazioni, consulta the section called "Creare una definizione di processo parallelo multinodo".

Argomenti

- Crea una definizione di lavoro a nodo singolo sulle risorse Amazon EC2
- Crea una definizione di lavoro a nodo singolo sulle risorse Fargate
- Crea una definizione di processo a nodo singolo sulle risorse Amazon EKS
- Crea una definizione di processo a nodo singolo con più contenitori sulle risorse Amazon EC2

Crea una definizione di lavoro a nodo singolo sulle risorse Amazon EC2

Completa i seguenti passaggi per creare una definizione di lavoro a nodo singolo sulle risorse Amazon Elastic Compute Cloud EC2 (Amazon).

Per creare una nuova definizione di lavoro sulle EC2 risorse di Amazon:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, scegli Regione AWS da usare.
3. Nel riquadro di navigazione a sinistra scegli Definizioni dei processi.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon). EC2
6. Per la configurazione della EC2 piattaforma, disattiva Enable multi-node parallel processing.
7. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
8. (Facoltativo) Per il timeout di esecuzione, immettete il valore di timeout (in secondi). Il timeout di esecuzione è il periodo di tempo che intercorre prima che un lavoro incompiuto venga terminato. Se un tentativo supera la durata del timeout, il tentativo viene interrotto e passa a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout. Il valore minimo è 60 secondi.
9. (Facoltativo) Attiva la priorità di pianificazione. Immettete un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore.
10. (Facoltativo) In Tentativi di lavoro, immettete il numero di volte in cui si AWS Batch tenta di spostare il lavoro allo `RUNNABLE` stato. Immettete un numero compreso tra 1 e 10.

11. (Facoltativo) Per le condizioni della strategia Retry, scegliete Aggiungi valutazione all'uscita. Inserisci almeno un valore di parametro, quindi scegli un'azione. Per ogni set di condizioni, l'azione deve essere impostata su Riprova o Esci. Queste azioni significano quanto segue:

    - Riprova: AWS Batch riprova fino al raggiungimento del numero di tentativi di lavoro specificato.
    - Esci: AWS Batch interrompe l'esecuzione di un nuovo tentativo.

    > **Important**
    >
    > Se scegli Aggiungi valutazione all'uscita, devi configurare almeno un parametro e scegliere un'azione o scegliere Rimuovi valutazione all'uscita.

12. (Facoltativo) Espandi Tag, quindi scegli Aggiungi tag per aggiungere tag alla risorsa. Inserisci una chiave e un valore facoltativo, quindi scegli Aggiungi tag.
13. (Facoltativo) Attiva i tag Propagate per propagare i tag dal processo e dalla definizione del processo al task Amazon ECS.
14. Scegli Pagina successiva.
15. Nella sezione Configurazione del contenitore:

    a. Per Image, scegli l'Dockerimmagine da usare per il tuo lavoro. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. Puoi anche specificare altri repository con *`repository-url`*`/`*`image`*`:`*`tag`*. Questo nome può contenere al massimo 225 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini (-), caratteri di sottolineatura (_), due 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.

    > **Note**
    >
    > L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

    - Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]`

denominazione (ad esempio,). `public.ecr.aws/registry_alias/my-web-app:latest`

- Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository[:tag]` denominazione completa (ad esempio,). `aws_account_id.dkr.ecr.region.amazonaws.com /my-web-app:latest`
- 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 repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

b. Per Command, inserisci i comandi nel campo come equivalente dell'array di stringhe JSON.

Questo parametro viene mappato su `Cmd` nella sezione Crea un container di Docker Remote API e il parametro `COMMAND` su docker run. Per ulteriori informazioni sul Docker`CMD` parametro, consulta https://docs.docker.com/engine/ reference/builder/ #cmd.

> **Note**
>
> È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

c. (Facoltativo) Per il ruolo Execution, specifica un ruolo IAM che conceda agli agenti del contenitore Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per le attività. Per ulteriori informazioni, consulta i ruoli IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

d. Per la configurazione di Job Role, scegli un ruolo IAM con autorizzazioni per. AWS APIs Questa funzionalità utilizza i ruoli IAM di Amazon ECS per le attività. Per ulteriori informazioni, consulta Ruoli IAM per le attività nella Guida per sviluppatori di Amazon Elastic Container Service.

> **Note**
>
> Qui vengono mostrati solo i ruoli con la relazione di trust Amazon Elastic Container Service Task Role. Per ulteriori informazioni sulla creazione di un ruolo IAM per i tuoi

AWS Batch lavori, consulta Creating an IAM Role and Policy for your Tasks nella Amazon Elastic Container Service Developer Guide.

16. Per Parametri, scegli Aggiungi parametri per aggiungere segnaposto di sostituzione dei parametri come coppie chiave e valori opzionali.
17. Nella sezione Configurazione dell'ambiente:
    a. Per v CPUs, inserisci il numero di v CPUs da riservare per il contenitore. Questo parametro è mappato a `CpuShares` nella sezione Create a container di Docker Remote API e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Devi specificare almeno un vCPU.
    b. Per Memoria, inserisci il limite di memoria disponibile per il contenitore. Se il contenitore tenta di superare la quantità di memoria specificata qui, il contenitore viene interrotto. Questo parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria.

    > **Note**
    >
    > Per massimizzare l'utilizzo delle risorse, dai la priorità alla memoria per i lavori di un tipo di istanza specifico. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

    c. Per Numero di GPUs, scegli il numero da GPUs riservare per il contenitore.
    d. (Facoltativo) Per le variabili di ambiente, scegliete Aggiungi variabile di ambiente per aggiungere variabili di ambiente come coppie nome-valore. Queste variabili vengono passate al contenitore.
    e. (Facoltativo) Per Segreti, scegliete Aggiungi segreto per aggiungere segreti come coppie nome-valore. Questi segreti sono esposti nel contenitore. Per ulteriori informazioni, LogConfiguration vedi:SecretOptions.
18. Scegli Pagina successiva.
19. Nella sezione Configurazione Linux:
    a. Per User (Utente) immetti il nome utente per l'utilizzo 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.

b. (Facoltativo) Per assegnare al job container autorizzazioni elevate sull'istanza host (simile a quella `root` dell'utente), trascina il cursore Privileged verso destra. Questo parametro è mappato a `Privileged` nella sezione Create a container di Docker Remote API e l'opzione `--privileged` a docker run.

c. (Facoltativo) Attiva Enable init process per eseguire un **`init`** processo all'interno del contenitore. Questo processo inoltra segnali e raccoglie i processi.

20. (Facoltativo) Nella sezione di configurazione del file system:

a. Attiva Abilita filesystem di sola lettura per rimuovere l'accesso in scrittura al volume.

b. In Dimensione della memoria condivisa, inserisci la dimensione (in MiB) del `/dev/shm` volume.

c. Per Dimensione massima di swap, inserisci la quantità totale di memoria di swap (in MiB) che il contenitore può utilizzare.

d. Per Swappiness, inserite un valore compreso tra 0 e 100 per indicare il comportamento di swappiness del contenitore. Se non specificate un valore e lo scambio è abilitato, il valore predefinito è 60. Per ulteriori informazioni, vedi:swappiness. LinuxParameters

e. (Facoltativo) Espandi Configurazione aggiuntiva.

f. (Facoltativo) Per Tmpfs, scegliete Aggiungi tmpfs per aggiungere un mount. `tmpfs`

g. (Facoltativo) Per i dispositivi, scegli Aggiungi dispositivo per aggiungere un dispositivo:

i. Per Container path (Percorso container), specifica il percorso dell'istanza del container per esporre il dispositivo mappato all'istanza host. Se lasci vuoto questo campo, il percorso dell'host viene utilizzato nel contenitore.

ii. Per Host path (Percorso host), specifica il percorso di un dispositivo nell'istanza host.

iii. Per Autorizzazioni, scegli una o più autorizzazioni da applicare al dispositivo. Le autorizzazioni disponibili sono READ, WRITE e MKNOD.

h. (Facoltativo) Per la configurazione dei volumi, scegliete Aggiungi volume per creare un elenco di volumi da passare al contenitore. Inserisci il nome e il percorso di origine per il volume, quindi scegli Aggiungi volume. Puoi anche scegliere di attivare Enable EFS.

i. (Facoltativo) Per i punti di montaggio, scegli Aggiungi configurazione dei punti di montaggio per aggiungere punti di montaggio per i volumi di dati. È necessario specificare il volume di origine e il percorso del contenitore. Questi punti di montaggio vengono passati Docker daemon a un'istanza del contenitore. Puoi anche scegliere di rendere il volume di sola lettura.

j. (Facoltativo) Per la configurazione Ulimits, scegli Aggiungi ulimit per aggiungere un `ulimits` valore per il contenitore. Inserisci i valori Name, Soft limit e Hard limit, quindi scegli Aggiungi ulimit.

21. Nella sezione Proprietà dell'attività:

a. Per Ruolo di esecuzione: condizionale, scegli un ruolo per consentire agli agenti Amazon ECS di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni sulla creazione di un ruolo di esecuzione, consulta. Tutorial: Creare il ruolo di esecuzione IAM

b. Scegli il comando Enable ECS execute per abilitare l'accesso diretto alla shell del contenitore Amazon ECS e bypassare il sistema operativo host. Devi scegliere un ruolo Task.

> **Important**
>
> Il comando ECS execute richiede che il file system sia scrivibile.

c. Per il ruolo Task, scegli un ruolo Amazon ECS Identity and Access Management (IAM) per consentire al contenitore di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni, consulta il ruolo IAM delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

22. (Facoltativo) Nella sezione Configurazione della registrazione:

a. Per Log driver, scegli il driver di registro da usare. Per ulteriori informazioni sui driver di registro disponibili, LogConfiguration vedi:LogDriver.

> **Note**
>
> Per impostazione predefinita, viene utilizzato il driver di `awslogs` registro.

b. Per Opzioni, scegliete Aggiungi opzione per aggiungere un'opzione. Immettete una coppia nome-valore, quindi scegliete l'opzione Aggiungi.

c. Per Segreti, scegli Aggiungi segreto. Inserisci una coppia nome-valore, quindi scegli Aggiungi segreto per aggiungere un segreto.

> **Tip**
>
> Per ulteriori informazioni, vedi:SecretOptions. LogConfiguration

23. Scegli Pagina successiva.
24. Per la revisione della definizione di Job, rivedi i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea definizione del lavoro.

Crea una definizione di lavoro a nodo singolo sulle risorse Fargate

Completa i seguenti passaggi per creare una definizione di processo a nodo singolo sulle AWS Fargate risorse.

Per creare una nuova definizione di lavoro sulle risorse di Fargate:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione in alto, scegli Regione AWS da usare.
3. Nel riquadro di navigazione a sinistra scegli Definizioni dei processi.
4. Scegli Create (Crea).
5. Per Tipo di orchestrazione, scegli Fargate. Per ulteriori informazioni, consulta Ambienti di elaborazione Fargate.
6. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
7. (Facoltativo) Per il timeout di esecuzione, immettete il valore di timeout (in secondi). Il timeout di esecuzione è il periodo di tempo che intercorre prima che un lavoro incompiuto venga terminato. Se un tentativo supera la durata del timeout, il tentativo viene interrotto e passa a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout. Il valore minimo è 60 secondi.
8. (Facoltativo) Attiva la priorità di pianificazione. Immettete un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore rispetto ai valori inferiori.
9. (Facoltativo) Espandi Tag, quindi scegli Aggiungi tag per aggiungere tag alla risorsa. Attiva i tag Propagate per propagare i tag dal processo e dalla definizione del lavoro.
10. Nella sezione di configurazione della piattaforma Fargate:
    a. Per la piattaforma Runtime, scegli l'architettura dell'ambiente di calcolo.
    b. Per Operating System Family, scegli il sistema operativo per l'ambiente di calcolo.
    c. Per Architettura CPU, scegli l'architettura vCPU.
    d. Per la versione della piattaforma Fargate, inserire **`LATEST`** o una versione specifica dell'ambiente di runtime.

e. (Facoltativo) Attivate Assegna IP pubblico per assegnare un indirizzo IP pubblico a un'interfaccia di rete Fargate Job. Per un processo in esecuzione in una sottorete privata per inviare traffico in uscita a Internet, la sottorete privata richiede il collegamento di un gateway NAT per instradare le richieste verso Internet. Potresti volerlo fare in modo da poter estrarre le immagini dei contenitori. Per ulteriori informazioni, consulta Networking di attività Amazon ECS nella Guida per lo sviluppatore di Amazon Elastic Container Service.

f. (Facoltativo) Per Archiviazione temporanea, immettete la quantità di spazio di archiviazione effimero da allocare all'attività. La quantità di storage temporaneo deve essere compresa tra 21 GiB e 200 GiB. Per impostazione predefinita, vengono allocati 20 GiB di storage temporaneo se non si immette un valore.

> **Note**
>
> Lo storage temporaneo richiede la versione 1.4 o successiva della piattaforma Fargate.

g. Per il ruolo Execution, specifica un ruolo IAM che conceda al container Amazon ECS e agli agenti Fargate l'autorizzazione a effettuare chiamate AWS API per tuo conto. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, compresi i prerequisiti di configurazione, consulta i ruoli IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

h. Per Job tentations, inserisci il numero di volte in cui AWS Batch tenta di portare il job a uno `RUNNABLE` stato. Immettete un numero compreso tra 1 e 10.

i. Facoltativo) Per le condizioni della strategia Retry, scegli Aggiungi valuta all'uscita. Inserisci almeno un valore di parametro, quindi scegli un'azione. Per ogni set di condizioni, l'azione deve essere impostata su Riprova o Esci. Queste azioni significano quanto segue:

- Riprova: AWS Batch riprova fino al raggiungimento del numero di tentativi di lavoro specificato.
- Esci: AWS Batch interrompe l'esecuzione di un nuovo tentativo.

> **Important**
>
> Se scegli Aggiungi valutazione all'uscita, devi configurare almeno un parametro e scegliere un'azione oppure scegliere Rimuovi valutazione all'uscita.

11. Scegli Pagina successiva.
12. Nella sezione Configurazione del contenitore:

    a. Per Immagine scegli l'immagine Docker da utilizzare per il processo. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. Puoi anche specificare altri repository con *repository-url*/*image*:*tag*. Questo nome può contenere al massimo 225 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini bassi (-), caratteri di sottolineatura (_), due punti (:), punti (.), barre (/) e simboli di numero (#). Questo parametro è mappato a `Image` nella sezione Crea un container dell'API remota Docker e al parametro `IMAGE` di docker run.

    > **Note**
    >
    > L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

    - Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]` denominazione (ad esempio,). `public.ecr.aws/`*registry_alias*`/`*my-web-app*`:`*latest*
    - Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository[:tag]` denominazione completa (ad esempio,). *aws_account_id*`.dkr.ecr.`*region*`.amazonaws.com /`*my-web-app*`:`*latest*
    - 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 repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

    b. Per Command, inserisci i comandi nel campo come equivalenti all'array di stringhe JSON.

    Questo parametro viene mappato su `Cmd` nella sezione Crea un container di Docker Remote API e il parametro `COMMAND` su docker run. Per ulteriori informazioni sul Docker `CMD` parametro, vedere https://docs.docker.com/engine/reference/builder/ #cmd.

> **Note**
>
> È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

c. (Facoltativo) Aggiungi parametri alla definizione del processo come mappature nome-valore per sovrascrivere i valori predefiniti della definizione del processo. Per aggiungere un parametro:

- Per Parametri, scegliete Aggiungi parametri, inserite una coppia nome-valore, quindi scegliete Aggiungi parametro.

> **Important**
>
> Se scegli Aggiungi parametro, devi configurare almeno un parametro o scegliere Rimuovi parametro

d. Nella sezione Configurazione dell'ambiente:

i. Per la configurazione del ruolo Job, scegli un ruolo IAM con autorizzazioni per. AWS APIs Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, consulta Ruoli IAM per le attività nella Guida per sviluppatori di Amazon Elastic Container Service.

> **Note**
>
> Qui vengono mostrati solo i ruoli con la relazione di trust Amazon Elastic Container Service Task Role. Per ulteriori informazioni su come creare un ruolo IAM per i tuoi AWS Batch lavori, consulta Creating an IAM Role and Policy for your Tasks nella Amazon Elastic Container Service Developer Guide.

ii. Per v CPUs, inserisci il numero di v CPUs da riservare per il contenitore. Questo parametro è mappato a `CpuShares` nella sezione Create a container di Docker Remote API e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Devi specificare almeno un vCPU.

iii. Per Memoria, inserisci il limite di memoria disponibile per il contenitore. Se il contenitore tenta di superare la memoria specificata qui, il contenitore viene fermato. Questo

parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria.

Se si utilizza GuardDuty Runtime Monitoring, si verifica un leggero sovraccarico di memoria per il GuardDuty security agent. Pertanto, il limite di memoria deve includere la dimensione del GuardDuty security agent. Per informazioni sui limiti di memoria del GuardDuty Security Agent, vedere Limiti di CPU e memoria nella Guida per l'GuardDuty utente. Per informazioni sulle best practice, consulta Come posso rimediare agli errori di memoria esaurita nelle mie attività di Fargate dopo aver abilitato il monitoraggio del runtime nella Amazon ECS Developer Guide.

> **Note**
>
> Per massimizzare l'utilizzo delle risorse, dai la priorità alla memoria per i lavori di un tipo di istanza specifico. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

e. (Facoltativo) Per le variabili di ambiente, scegliete Aggiungi variabile di ambiente per aggiungere variabili di ambiente come coppie nome-valore. Queste variabili vengono passate al contenitore.

f. (Facoltativo) Per Segreti, scegliete Aggiungi segreto per aggiungere segreti come coppie nome-valore. Questi segreti sono esposti nel contenitore. Per ulteriori informazioni, LogConfiguration vedi:SecretOptions.

g. Scegli Pagina successiva.

13. (Facoltativo) Nella sezione di configurazione Linux:

a. Per Utente, inserisci un nome utente da utilizzare all'interno del contenitore.

b. Attiva Abilita processo init per eseguire un processo init all'interno del container. Questo processo inoltra segnali e raccoglie processi.

c. Attiva Abilita filesystem di sola lettura per rimuovere l'accesso in scrittura al volume.

d. (Facoltativo) Espandi Configurazione aggiuntiva.

e. Per la configurazione dei punti di montaggio, scegli Aggiungi configurazione dei punti di montaggio per aggiungere punti di montaggio per i volumi di dati. È necessario specificare il volume di origine e il percorso del contenitore. Questi punti di montaggio vengono passati Docker daemon a un'istanza del contenitore.

f. Per la configurazione dei volumi, scegli Aggiungi volume per creare un elenco di volumi da passare al contenitore. Inserisci un nome e un percorso di origine per il volume, quindi scegli Aggiungi volume.

g. Nella sezione Proprietà dell'attività:

   i. Per Ruolo di esecuzione: condizionale, scegli un ruolo per consentire agli agenti Amazon ECS di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni sulla creazione di un ruolo di esecuzione, consulta. Tutorial: Creare il ruolo di esecuzione IAM

   ii. Scegli il comando Enable ECS execute per abilitare l'accesso diretto alla shell del contenitore Amazon ECS e bypassare il sistema operativo host. Devi scegliere un ruolo Task.

   > **Important**
   >
   > Il comando ECS execute richiede che il file system sia scrivibile.

   iii. Per il ruolo Task, scegli un ruolo Amazon ECS Identity and Access Management (IAM) per consentire al contenitore di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni, consulta il ruolo IAM delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

h. Nella sezione Configurazione della registrazione:

   i. (Facoltativo) Per il driver di registro, scegliete il driver di registro da utilizzare. Per ulteriori informazioni sui driver di registro disponibili, LogConfiguration vedi:LogDriver.

   > **Note**
   >
   > Per impostazione predefinita, viene utilizzato il driver di `awslogs` registro.

   ii. (Facoltativo) Per Opzioni, scegliete Aggiungi opzione per aggiungere un'opzione. Immettete una coppia nome-valore, quindi scegliete l'opzione Aggiungi.

   iii. (Facoltativo) Per Segreti, scegliete Aggiungi segreto per aggiungere un segreto. Quindi, inserisci una coppia nome-valore e scegli Aggiungi segreto.

> **Tip**
>
> Per ulteriori informazioni, vedi:SecretOptions. LogConfiguration

14. Scegli Pagina successiva.
15. Per la revisione della definizione di Job, rivedi i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea definizione del lavoro.

Crea una definizione di processo a nodo singolo sulle risorse Amazon EKS

Completa i seguenti passaggi per creare una definizione di processo a nodo singolo su Amazon Elastic Kubernetes Service (Amazon EKS).

Per creare una nuova definizione di lavoro sulle risorse di Amazon EKS:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione in alto, scegli Regione AWS da usare.
3. Nel riquadro di navigazione a sinistra scegli Definizioni dei processi.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Elastic Kubernetes Service (EKS).
6. Per Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
7. (Facoltativo) Per il timeout di esecuzione, immettete il valore di timeout (in secondi). Il timeout di esecuzione è il periodo di tempo che intercorre prima che un lavoro incompiuto venga terminato. Se un tentativo supera la durata del timeout, il tentativo viene interrotto e passa a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout. Il valore minimo è 60 secondi.
8. (Facoltativo) Attiva la priorità di pianificazione. Immettete un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore rispetto ai valori inferiori.
9. (Facoltativo) Espandi Tag, quindi scegli Aggiungi tag per aggiungere tag alla risorsa.
10. Scegli Pagina successiva.
11. Nella sezione delle podproprietà EKS:

    a. Per il nome dell'account di servizio, inserisci un account che fornisca un'identità per i processi eseguiti in unpod.

b. Attiva la rete Host per utilizzare il modello Kubernetes pod di rete e apri una porta di ascolto per le connessioni in entrata. Disattiva questa impostazione solo per le comunicazioni in uscita.

c. Per la politica DNS, scegli una delle seguenti opzioni:

- Nessun valore (null): pod ignora le impostazioni DNS dall'ambiente. Kubernetes
- Predefinita: pod eredita la configurazione della risoluzione dei nomi dal nodo su cui viene eseguito.

> **Note**
>
> Se non viene specificata una policy DNS, Predefinita non è la policy DNS predefinita. Viene invece utilizzato. ClusterFirst

- ClusterFirst— Qualsiasi query DNS che non corrisponde al suffisso del dominio del cluster configurato viene inoltrata al nameserver upstream ereditato dal nodo.
- ClusterFirstWithHostNet— Da utilizzare se la rete host è attiva.

d. (Facoltativo) Per i volumi, seleziona Aggiungi volume, quindi:

i. Aggiungi un nome per il tuo volume.

ii. (Facoltativo) Aggiungi il percorso dell'host per la directory sull'host.

iii. (Facoltativo) Aggiungi un limite medio e uno di dimensione per configurare Kubernetes EmptyDir.

iv. (Facoltativo) Fornisci un nome segreto per il pod e indica se il segreto è Facoltativo.

v. (Facoltativo) Definisci il nome di un claim per allegare un claim Kubernetes Persistent Volume al pod e indica se è di sola lettura.

e. (Facoltativo) Per le etichette Pod, scegli Aggiungi etichette per pod, quindi inserisci una coppia nome-valore.

> **Important**
>
> Il prefisso per l'etichetta di un pod non può contenere`kubernetes.io/`, `k8s.io/` o. `batch.amazonaws.com/`

f. (Facoltativo) Per le annotazioni Pod, scegli Aggiungi annotazioni, quindi inserisci una coppia nome-valore.

> **Important**
>
> Il prefisso per un'annotazione del pod non può contenere, o. `kubernetes.io/` `k8s.io/` `batch.amazonaws.com/`

g. Scegli Pagina successiva.

h. Nella sezione Configurazione del contenitore:

i. Per Nome, inserisci un nome univoco per il contenitore. Il nome deve iniziare con una lettera o un numero e può contenere fino a 63 caratteri. Può contenere lettere maiuscole e minuscole, numeri e trattini (-).

ii. Per Image, scegli l'immagine da usare per il tuo Docker lavoro. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. Puoi anche specificare altri repository con *`repository-url`*`/`*`image`*`:`*`tag`*. Il nome può contenere fino a 255 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini bassi (-), caratteri di sottolineatura (_), due punti (:), punti (.), barre (/) e simboli di numero (#). Questo parametro è `Image` mappato alla sezione Crea un contenitore dell'API Docker Remote e al `IMAGE` parametro di docker run

> **Note**
>
> L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

- Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]` denominazione (ad esempio,). `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*
- Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository[:tag]` denominazione completa (ad esempio,). *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com /`*`my-web-app`*`:`*`latest`*
- 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 repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

iii. (Facoltativo) Per la policy Image pull, scegli quando estrarre le immagini.

iv. (Facoltativo) In Comando, inserite un JSON comando da passare al contenitore.

v. (Facoltativo) In Argomenti, inserite gli argomenti da passare al contenitore. Se non viene fornito un argomento, viene utilizzato il comando container image.

i. (Facoltativo) È possibile aggiungere parametri alla definizione del processo come mappature nome-valore per sovrascrivere i valori predefiniti della definizione del processo. Per aggiungere un parametro:

- Per Parametri, inserisci una coppia nome-valore, quindi scegli Aggiungi parametro.

> **Important**
>
> Se scegli Aggiungi parametro, devi configurare almeno un parametro o scegliere Rimuovi parametro

j. Nella sezione Configurazione dell'ambiente:

i. Per v CPUs, inserisci il numero di v CPUs da riservare per il contenitore. Questo parametro è mappato a `CpuShares` nella sezione Create a container di Docker Remote API e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Devi specificare almeno un vCPU.

ii. Per Memoria, inserisci il limite di memoria disponibile per il contenitore. Se il contenitore tenta di superare la memoria specificata qui, il contenitore viene interrotto. Questo parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria.

> **Note**
>
> Per massimizzare l'utilizzo delle risorse, dai la priorità alla memoria per i lavori di un tipo di istanza specifico. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

k. (Facoltativo) Per le variabili di ambiente, scegliete Aggiungi variabile di ambiente per aggiungere variabili di ambiente come coppie nome-valore. Queste variabili vengono passate al contenitore.

l. (Facoltativo) Per Volume Mount:

   i. Scegli Aggiungi montaggio per volume.

   ii. Inserisci un nome, quindi inserisci un percorso di montaggio nel contenitore in cui è montato il volume. Immettete SubPatha per specificare un sottopercorso all'interno del volume di riferimento anziché la relativa radice.

   iii. Scegliete Sola lettura per rimuovere le autorizzazioni di scrittura sul volume.

   iv. Scegli Aggiungi supporto per volume.

m. (Facoltativo) Per Esegui come utente, inserisci un ID utente per eseguire il processo del contenitore.

> **Note**
>
> L'ID utente deve esistere nell'immagine per consentire l'esecuzione del contenitore.

n. (Facoltativo) Per Esegui come gruppo, inserisci un ID di gruppo per eseguire il runtime del processo del contenitore.

> **Note**
>
> L'ID del gruppo deve esistere nell'immagine per consentire l'esecuzione del contenitore.

o. (Facoltativo) Per assegnare al contenitore del lavoro autorizzazioni elevate sull'istanza host (analogamente a quelle `root` dell'utente), trascina il cursore Privileged verso destra. Questo parametro è mappato a `Privileged` nella sezione Create a container di Docker Remote API e l'opzione `--privileged` a docker run.

p. (Facoltativo) Attiva il filesystem root di sola lettura per rimuovere l'accesso in scrittura al filesystem root.

q. (Facoltativo) Attivate Esegui come utente non root per eseguire i contenitori in modalità utente non root. pod

> **Note**
>
> Se l'opzione Esegui come non-root è attivata, kubelet convalida l'immagine in fase di esecuzione per verificare che non venga eseguita come UID 0.

r. Scegli Pagina successiva.

12. Per la revisione della definizione di Job, rivedi i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea definizione del lavoro.

Crea una definizione di processo a nodo singolo con più contenitori sulle risorse Amazon EC2

Completa i seguenti passaggi per creare una definizione di lavoro a nodo singolo con più contenitori su risorse Amazon Elastic Compute Cloud EC2 (Amazon).

Per creare una nuova definizione di lavoro sulle EC2 risorse di Amazon:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, scegli Regione AWS da usare.
3. Nel riquadro di navigazione a sinistra scegli Definizioni dei processi.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon). EC2
6. Per la struttura delle definizioni di Job, disattiva l'elaborazione della struttura Use legacy ContainerProperties.
7. Per la configurazione della EC2 piattaforma, disattiva Enable multi-node parallel processing.
8. Scegli Next (Successivo).
9. Nella sezione Configurazione generale, inserisci quanto segue:

    a. Per Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

    b. Per Timeout di esecuzione, facoltativo, inserisci il valore di timeout (in secondi). Il timeout di esecuzione è il periodo di tempo che intercorre prima che un lavoro incompiuto venga terminato. Se un tentativo supera la durata del timeout, il tentativo viene interrotto e passa

a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout. Il valore minimo è 60 secondi.

c. Attiva la priorità di pianificazione (facoltativa). Inserisci un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore.

d. Espandi tag: facoltativo, quindi scegli Aggiungi tag per aggiungere tag alla risorsa. Inserisci una chiave e un valore facoltativo, quindi scegli Aggiungi tag.

e. Attiva i tag Propagate per propagare i tag dal processo e dalla definizione del processo al task Amazon ECS.

10. Nella sezione Retry strategy - opzionale, inserisci quanto segue:

a. Per Job tentations, inserisci il numero di volte in cui AWS Batch tenta di portare il job allo `RUNNABLE` status. Immettete un numero compreso tra 1 e 10.

b. Per le condizioni della strategia Retry, scegli Aggiungi valuta all'uscita. Inserisci almeno un valore di parametro, quindi scegli un'azione. Per ogni set di condizioni, l'azione deve essere impostata su Riprova o Esci. Queste azioni significano quanto segue:

- Riprova: AWS Batch riprova fino al raggiungimento del numero di tentativi di lavoro specificato.
- Esci: AWS Batch interrompe l'esecuzione di un nuovo tentativo.

> **Important**
>
> Se scegli Aggiungi valutazione all'uscita, devi configurare almeno un parametro e scegliere un'azione o scegliere Rimuovi valutazione all'uscita.

11. Nella sezione Proprietà dell'attività, inserisci quanto segue:

a. Per Ruolo di esecuzione: condizionale, scegli un ruolo per consentire agli agenti Amazon ECS di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni sulla creazione di un ruolo di esecuzione, consulta. Tutorial: Creare il ruolo di esecuzione IAM

b. Scegli il comando Enable ECS execute per abilitare l'accesso diretto alla shell del contenitore Amazon ECS e bypassare il sistema operativo host. Devi scegliere un ruolo Task.

> **Important**
>
> Il comando ECS execute richiede che il file system sia scrivibile.

c. Per il ruolo Task, scegli un ruolo Amazon ECS Identity and Access Management (IAM) per consentire al contenitore di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni, consulta il ruolo IAM delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

d. Per la modalità IPC scegli `hosttask`, o. `none` Se `host` specificato, tutti i contenitori che rientrano nelle attività che hanno specificato la modalità IPC host sulla stessa istanza di contenitore condividono le stesse risorse IPC con l'istanza Amazon EC2 host. Se viene specificata un'attività, tutti i contenitori che si trovano all'interno dell'attività specificata condividono le stesse risorse IPC. Se non ne viene specificato nessuno, le risorse IPC all'interno dei contenitori di un'attività sono private e non condivise con altri contenitori in un'attività o nell'istanza del contenitore. 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.

e. Per la modalità PID scegli `host otask`. Ad esempio, i sidecar di monitoraggio potrebbero aver bisogno di `pidMode` per accedere a informazioni su altri container in esecuzione nella stessa attività. Se `host` specificato, tutti i contenitori all'interno delle attività che hanno specificato la modalità PID dell'host sulla stessa istanza di contenitore condividono lo stesso spazio dei nomi di processo con l'istanza Amazon EC2 host. Se è stato specificato `task`, tutti i container all'interno dell'attività specificata condividono lo stesso spazio dei nomi del processo. Se non è stato specificato alcun valore, l'impostazione predefinita è uno spazio dei nomi privato per ogni container.

12. Nella sezione Risorse consumabili, inserisci quanto segue:

    a. Inserisci un nome univoco e il valore richiesto.

    b. Puoi aggiungere altre risorse consumabili selezionando Aggiungi risorsa consumabile.

13. Nella sezione Archiviazione, inserisci quanto segue:

    a. Inserisci un nome e un percorso di origine per il volume, quindi scegli Aggiungi volume. Puoi anche scegliere di attivare Enable EFS.

    b. Puoi aggiungere altri volumi scegliendo Aggiungi volume.

14. Per Parametri, scegliete Aggiungi parametri per aggiungere segnaposto sostitutivi dei parametri come coppie Chiave e Valore opzionali.
15. Scegliete Pagina successiva.
16. Nella sezione Configurazione del contenitore:

    a. In Name (Nome) immetti un nome per il container.

    b. Per Essential container, abilita se il contenitore è essenziale.

    c. Per Image, scegli l'Dockerimmagine da usare per il tuo lavoro. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. Puoi anche specificare altri repository con *repository-url*/*image*:*tag*. Questo nome può contenere al massimo 225 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini (-), caratteri di sottolineatura (_), due 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.

    > **Note**
    >
    > L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

    - Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]` denominazione (ad esempio,). `public.ecr.aws/`*registry_alias*`/`*my-web-app*`:`*latest*
    - Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository[:tag]` denominazione completa (ad esempio,). *aws_account_id*`.dkr.ecr.`*region*`.amazonaws.com /`*my-web-app*`:`*latest*
    - 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 repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

    d. Per i requisiti relativi alle risorse, configura ciascuno dei seguenti elementi:

i. Per v CPUs, scegli il numero di CPUs per il contenitore.

ii. Per Memoria, scegli la quantità di memoria per il contenitore.

iii. Per GPU: facoltativo, scegli il numero di GPUs per il contenitore.

e. Per User (Utente) immetti il nome utente per l'utilizzo all'interno del container.

f. Attiva Abilita filesystem di sola lettura per rimuovere l'accesso in scrittura al volume.

g. Attiva Privileged per concedere al job container autorizzazioni elevate sull'istanza host, simili a quelle dell'utente root.

h. Per Command, inserisci i comandi nel campo come equivalente dell'array di stringhe JSON.

Questo parametro viene mappato su `Cmd` nella sezione Crea un container di Docker Remote API e il parametro `COMMAND` su docker run. Per ulteriori informazioni sul Docker`CMD` parametro, vedete https://docs.docker.com/engine/ reference/builder/ #cmd.

> **Note**
>
> È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

i. Per le credenziali del repository, facoltativo, inserisci l'ARN del segreto contenente le tue credenziali.

j. Per le variabili di ambiente, facoltativo, scegli Aggiungi variabili di ambiente per aggiungere variabili di ambiente da passare al contenitore.

k. Nella sezione Parametri Linux - opzionale:

i. Attiva Abilita processo init per eseguire un processo init all'interno del container.

ii. Per Dimensione della memoria condivisa, inserisci la dimensione (in MiB) del volume / dev/shm

iii. Per Dimensione massima di swap, inserisci la quantità totale di memoria di swap (in MiB) che il contenitore può utilizzare.

iv. Per Swappiness, inserite un valore compreso tra 0 e 100 per indicare il comportamento di swappiness del contenitore. Se non specificate un valore e lo scambio è abilitato, il valore predefinito è 60.

v. Per i dispositivi, scegli Aggiungi dispositivo per aggiungere un dispositivo:

A. Per Container path (Percorso container), specifica il percorso dell'istanza del container per esporre il dispositivo mappato all'istanza host. Se lasci vuoto questo campo, il percorso dell'host viene utilizzato nel contenitore.

B. Per Host path (Percorso host), specifica il percorso di un dispositivo nell'istanza host.

C. Per Autorizzazioni, scegli una o più autorizzazioni da applicare al dispositivo. Le autorizzazioni disponibili sono READ, WRITE e MKNOD.

vi. Per Tmpfs, scegli Aggiungi tmpfs per aggiungere un mount. `tmpfs`

l.

> **Note**
>
> La registrazione di Firelens deve essere eseguita in un contenitore dedicato. Per configurare la registrazione Firelens:
>
> - In ogni contenitore, ad eccezione del contenitore firelens dedicato, imposta il driver Logging su `awsfirelens`
> - Nel tuo contenitore Firelens, imposta la configurazione Firelens - opzionale e la configurazione Logging - opzionale sulla destinazione di registrazione

Nella sezione Configurazione Firelens - opzionale:

> **Important**
>
> AWS Batch applica la modalità `host` di rete su job Amazon ECS non MNP e non Fargate. L'utente root è richiesto per Amazon ECS Firelens. Quando si eseguono attività che utilizzano la modalità di `host` rete, Amazon ECS sconsiglia di eseguire contenitori utilizzando l'utente root (UID 0) per una maggiore sicurezza. Pertanto, tutti i lavori ECS non MNP e non Fargate con registrazione Firelens non soddisferanno le migliori pratiche di sicurezza.

i. Per Tipo**`fluentd`**, **`fluentbit`** scegliete una delle due opzioni o.

ii. In Opzioni, inserite la name/value coppia di opzioni. È possibile aggiungere altre opzioni utilizzando l'opzione Aggiunta.

m. Nella sezione Configurazione della registrazione, opzionale:

i. Per Log driver, scegli il driver di registro da usare. Per ulteriori informazioni sui driver di registro disponibili, LogConfiguration vedi:LogDriver.

> **Note**
>
> Per impostazione predefinita, viene utilizzato il driver di `awslogs` registro.

ii. Per Opzioni, scegli Aggiungi opzione per aggiungere un'opzione. Immettete una coppia nome-valore, quindi scegliete l'opzione Aggiungi.

iii. Per Segreti, scegli Aggiungi segreto. Inserisci una coppia nome-valore, quindi scegli Aggiungi segreto per aggiungere un segreto.

> **Tip**
>
> Per ulteriori informazioni, vedi:SecretOptions. LogConfiguration

n. Per Punti di montaggio, facoltativo, scegliete Aggiungi punti di montaggio per aggiungere punti di montaggio per i volumi di dati. È necessario specificare il volume di origine e il percorso del contenitore.

o. Per Segreti: facoltativo, scegli Aggiungi segreto per aggiungere un segreto. Quindi, inserisci una coppia nome-valore e scegli Aggiungi segreto.

> **Tip**
>
> Per ulteriori informazioni, vedi:SecretOptions. LogConfiguration

p. Per Ulimiti: facoltativo, scegli Aggiungi ulimit per aggiungere un valore per il contenitore. `ulimits` Inserisci i valori Name, Soft limit e Hard limit, quindi scegli Aggiungi ulimit.

q. Per Dipendenze, facoltativo, scegli Aggiungi dipendenze dal contenitore. Scegli il nome del contenitore e il suo stato per determinare quando verrà avviato.

17. Se hai configurato un solo contenitore, devi scegliere Aggiungi contenitore e completare la configurazione del nuovo contenitore. Altrimenti, scegli Avanti per rivedere.

Creare una definizione di processo parallelo multinodo

Prima di poter eseguire i lavori in AWS Batch, è necessario creare una definizione di processo. Questo processo varia leggermente tra processi paralleli a nodo singolo e multinodo. Questo argomento descrive in particolare come creare una definizione di processo per un processo parallelo a AWS Batch più nodi (noto anche come pianificazione di gruppo). Per ulteriori informazioni, consulta Lavori paralleli multinodo.

> **Note**
>
> AWS Fargate non supporta i lavori paralleli multinodo.

Indice

Tutorial: crea una definizione di processo parallelo multinodo sulle risorse Amazon EC2

Per creare una definizione di processo parallelo multinodo su risorse Amazon Elastic Compute Cloud EC2 (Amazon).

> **Note**
>
> Per creare una definizione di processo a nodo singolo, consulta. Crea una definizione di lavoro a nodo singolo sulle risorse Amazon EC2

Per creare una definizione di processo parallelo a più nodi sulle EC2 risorse Amazon:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
3. Nel riquadro di navigazione, scegli Job definition.
4. Scegli Create (Crea).
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon). EC2

6. Per Abilita parallelo a più nodi, attiva il parallelo a più nodi.
7. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può contenere fino a 128 caratteri e contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
8. (Facoltativo) Per il timeout di esecuzione, specificate il numero massimo di secondi in cui desiderate che i tentativi di processo vengano eseguiti. Se un tentativo supera la durata del timeout, il tentativo viene interrotto e passa a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout.
9. (Facoltativo) Attiva la priorità di pianificazione. Immettete un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore rispetto ai valori inferiori.
10. (Facoltativo) In Tentativi di lavoro, immettete il numero di volte in cui si AWS Batch tenta di spostare il lavoro allo `RUNNABLE` stato. Immettete un numero compreso tra 1 e 10.
11. (Facoltativo) Per le condizioni della strategia Retry, scegliete Aggiungi valutazione all'uscita. Inserisci almeno un valore di parametro, quindi scegli un'azione. Per ogni set di condizioni, l'azione deve essere impostata su Riprova o Esci. Queste azioni significano quanto segue:

    - Riprova: AWS Batch riprova fino al raggiungimento del numero di tentativi di lavoro specificato.
    - Esci: AWS Batch interrompe l'esecuzione di un nuovo tentativo.

    > **Important**
    >
    > Se scegli Aggiungi valutazione all'uscita, devi configurare almeno un parametro e scegliere un'azione o scegliere Rimuovi valutazione all'uscita.

12. (Facoltativo) Espandi Tag, quindi scegli Aggiungi tag per aggiungere tag alla risorsa. Inserisci una chiave e un valore opzionale, quindi scegli Aggiungi tag. Puoi anche attivare i tag Propagate per propagare i tag dal processo e dalla definizione del processo al task Amazon ECS.
13. Scegli Pagina successiva.
14. In Number of nodes (Numero di nodi), inserire il numero totale di nodi da utilizzare per il processo.
15. Per Main node (Nodo principale), immettere il nodo indice da utilizzare per il nodo principale. L'indice del nodo principale predefinito è `0`.
16. Per Tipo di istanza, scegli un tipo di istanza.

> **Note**
>
> Il tipo di istanza scelto si applica a tutti i nodi.

17. Per Parametri, scegliete Aggiungi parametri per aggiungere segnaposto sostitutivi dei parametri come coppie Chiave e Valore facoltative.
18. Nella sezione Intervalli di nodi:

    a. Seleziona Aggiungi intervallo di nodi. Questo crea una sezione relativa all'intervallo di nodi.

    b. Per i Target nodes (Nodi di destinazione), specificare l'intervallo per il gruppo di nodi, utilizzando la notazione *`range_start`*`:`*`range_end`*.

    È possibile creare fino a cinque intervalli di nodi per i nodi specificati per il job. Gli intervalli di nodo utilizzano il valore di indice per un nodo e l'indice di nodo inizia da 0. Assicurati che il valore dell'indice finale dell'intervallo del gruppo di nodi finale sia inferiore di uno rispetto al numero di nodi specificato. Ad esempio, supponiamo di aver specificato 10 nodi e di voler utilizzare un singolo gruppo di nodi. Quindi, l'intervallo finale è 9.

    c. Per Image, scegli l'Dockerimmagine da usare per il tuo lavoro. Per impostazione predefinita, le immagini nel registro Docker Hub sono disponibili. Puoi anche specificare altri repository con *`repository-url`*`/`*`image`*`:`*`tag`*. Il nome può contenere fino a 225 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini (-), caratteri di sottolineatura (_), due 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.

    > **Note**
    >
    > L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

    - Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]` denominazione (ad esempio,). `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*

- Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository[:tag]` denominazione completa. Ad esempio, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*
- 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 repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

d. Per Command, inserisci i comandi nel campo come equivalente dell'array di stringhe JSON.

   Questo parametro viene mappato su `Cmd` nella sezione Crea un container di Docker Remote API e il parametro `COMMAND` su docker run. Per ulteriori informazioni sul DockerCMD parametro, consulta https://docs.docker.com/engine/ reference/builder/ #cmd.

   > **Note**
   >
   > È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

e. Per v CPUs, specifica il numero di v da CPUs riservare per il contenitore. Questo parametro è mappato a `CpuShares` nella sezione Create a container di Docker Remote API e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Devi specificare almeno un vCPU.

f. Per Memory (Memoria) specifica il limite rigido (in MiB) della memoria da presentare al container del processo. Se il contenitore tenta di superare la memoria specificata qui, il contenitore viene fermato. Questo parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria.

   > **Note**
   >
   > Per massimizzare l'utilizzo delle risorse, è possibile fornire ai job quanta più memoria possibile per un particolare tipo di istanza. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

g. (Facoltativo) Per Numero di GPUs, specificate il numero di GPUs utilizzi del lavoro. Il processo viene eseguito su un contenitore il GPUs cui numero specificato è bloccato in quel contenitore.

h. (Facoltativo) Per Job role, puoi specificare un ruolo IAM che fornisca al contenitore del tuo job le autorizzazioni per utilizzare il AWS APIs. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, compresi i prerequisiti di configurazione, consulta IAM Roles for Tasks nella Amazon Elastic Container Service Developer Guide.

> **Note**
>
> Per i lavori eseguiti sulle risorse di Fargate, è richiesto un ruolo lavorativo.

> **Note**
>
> Qui vengono mostrati solo i ruoli con la relazione di trust Amazon Elastic Container Service Task Role. Per ulteriori informazioni sulla creazione di un ruolo IAM per i tuoi AWS Batch lavori, consulta Creating an IAM Role and Policy for your Tasks nella Amazon Elastic Container Service Developer Guide.

i. (Facoltativo) Per il ruolo Execution, specifica un ruolo IAM che conceda agli agenti del contenitore Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, consulta i ruoli IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

19. (Facoltativo) Espandi la configurazione aggiuntiva:

a. Per le variabili di ambiente, scegli Aggiungi variabile di ambiente per aggiungere variabili di ambiente come coppie nome-valore. Queste variabili vengono passate al contenitore.

b. Per la configurazione del ruolo Job, puoi specificare un ruolo IAM che fornisca al contenitore del tuo job le autorizzazioni per utilizzare il AWS APIs. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, compresi i prerequisiti di configurazione, consulta IAM Roles for Tasks nella Amazon Elastic Container Service Developer Guide.

> **Note**
>
> Per i lavori eseguiti sulle risorse di Fargate, è richiesto un ruolo lavorativo.

> **Note**
>
> Qui vengono mostrati solo i ruoli con la relazione di trust Amazon Elastic Container Service Task Role. Per ulteriori informazioni su come creare un ruolo IAM per i tuoi AWS Batch lavori, consulta Creating an IAM Role and Policy for your Tasks nella Amazon Elastic Container Service Developer Guide.

c. Per il ruolo Execution, specifica un ruolo IAM che conceda agli agenti del contenitore Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, consulta i ruoli IAM di esecuzione delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

20. Nella sezione Configurazione di sicurezza:

    a. (Facoltativo) Per assegnare privilegi elevati al container del job sull'istanza host (simili a quelli dell'**`root`**utente), attiva Privileged. Questo parametro è mappato a `Privileged` nella sezione Create a container di Docker Remote API e l'opzione `--privileged` a docker run.

    b. (Facoltativo) Per Utente, inserisci il nome utente da utilizzare all'interno del contenitore. Questo parametro è mappato a `User` nella sezione Create a container di Docker Remote API e l'opzione `--user` a docker run.

    c. (Facoltativo) Per Segreti, scegli Aggiungi segreto per aggiungere segreti come coppie nome-valore. Questi segreti sono esposti nel contenitore. Per ulteriori informazioni, LogConfiguration vedi:SecretOptions.

21. Nella sezione di configurazione Linux:

    a. Attiva Abilita filesystem di sola lettura per rimuovere l'accesso in scrittura al volume.

    b. (Facoltativo) Attiva Abilita init processo per eseguire un `init` processo all'interno del contenitore. Questo processo inoltra segnali e raccoglie i processi.

    c. In Dimensione della memoria condivisa, inserisci la dimensione (in MiB) del `/dev/shm` volume.

d. Per Dimensione massima di swap, inserisci la quantità totale di memoria di swap (in MiB) che il contenitore può utilizzare.

e. Per Swappiness, inserite un valore compreso tra 0 e 100 per indicare il comportamento di swappiness del contenitore. Se non specificate un valore e lo scambio è abilitato, il valore predefinito è 60. Per ulteriori informazioni, vedi:swappiness. LinuxParameters

f. (Facoltativo) Per i dispositivi, scegli Aggiungi dispositivo per aggiungere un dispositivo:

   i. Per Container path (Percorso container), specifica il percorso dell'istanza del container per esporre il dispositivo mappato all'istanza host. Se lasci vuoto questo campo, il percorso dell'host viene utilizzato nel contenitore.

   ii. Per Host path (Percorso host), specifica il percorso di un dispositivo nell'istanza host.

   iii. Per Autorizzazioni, scegli una o più autorizzazioni da applicare al dispositivo. Le autorizzazioni disponibili sono READ, WRITE e MKNOD.

22. (Facoltativo) Per i punti di montaggio, scegliete Aggiungi configurazione dei punti di montaggio per aggiungere punti di montaggio per i volumi di dati. È necessario specificare il volume di origine e il percorso del contenitore. Questi punti di montaggio vengono passati al Docker demone su un'istanza del contenitore. Puoi anche scegliere di rendere il volume di sola lettura.

23. (Facoltativo) Per la configurazione Ulimits, scegli Aggiungi ulimit per aggiungere un `ulimits` valore per il contenitore. Inserisci i valori Name, Soft limit e Hard limit, quindi scegli Aggiungi ulimit.

24. (Facoltativo) Per la configurazione dei volumi, scegli Aggiungi volume per creare un elenco di volumi da passare al contenitore. Inserisci il nome e il percorso di origine per il volume, quindi scegli Aggiungi volume. Puoi anche scegliere di attivare Enable EFS.

25. (Facoltativo) Per Tmpfs, scegli Aggiungi tmpfs per aggiungere un mount. `tmpfs`

26. Nella sezione Proprietà dell'attività:

    a. Per Ruolo di esecuzione: condizionale, scegli un ruolo per consentire agli agenti Amazon ECS di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni sulla creazione di un ruolo di esecuzione, consulta. Tutorial: Creare il ruolo di esecuzione IAM

    b.

> **Important**
>
> Per utilizzare il comando di esecuzione ECS, l'ambiente di calcolo deve soddisfare le considerazioni relative all'ambiente di calcolo per i lavori paralleli a più nodi.

Scegli il comando Enable ECS execute per abilitare l'accesso diretto alla shell del contenitore Amazon ECS e bypassare il sistema operativo host. Devi scegliere un ruolo Task.

> **Important**
>
> Il comando ECS execute richiede che il file system sia scrivibile.

c. Per il ruolo Task, scegli un ruolo Amazon ECS Identity and Access Management (IAM) per consentire al contenitore di effettuare chiamate AWS API per tuo conto. Per ulteriori informazioni, consulta il ruolo IAM delle attività di Amazon ECS nella Amazon Elastic Container Service Developer Guide.

27. (Facoltativo) Nella sezione Configurazione della registrazione:

a. Per Log driver, scegli il driver di registro da usare. Per ulteriori informazioni sui driver di registro disponibili, LogConfiguration vedi:LogDriver.

> **Note**
>
> Per impostazione predefinita, viene utilizzato il driver di `awslogs` registro.

b. Per Opzioni, scegli Aggiungi opzione per aggiungere un'opzione. Immettete una coppia nome-valore, quindi scegliete l'opzione Aggiungi.

c. Per Segreti, scegli Aggiungi segreto. Inserisci una coppia nome-valore, quindi scegli Aggiungi segreto per aggiungere un segreto.

> **Tip**
>
> Per ulteriori informazioni, vedi:SecretOptions. LogConfiguration

28. Scegli Pagina successiva.

29. Per la revisione della definizione di Job, rivedi i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea definizione del lavoro.

Modello di definizione del lavoro che utilizza ContainerProperties

Di seguito è riportato un modello di definizione del processo vuoto che include un singolo contenitore. È possibile utilizzare questo modello per creare la definizione del processo, che può quindi essere salvata in un file e utilizzata con l' AWS CLI `--cli-input-json`opzione. Per ulteriori informazioni su questi parametri, consultare JobDefinition.

> **Note**
>
> È possibile generare un modello di definizione del processo a contenitore singolo con il seguente AWS CLI comando:
>
> ```
> $ aws batch register-job-definition --generate-cli-skeleton
> ```

```
{
    "jobDefinitionName": "",
    "type": "container",
    "parameters": {
        "KeyName": ""
    },
    "schedulingPriority": 0,
    "containerProperties": {
        "image": "",
        "vcpus": 0,
        "memory": 0,
        "command": [
            ""
        ],
        "jobRoleArn": "",
        "executionRoleArn": "",
        "volumes": [
            {
                "host": {
                    "sourcePath": ""
                },
                "name": "",
                "efsVolumeConfiguration": {
                    "fileSystemId": "",
                    "rootDirectory": "",
                    "transitEncryption": "ENABLED",
```

```
                "transitEncryptionPort": 0,
                "authorizationConfig": {
                    "accessPointId": "",
                    "iam": "DISABLED"
                }
            }
        }
    ],
    "environment": [
        {
            "name": "",
            "value": ""
        }
    ],
    "mountPoints": [
        {
            "containerPath": "",
            "readOnly": true,
            "sourceVolume": ""
        }
    ],
    "readonlyRootFilesystem": true,
    "privileged": true,
    "ulimits": [
        {
            "hardLimit": 0,
            "name": "",
            "softLimit": 0
        }
    ],
    "user": "",
    "instanceType": "",
    "resourceRequirements": [
        {
            "value": "",
            "type": "MEMORY"
        }
    ],
    "linuxParameters": {
        "devices": [
            {
                "hostPath": "",
                "containerPath": "",
                "permissions": [
```

```
                        "WRITE"
                    ]
                }
            ],
            "initProcessEnabled": true,
            "sharedMemorySize": 0,
            "tmpfs": [
                {
                    "containerPath": "",
                    "size": 0,
                    "mountOptions": [
                        ""
                    ]
                }
            ],
            "maxSwap": 0,
            "swappiness": 0
        },
        "logConfiguration": {
            "logDriver": "syslog",
            "options": {
                "KeyName": ""
            },
            "secretOptions": [
                {
                    "name": "",
                    "valueFrom": ""
                }
            ]
        },
        "secrets": [
            {
                "name": "",
                "valueFrom": ""
            }
        ],
        "networkConfiguration": {
            "assignPublicIp": "DISABLED"
        },
        "fargatePlatformConfiguration": {
            "platformVersion": ""
        }
    },
    "nodeProperties": {
```

```
        "numNodes": 0,
        "mainNode": 0,
        "nodeRangeProperties": [
            {
                "targetNodes": "",
                "container": {
                    "image": "",
                    "vcpus": 0,
                    "memory": 0,
                    "command": [
                        ""
                    ],
                    "jobRoleArn": "",
                    "executionRoleArn": "",
                    "volumes": [
                        {
                            "host": {
                                "sourcePath": ""
                            },
                            "name": "",
                            "efsVolumeConfiguration": {
                                "fileSystemId": "",
                                "rootDirectory": "",
                                "transitEncryption": "DISABLED",
                                "transitEncryptionPort": 0,
                                "authorizationConfig": {
                                    "accessPointId": "",
                                    "iam": "ENABLED"
                                }
                            }
                        }
                    ],
                    "environment": [
                        {
                            "name": "",
                            "value": ""
                        }
                    ],
                    "mountPoints": [
                        {
                            "containerPath": "",
                            "readOnly": true,
                            "sourceVolume": ""
                        }
```

```
                    ],
                    "readonlyRootFilesystem": true,
                    "privileged": true,
                    "ulimits": [
                        {
                            "hardLimit": 0,
                            "name": "",
                            "softLimit": 0
                        }
                    ],
                    "user": "",
                    "instanceType": "",
                    "resourceRequirements": [
                        {
                            "value": "",
                            "type": "MEMORY"
                        }
                    ],
                    "linuxParameters": {
                        "devices": [
                            {
                                "hostPath": "",
                                "containerPath": "",
                                "permissions": [
                                    "WRITE"
                                ]
                            }
                        ],
                        "initProcessEnabled": true,
                        "sharedMemorySize": 0,
                        "tmpfs": [
                            {
                                "containerPath": "",
                                "size": 0,
                                "mountOptions": [
                                    ""
                                ]
                            }
                        ],
                        "maxSwap": 0,
                        "swappiness": 0
                    },
                    "logConfiguration": {
                        "logDriver": "awslogs",
```

```
                    "options": {
                        "KeyName": ""
                    },
                    "secretOptions": [
                        {
                            "name": "",
                            "valueFrom": ""
                        }
                    ]
                },
                "secrets": [
                    {
                        "name": "",
                        "valueFrom": ""
                    }
                ],
                "networkConfiguration": {
                    "assignPublicIp": "DISABLED"
                },
                "fargatePlatformConfiguration": {
                    "platformVersion": ""
                }
            }
        }
    ]
},
"retryStrategy": {
    "attempts": 0,
    "evaluateOnExit": [
        {
            "onStatusReason": "",
            "onReason": "",
            "onExitCode": "",
            "action": "RETRY"
        }
    ]
},
"propagateTags": true,
"timeout": {
    "attemptDurationSeconds": 0
},
"tags": {
    "KeyName": ""
},
```

```
    "platformCapabilities": [
        "EC2"
    ],
    "eksProperties": {
        "podProperties": {
            "serviceAccountName": "",
            "hostNetwork": true,
            "dnsPolicy": "",
            "containers": [
                {
                    "name": "",
                    "image": "",
                    "imagePullPolicy": "",
                    "command": [
                        ""
                    ],
                    "args": [
                        ""
                    ],
                    "env": [
                        {
                            "name": "",
                            "value": ""
                        }
                    ],
                    "resources": {
                        "limits": {
                            "KeyName": ""
                        },
                        "requests": {
                            "KeyName": ""
                        }
                    },
                    "volumeMounts": [
                        {
                            "name": "",
                            "mountPath": "",
                            "readOnly": true
                        }
                    ],
                    "securityContext": {
                        "runAsUser": 0,
                        "runAsGroup": 0,
                        "privileged": true,
```

```
                        "readOnlyRootFilesystem": true,
                        "runAsNonRoot": true
                    }
                }
            ],
            "volumes": [
                {
                    "name": "",
                    "hostPath": {
                        "path": ""
                    },
                    "emptyDir": {
                        "medium": "",
                        "sizeLimit": ""
                    },
                    "secret": {
                        "secretName": "",
                        "optional": true
                    }
                }
            ]
        }
    }
}
```

Parametri di definizione del lavoro per ContainerProperties

Le definizioni di Job utilizzate `ContainerProperties`sono suddivise in più parti:

- Il nome della definizione del processo
- Il tipo di definizione del processo
- I valori predefiniti del segnaposto per la sostituzione dei parametri
- Le proprietà del contenitore per il lavoro
- Le proprietà di Amazon EKS per la definizione del processo necessarie per i lavori eseguiti sulle risorse Amazon EKS
- Le proprietà del nodo necessarie per un processo parallelo a più nodi
- Le funzionalità della piattaforma necessarie per i lavori eseguiti sulle risorse di Fargate
- I dettagli di propagazione dei tag predefiniti della definizione del processo
- La strategia di riprova predefinita per la definizione del processo

- La priorità di pianificazione predefinita per la definizione del processo
- I tag predefiniti per la definizione del processo
- Il timeout predefinito per la definizione del processo

Indice

Nome della definizione del processo

`jobDefinitionName`

Quando si registra una definizione di processo, è necessario specificare un nome, Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_). Alla prima definizione di processo registrata con quel nome viene assegnata una revisione pari a 1. Tutte le successive definizioni del processo registrate con tale nome vengono contrassegnate con un numero di revisione incrementale.

Tipo: stringa

Campo obbligatorio: sì

Tipo

`type`

Quando si registra una definizione di processo, è necessario specificare il tipo di processo. Se il processo viene eseguito su risorse Fargate, `multinode` non è supportato. Per ulteriori informazioni sui processi in parallelo a più nodi, consulta Creare una definizione di processo parallelo multinodo.

-Tipo: stringa

Valori validi: `container` | `multinode`

Campo obbligatorio: sì

Parametri

`parameters`

Quando invii un lavoro, puoi specificare parametri che sostituiscono i segnaposto o sostituiscono i parametri di definizione del lavoro predefiniti. I parametri delle richieste di invio del processo hanno priorità rispetto ai valori predefiniti di una definizione del processo. Ciò significa che è possibile utilizzare la stessa definizione di processo per più lavori che utilizzano lo stesso formato. È inoltre possibile modificare a livello di codice i valori del comando al momento dell'invio.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Quando si registra una definizione di processo, è possibile utilizzare segnaposto di sostituzione dei parametri nel campo `command` delle proprietà del container del processo. La sintassi è esposta di seguito.

```
"command": [
    "ffmpeg",
    "-i",
    "Ref::inputfile",
    "-c",
    "Ref::codec",
    "-o",
    "Ref::outputfile"
```

```
]
```

Nell'esempio precedente, sono presenti segnaposto per la sostituzione del parametro *Ref::inputfile*, *Ref::codec* e *Ref::outputfile* nel comando. È possibile utilizzare l'`parameters`oggetto nella definizione del processo per impostare valori predefiniti per questi segnaposto. Ad esempio, per impostare un valore predefinito per il segnaposto *Ref::codec*, è necessario specificare quanto segue nella definizione del processo:

```
"parameters" : {"codec" : "mp4"}
```

Quando questa definizione di processo viene inviata per l'esecuzione, l'*Ref::codec*argomento nel comando per il contenitore viene sostituito con il valore predefinito,. `mp4`

Proprietà del contenitore

Quando registri una definizione di processo, specifica un elenco di proprietà del contenitore che vengono passate al demone Docker su un'istanza del contenitore quando il lavoro viene inserito. Di seguito sono indicate le proprietà del container consentite in una definizione di processo. Per i processi a nodo singolo, queste proprietà del container vengono impostate a livello di definizione del processo. Per i processi paralleli a più nodi, le proprietà del container vengono impostate nel livello Proprietà del nodo, per ciascun gruppo di nodi.

`command`

Il comando che viene inviato al container. Questo parametro viene mappato su `Cmd` nella sezione Crea un container di Docker Remote API e il parametro `COMMAND` su docker run. Per ulteriori informazioni sul `CMD` parametro Docker, vedere reference/builder/ #cmd. https://docs.docker.com/engine/

```
"command": ["string", ...]
```

Tipo: array di stringhe

Campo obbligatorio: no

`environment`

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 d'ambiente non crittografate per informazioni sensibili, ad esempio dati di credenziali.

> **Note**
>
> Le variabili di ambiente non devono iniziare con. `AWS_BATCH` Questa convenzione di denominazione è riservata alle variabili impostate dal AWS Batch servizio.

Tipo: array di coppie chiave-valore

Campo obbligatorio: no

`name`

Il nome della variabile di ambiente.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `environment`.

`value`

Il valore della variabile di ambiente.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `environment`.

```
"environment" : [
    { "name" : "envName1", "value" : "envValue1" },
    { "name" : "envName2", "value" : "envValue2" }
]
```

`executionRoleArn`

Quando registri una definizione di lavoro, puoi specificare un ruolo IAM. Il ruolo fornisce all'agente container Amazon ECS le autorizzazioni per richiamare le azioni API specificate nelle politiche associate per tuo conto. I lavori eseguiti su risorse Fargate devono fornire un ruolo di esecuzione. Per ulteriori informazioni, consulta AWS Batch Ruolo di esecuzione IAM.

- Tipo: stringa

Campo obbligatorio: no

`fargatePlatformConfiguration`

La configurazione della piattaforma per i lavori eseguiti sulle risorse di Fargate. I lavori eseguiti su EC2 risorse non devono specificare questo parametro.

Tipo: oggetto FargatePlatformConfiguration

Campo obbligatorio: no

`platformVersion`

La versione della piattaforma AWS Fargate da utilizzare per i lavori o per `LATEST` utilizzare una versione recente e approvata della piattaforma Fargate AWS .

- Tipo: stringa

Impostazione predefinita: `LATEST`

Campo obbligatorio: no

`image`

L'immagine utilizzata per iniziare un lavoro. Questa stringa viene trasmessa direttamente al daemon Docker. Le immagini nel registro Docker Hub sono disponibili di default. Puoi anche specificare altri repository con *`repository-url`*`/`*`image`*`:`*`tag`*. 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.

> **Note**
>
> L'architettura immagine Docker deve corrispondere all'architettura del processore delle risorse di calcolo su cui sono pianificate. Ad esempio, immagini Docker basate su ARM possono essere eseguite solo su risorse di calcolo basate su ARM.

- Le immagini negli archivi pubblici di Amazon ECR utilizzano le convenzioni complete `registry/repository[:tag]` o di `registry/repository[@digest]` denominazione (ad esempio,). `public.ecr.aws/`*`registry_alias`*`/`*`my-web-app`*`:`*`latest`*

- Le immagini nei repository Amazon ECR utilizzano la convenzione di `registry/repository:[tag]` denominazione completa. Ad esempio, *`aws_account_id`*`.dkr.ecr.`*`region`*`.amazonaws.com/`*`my-web-app`*`:`*`latest`*.
- 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 esempi, `amazon/amazon-ecs-agent`).
- Le immagini in altri repository online vengono ulteriormente qualificate tramite un nome di dominio (ad esempio `quay.io/assemblyline/ubuntu`).

Tipo: stringa

Campo obbligatorio: sì

`instanceType`

Il tipo di istanza da usare per un processo parallelo a più nodi. Attualmente tutti i gruppi di nodi in un processo parallelo a più nodi devono utilizzare lo stesso tipo di istanza. Questo parametro non è valido per i processi di container a nodo singolo o per i lavori eseguiti su risorse Fargate.

-Tipo: stringa

Campo obbligatorio: no

`jobRoleArn`

Quando registri una definizione di lavoro, puoi specificare un ruolo IAM. che fornisce al container del processo le autorizzazioni necessarie per chiamare le operazioni dell'API specificate nelle policy associate per conto dell'utente. Per ulteriori informazioni, consulta Ruoli IAM per le attività nella Guida per sviluppatori di Amazon Elastic Container Service.

-Tipo: stringa

Campo obbligatorio: no

`linuxParameters`

Modifiche specifiche di Linux che vengono applicate al container, ad esempio i dettagli per le mappature dei dispositivi.

```
"linuxParameters": {
    "devices": [
```

```
        {
            "hostPath": "string",
            "containerPath": "string",
            "permissions": [
                "READ", "WRITE", "MKNOD"
            ]
        }
    ],
    "initProcessEnabled": true|false,
    "sharedMemorySize": 0,
    "tmpfs": [
        {
            "containerPath": "string",
            "size": integer,
            "mountOptions": [
                "string"
            ]
        }
    ],
    "maxSwap": integer,
    "swappiness": integer
}
```

Tipo: oggetto LinuxParameters

Campo obbligatorio: no

`devices`

Elenco dei dispositivi mappati 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**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Tipo: matrice di oggetti Device

Campo obbligatorio: no

`hostPath`

Percorso in cui si trova il dispositivo disponibile nell'istanza del contenitore host.

Tipo: stringa

Campo obbligatorio: sì

`containerPath`

Si trova il percorso in cui è esposto il dispositivo nel contenitore. Se questo non è specificato, il dispositivo viene esposto nello stesso percorso del percorso host.

-Tipo: stringa

Campo obbligatorio: no

`permissions`

Autorizzazioni per il dispositivo nel container. Se questo non è specificato, le autorizzazioni sono impostate su `READWRITE`, e`MKNOD`.

Tipo: matrice di stringhe

Campo obbligatorio: no

Valori validi: `READ` | `WRITE` | `MKNOD`

`initProcessEnabled`

Se true, esegue 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. Per controllare la versione Docker Remote API nell'istanza di container, accedi all'istanza di container ed esegui il seguente comando: `sudo docker version | grep "Server API version"`

Tipo: Booleano

Campo obbligatorio: no

`maxSwap`

La quantità totale di memoria di swap (in MiB) che un job 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`. Per ulteriori informazioni, consulta la sezione relativa ai dettagli `--memory-swap` nella documentazione Docker.

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 `maxSwap` parametro viene omesso,

il contenitore utilizza la configurazione di swap per l'istanza del contenitore su cui viene eseguito. È necessario impostare un valore `maxSwap` per il parametro `swappiness` da utilizzare.

> **Note**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Tipo: integer

Campo obbligatorio: no

`sharedMemorySize`

Valore per le dimensioni (in MiB) del volume `/dev/shm`. Questo parametro è mappato all'opzione `--shm-size` su docker run.

> **Note**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Tipo: integer

Campo obbligatorio: no

`swappiness`

In questo modo è possibile ottimizzare il comportamento swappiness di memoria di un container. Un `swappiness` valore di `0` fa sì che lo scambio non avvenga a meno che non sia assolutamente necessario. 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. Se `maxSwap` è impostato su 0, il container non utilizza lo swap. Questo parametro è mappato all'opzione `--memory-swappiness` su docker run.

Quando utilizzi una configurazione swap container, considera quanto segue:

- Lo spazio di swap deve essere abilitato e allocato sull'istanza di container per consentire ai container di utilizzarlo.

> **Note**
>
> Gli Amazon ECS ottimizzati AMIs non hanno lo swap abilitato per impostazione predefinita. È necessario abilitare lo swap sull'istanza per utilizzare questa funzionalità. Per ulteriori informazioni, consulta Instance Store Swap Volumes nella Amazon EC2 User Guide o Come posso allocare la memoria per funzionare come spazio di swap in un' EC2 istanza Amazon utilizzando un file di swap? .

- I parametri dello spazio di swap sono supportati solo per le definizioni dei processi che utilizzano EC2 risorse.
- Se i parametri `maxSwap` e `swappiness` vengono omessi dalla definizione di un processo, per ogni container viene ripristinato il valore `swappiness` predefinito di 60. L'utilizzo totale dello swap è limitato a due volte la riserva di memoria del contenitore.

> **Note**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Tipo: integer

Campo obbligatorio: no

`tmpfs`

Il percorso del container, le opzioni di montaggio e la dimensione montaggio tmpfs.

Tipo: matrice di oggetti Tmpfs

> **Note**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Campo obbligatorio: no

`containerPath`

Il percorso assoluto del file nel container in cui è montato il volume tmpfs.

Tipo: stringa

Campo obbligatorio: sì

`mountOptions`

L'elenco delle opzioni di montaggio del volume tmpfs.

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`»

Tipo: matrice di stringhe

Campo obbligatorio: no

`size`

Le dimensioni (in MiB) del volume tmpfs.

Tipo: integer

Campo obbligatorio: sì

`logConfiguration`

La specifica di configurazione del registro per il lavoro.

Questo parametro è mappato a `LogConfig` nella sezione Crea un container di API Docker Remote 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 contenitore può utilizzare un driver di registrazione diverso dal demone Docker specificando un driver di registro con questo parametro nella definizione del contenitore. Per utilizzare un driver di registrazione diverso per un contenitore, il sistema di registro deve essere configurato sull'istanza del contenitore o su un altro server di registro per fornire opzioni di registrazione remota. Per ulteriori informazioni sulle opzioni per diversi driver di log supportati, consulta Configure logging drivers (Configurazione dei driver di log) nella documentazione di Docker.

> **Note**
>
> AWS Batch attualmente supporta un sottoinsieme dei driver di registrazione disponibili per il demone Docker (mostrati nel tipo di dati). LogConfiguration

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container. Per controllare la versione Docker Remote API nell'istanza di container, accedi all'istanza di container ed esegui il seguente comando: `sudo docker version | grep "Server API version"`

```
"logConfiguration": {
    "devices": [
        {
            "logDriver": "string",
            "options": {
                "optionName1" : "optionValue1",
                "optionName2" : "optionValue2"
            }
            "secretOptions": [
              {
                  "name" : "secretOptionName1",
                  "valueFrom" : "secretOptionArn1"
              },
              {
                  "name" : "secretOptionName2",
                  "valueFrom" : "secretOptionArn2"
              }
            ]
        }
    ]
}
```

Tipo: oggetto LogConfiguration

Campo obbligatorio: no

`logDriver`

Il driver di registro da utilizzare per il lavoro. Per impostazione predefinita, AWS Batch abilita il driver di `awslogs` registro. I valori validi elencati per questo parametro sono driver di log con i quali l'agente del container Amazon ECS può comunicare per impostazione predefinita.

Questo parametro è mappato a `LogConfig` nella sezione Crea un container di API Docker Remote e l'opzione `--log-driver` a docker run. Per impostazione predefinita, i job utilizzano lo stesso driver di registrazione utilizzato dal demone Docker. Tuttavia, il job può utilizzare un driver di registrazione diverso dal daemon Docker specificando un driver di log con questo parametro nella definizione del processo. Se si desidera specificare un altro driver di

registrazione per un job, il sistema di log deve essere configurato sull'istanza del contenitore nell'ambiente di calcolo. Oppure, in alternativa, configuralo su un altro server di registro per fornire opzioni di registrazione remota. Per ulteriori informazioni sulle opzioni per diversi driver di log supportati, consulta Configure logging drivers (Configurazione dei driver di log) nella documentazione di Docker.

> **Note**
>
> AWS Batch attualmente supporta un sottoinsieme dei driver di registrazione disponibili per il demone Docker. Ulteriori driver di log potranno essere disponibili nei rilasci futuri dell'agente del container Amazon ECS.

I driver di log supportati sono `awslogs`, `fluentd`, `gelf`, `json-file`, `journald`, `logentries`, `syslog` e `splunk`.

> **Note**
>
> I lavori eseguiti sulle risorse Fargate sono limitati ai driver `awslogs` e `splunk` log.

Questo parametro richiede la versione 1.18 o successiva di Docker Remote API sull'istanza di container. Per controllare la versione Docker Remote API nell'istanza di container, accedi all'istanza di container ed esegui il seguente comando: `sudo docker version | grep "Server API version"`

> **Note**
>
> L'agente container Amazon ECS che viene eseguito su un'istanza di contenitore deve registrare i driver di registrazione disponibili su quell'istanza con la variabile di `ECS_AVAILABLE_LOGGING_DRIVERS` ambiente. Altrimenti, i contenitori posizionati su quell'istanza non possono utilizzare queste opzioni di configurazione del registro. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Container Service.

`awslogs`

Speciifica il driver di registrazione Amazon CloudWatch Logs. Per ulteriori informazioni, consulta Usa il driver di registro awslogs il driver di registrazione di Amazon CloudWatch Logs nella documentazione di Docker.

`fluentd`

Specifica il driver di log Fluentd. Per ulteriori informazioni, inclusi l'utilizzo e le opzioni, consulta il driver di registrazione Fluentd nella documentazione di Docker.

`gelf`

Specifica il driver di log GELF (Greylog Extended Format). Per ulteriori informazioni, inclusi l'utilizzo e le opzioni, consulta il driver di registrazione Graylog Extended Format nella documentazione Docker.

`journald`

Specifica il driver di log journald. Per ulteriori informazioni, incluso l'utilizzo e le opzioni, consulta il driver di registrazione Journald nella documentazione di Docker.

`json-file`

Specifica il driver di log del file JSON. Per ulteriori informazioni, incluso l'utilizzo e le opzioni, consulta il driver di registrazione dei file JSON nella documentazione di Docker.

`splunk`

Specifica il driver di log Splunk. Per ulteriori informazioni, tra cui utilizzo e opzioni, consulta il driver di registrazione Splunk nella documentazione di Docker.

`syslog`

Specifica il driver di log syslog. Per ulteriori informazioni, incluso l'utilizzo e le opzioni, consulta il driver di registrazione Syslog nella documentazione di Docker.

Tipo: stringa

Campo obbligatorio: sì

Valori validi: `awslogs` | `fluentd` | `gelf` | `journald` | `json-file` | `splunk` | `syslog`

> **Note**
>
> Se disponi di un driver personalizzato non elencato in precedenza che desideri utilizzare con l'agente container Amazon ECS, puoi eseguire il fork del progetto Amazon ECS container agent disponibile su GitHub e personalizzarlo per funzionare con quel driver. Ti consigliamo di inviare le richieste pull per le modifiche che desideri siano incluse. Tuttavia, Amazon Web Services attualmente non supporta le richieste che eseguono copie modificate di questo software.

`options`

Opzioni di configurazione del registro da inviare a un driver di registro per il lavoro.

Questo parametro richiede la versione 1.19 o successiva di Docker Remote API sull'istanza di container.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

`secretOptions`

Un oggetto che rappresenta il segreto da inviare alla configurazione di log. Per ulteriori informazioni, consulta Specificare dati sensibili.

Tipo: matrice di oggetti

Campo obbligatorio: no

`name`

Il nome dell'opzione del driver di registro da impostare nel processo.

Tipo: stringa

Campo obbligatorio: sì

`valueFrom`

L'Amazon Resource Name (ARN) del segreto da esporre alla configurazione di log del contenitore. I valori supportati sono l'ARN completo del segreto di Secrets Manager o l'ARN completo del parametro nell'archivio dei parametri SSM.

> **Note**
>
> Se il parametro SSM Parameter Store esiste nella Regione AWS stessa operazione che stai avviando, puoi utilizzare l'ARN completo o il nome del parametro. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

Tipo: stringa

Campo obbligatorio: sì

`memory`

Questo parametro è obsoleto, utilizzalo al suo posto. *`resourceRequirements`*

Il numero di MiB di memoria riservati per il lavoro.

Come esempio di utilizzo`resourceRequirements`, se la definizione del processo contiene una sintassi simile alla seguente.

```
"containerProperties": {
  "memory": 512
}
```

La sintassi equivalente utilizzata `resourceRequirements` è la seguente.

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "MEMORY",
      "value": "512"
    }
  ]
}
```

Tipo: integer

Campo obbligatorio: sì

`mountPoints`

I punti di montaggio per i volumi di dati nel container. Questo parametro è mappato a `Volumes` nella sezione Create a container di Docker Remote API e l'opzione `--volume` a docker run.

```
"mountPoints": [
                {
                  "sourceVolume": "string",
                  "containerPath": "string",
                  "readOnly": true|false
                }
              ]
```

Tipo: array di oggetti

Campo obbligatorio: no

`sourceVolume`

Il nome del volume da montare.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `mountPoints`.

`containerPath`

Il percorso sul contenitore in cui montare il volume host.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `mountPoints`.

`readOnly`

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.

Tipo: Booleano

Campo obbligatorio: no

Impostazione predefinita: False

`networkConfiguration`

La configurazione di rete per i lavori eseguiti su risorse Fargate. I lavori eseguiti su EC2 risorse non devono specificare questo parametro.

```
"networkConfiguration": {
   "assignPublicIp": "string"
}
```

Tipo: array di oggetti

Campo obbligatorio: no

`assignPublicIp`

Indica se il processo ha un indirizzo IP pubblico. Questo è necessario se il job richiede l'accesso alla rete in uscita.

-Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Impostazione predefinita: `DISABLED`

`privileged`

Se il parametro è true, al container vengono assegnati privilegi elevati nell'istanza di container host (simile all'utente `root`). Questo parametro è mappato a `Privileged` nella sezione Create a container di Docker Remote API e l'opzione `--privileged` a docker run. Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate. Non fornirlo né specificarlo come falso.

```
"privileged": true|false
```

Tipo: Booleano

Campo obbligatorio: no

`readonlyRootFilesystem`

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.

```
"readonlyRootFilesystem": true|false
```

Tipo: Booleano

Campo obbligatorio: no

`resourceRequirements`

Il tipo e la quantità di una risorsa da assegnare a un container. Le risorse supportate includono `GPU`, `MEMORY` e `VCPU`.

```
"resourceRequirements" : [
  {
    "type": "GPU",
    "value": "number"
  }
]
```

Tipo: array di oggetti

Campo obbligatorio: no

`type`

Il tipo di risorsa da assegnare a un container. Le risorse supportate includono `GPU`, `MEMORY` e `VCPU`.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `resourceRequirements`.

`value`

La quantità di risorsa specificata da prenotare per il container. I valori variano in base al `type` specificato.

type="GPU"

Il numero di materiali fisici GPUs da riservare per il contenitore. Il numero di contenitori GPUs riservati per tutti i contenitori di un job non può superare il numero di risorse di calcolo disponibili GPUs sulla quale il job viene avviato.

type="MEMORY"

Il limite rigido (in MiB) della memoria da presentare al container. Se il container tenta di superare la memoria specificata qui, viene terminato. Questo parametro è mappato a `Memory` nella sezione Crea un container di API Docker Remote e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria. Questo è obbligatorio ma può essere specificato in più posizioni per i processi MNP (Multi-Node Parallel). Deve essere specificato almeno una volta per ogni nodo. Questo parametro è mappato a `Memory` nella sezione Crea un container di API Docker Remote e l'opzione `--memory` a docker run.

> **Note**
>
> Se stai cercando di massimizzare l'utilizzo delle risorse fornendo ai job quanta più memoria possibile per un particolare tipo di istanza, consulta. Gestione della memoria delle risorse di calcolo

Per i lavori eseguiti su risorse Fargate, `value` devono corrispondere a uno dei valori supportati. Inoltre, i `VCPU` valori devono essere uno dei valori supportati per quel valore di memoria.

VCPU	MEMORY
0,25 vCPU	512, 1024 e 2048 MiB
0,5 vCPU	1024-4096 MiB con incrementi di 1024 MiB
1 vCPU	2048-8192 MiB con incrementi di 1024 MiB
2 vCPU	4096-16384 MiB con incrementi di 1024 MiB
4 vCPU	8192-30720 MiB con incrementi di 1024 MiB
8 vCPU	16384-61440 MiB con incrementi di 4096 MiB
16 vCPU	32768-122880 MiB con incrementi di 8192 MiB

type="VCPU"

Il numero di v riservato al lavoro. CPUs Questo parametro è mappato a `CpuShares` nella sezione Crea un container di API Docker Remote e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Per i lavori eseguiti su EC2 risorse, è necessario specificare almeno una vCPU. È obbligatorio, ma può essere specificato in diversi punti. Deve essere specificato almeno una volta per ogni nodo.

Per i lavori eseguiti su risorse Fargate, `value` deve corrispondere a uno dei valori supportati e i `MEMORY` valori devono essere uno dei valori supportati per quel valore VCPU. I valori supportati sono 0,25, 0,5, 1, 2, 4, 8 e 16.

L'impostazione predefinita per la quota di conteggio delle risorse vCPU Fargate On-Demand è 6 v. CPUs Per ulteriori informazioni sulle quote Fargate, vedere Quote AWS Fargate nel. Riferimenti generali di Amazon Web Services

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `resourceRequirements`.

`secrets`

I segreti del lavoro che sono esposti come variabili di ambiente. Per ulteriori informazioni, consulta Specificare dati sensibili.

```
"secrets": [
    {
      "name": "secretName1",
      "valueFrom": "secretArn1"
    },
    {
      "name": "secretName2",
      "valueFrom": "secretArn2"
    }
    ...
]
```

Tipo: array di oggetti

Campo obbligatorio: no

`name`

Il nome della variabile di ambiente che contiene il segreto.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `secrets`.

`valueFrom`

Il segreto da esporre al container. I valori supportati sono l'Amazon Resource Name (ARN) completo del segreto Secrets Manager o l'ARN completo del parametro nell'SSM Parameter Store.

> **Note**
>
> Se il parametro SSM Parameter Store esiste nello Regione AWS stesso processo che stai avviando, puoi utilizzare l'ARN completo o il nome del parametro. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

- Tipo: stringa

Campo obbligatorio: sì, quando viene usato `secrets`.

`ulimits`

Un elenco di valori `ulimits` da impostare nel container. Questo parametro è mappato a `Ulimits` nella sezione Create a container di Docker Remote API e l'opzione `--ulimit` a docker run.

```
"ulimits": [
  {
    "name": string,
    "softLimit": integer,
    "hardLimit": integer
  }
  ...
]
```

Tipo: array di oggetti

Campo obbligatorio: no

`name`

`type` di `ulimit`.

- Tipo: stringa

  Campo obbligatorio: sì, quando viene usato `ulimits`.

`hardLimit`

Il limite rigido per il tipo `ulimit`.

Tipo: integer

Campo obbligatorio: sì, quando viene usato `ulimits`.

`softLimit`

Il limite flessibile per il tipo `ulimit`.

Tipo: integer

Campo obbligatorio: sì, quando viene usato `ulimits`.

`user`

Il nome utente per l'utilizzo 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.

```
"user": "string"
```

- Tipo: stringa

Campo obbligatorio: no

`vcpus`

Questo parametro è obsoleto, utilizzalo al suo posto. *`resourceRequirements`*

Il numero di v CPUs riservato per il contenitore.

Come esempio di utilizzo`resourceRequirements`, se la definizione del processo contiene righe simili a queste:

```
"containerProperties": {
```

```
  "vcpus": 2
}
```

Le righe equivalenti utilizzate `resourceRequirements` sono le seguenti.

```
"containerProperties": {
  "resourceRequirements": [
    {
      "type": "VCPU",
      "value": "2"
    }
  ]
}
```

Tipo: integer

Campo obbligatorio: sì

`volumes`

Quando si registra una definizione di processo, è possibile specificare un elenco di volumi da passare al daemon Docker su un'istanza di container. Nelle proprietà del container sono consentiti i seguenti parametri:

```
"volumes": [
  {
    "name": "string",
    "host": {
      "sourcePath": "string"
    },
    "efsVolumeConfiguration": {
      "authorizationConfig": {
        "accessPointId": "string",
        "iam": "string"
      },
      "fileSystemId": "string",
      "rootDirectory": "string",
      "transitEncryption": "string",
      "transitEncryptionPort": number
    }
  }
]
```

`name`

Nome del volume. Il nome può contenere un massimo di 255 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura. Nel parametro `sourceVolume` della definizione del container `mountPoints` viene fatto riferimento a questo nome.

- Tipo: stringa

Campo obbligatorio: no

`host`

Il contenuto del parametro `host` determina se il volume dati persiste nell'istanza di container host e dove viene archiviato. Se il parametro `host` è vuoto, il daemon Docker assegna automaticamente un percorso dell'host per il volume di dati. Tuttavia, non è garantito che i dati persistano dopo l'interruzione dell'esecuzione del contenitore ad essi associato.

> **Note**
>
> Questo parametro non è applicabile ai processi eseguiti sulle risorse di Fargate.

Tipo: oggetto

Campo obbligatorio: no

`sourcePath`

Il percorso sull'istanza di container dell'host presentato 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 di container dell'host finché non viene eliminato manualmente. Se il valore `sourcePath` non è presente nell'istanza di container host, viene creato automaticamente dal daemon Docker. Se la posizione è presente, i contenuti della cartella del percorso di origine vengono esportati.

- Tipo: stringa

Campo obbligatorio: no

`efsVolumeConfiguration`

Questo parametro viene specificato quando utilizzi un file system Amazon Elastic File System per l'archiviazione di attività. Per ulteriori informazioni, consulta Volumi Amazon EFS.

Tipo: oggetto

Campo obbligatorio: no

`authorizationConfig`

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

- Tipo: stringa

Campo obbligatorio: no

`accessPointId`

L'ID del punto di accesso Amazon EFS da utilizzare. Se viene specificato un punto di accesso, il valore della directory principale specificato in `EFSVolumeConfiguration` deve essere omesso o impostato su. `/` Ciò impone 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.

- Tipo: stringa

Campo obbligatorio: no

`iam`

Determina se utilizzare il ruolo IAM del AWS Batch lavoro definito in una definizione di processo 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 Usa i punti di accesso Amazon EFS.

- Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

`fileSystemId`

L'ID del file system Amazon EFS da utilizzare.

- Tipo: stringa

Campo obbligatorio: no

`rootDirectory`

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. Se lo specifichi `/`, ha lo stesso effetto dell'omissione di questo parametro. La lunghezza massima è 4.096 caratteri.

> **Important**
>
> Se un punto di accesso EFS è specificato in `authorizationConfig`, il parametro della directory principale deve essere omesso o impostato `/` su. Ciò impone il percorso impostato sul punto di accesso Amazon EFS.

- Tipo: stringa

Campo obbligatorio: no

`transitEncryption`

Indica 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 Amazon EFS IAM, 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.

- Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

`transitEncryptionPort`

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. Il valore deve essere compreso tra 0 e 65.535. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System.

Tipo: integer

Campo obbligatorio: no

Proprietà Amazon EKS

Un oggetto con varie proprietà specifiche dei processi basati su Amazon ECS. Questo non deve essere specificato per le definizioni dei job basate su Amazon ECS.

`podProperties`

Le proprietà delle risorse del Kubernetes pod di un lavoro.

Tipo: oggetto EksPodProperties

Campo obbligatorio: no

`containers`

Le proprietà del container utilizzato nel pod Amazon EKS.

Tipo: oggetto EksContainer

Campo obbligatorio: no

`args`

Un array di argomenti per il punto di ingresso. Se non è specificato, viene utilizzato il `CMD` dell'immagine del container. Corrisponde al `args` membro nella parte Entrypoint del Pod in. Kubernetes I riferimenti alle variabili di ambiente vengono espansi utilizzando l'ambiente del container.

Se la variabile di ambiente a cui si fa riferimento non esiste, il riferimento nel comando non viene modificato. Ad esempio, se il riferimento è a "`$(NAME1)`" e la variabile di ambiente `NAME1` non esiste, la stringa del comando rimarrà "`$(NAME1)`". `$$` viene sostituito con `$` e la stringa risultante non viene espansa. Ad esempio, `$$(VAR_NAME)` viene passato come `$(VAR_NAME)` a prescindere dall'esistenza della variabile di ambiente `VAR_NAME`.

Per ulteriori informazioni, vedete CMD nel riferimento a Dockerfile e Definire un comando e argomenti per un pod nella documentazione. Kubernetes

Tipo: matrice di stringhe

Campo obbligatorio: no

`command`

Il punto di ingresso per il container. Non viene eseguito in una shell (interprete di comandi). Se non è specificato, viene utilizzato il `ENTRYPOINT` dell'immagine del container. I riferimenti alle variabili di ambiente vengono espansi utilizzando l'ambiente del container.

Se la variabile di ambiente a cui si fa riferimento non esiste, il riferimento nel comando non viene modificato. Ad esempio, se il riferimento è a "`$(NAME1)`" e la variabile di ambiente `NAME1` non esiste, la stringa del comando rimarrà "`$(NAME1)`". `$$` viene sostituito con `$` e la stringa risultante non viene espansa. Ad esempio, `$$(VAR_NAME)` verrà passato come `$(VAR_NAME)` a prescindere dall'esistenza della variabile di ambiente `VAR_NAME`. Il punto di ingresso non può essere aggiornato. Per ulteriori informazioni, vedere ENTRYPOINT nel riferimento a Dockerfile e Definire un comando e argomenti per un contenitore e Entrypoint nella documentazione. Kubernetes

Tipo: matrice di stringhe

Campo obbligatorio: no

`env`

Le variabili di ambiente da passare a un container.

> **Note**
>
> Le variabili di ambiente non possono iniziare con "`AWS_BATCH`". Questa convenzione di denominazione è riservata alle variabili che impostano. AWS Batch

Tipo: matrice di oggetti EksContainerEnvironmentVariable

Campo obbligatorio: no

`name`

Il nome della variabile di ambiente.

Tipo: stringa

Campo obbligatorio: sì

`value`

Il valore della variabile di ambiente.

-Tipo: stringa

Campo obbligatorio: no

`image`

L'immagine Docker utilizzata per avviare il container.

Tipo: stringa

Campo obbligatorio: sì

`imagePullPolicy`

La policy di estrazione immagini per il container. I valori supportati sono `Always`, `IfNotPresent` e `Never`. Questo parametro per impostazione predefinita è `IfNotPresent`. Tuttavia, se è specificato il tag `:latest`, viene ripristinata l'impostazione predefinita `Always`. Per ulteriori informazioni, consulta Aggiornamento delle immagini nella Kubernetesdocumentazione.

-Tipo: stringa

Campo obbligatorio: no

`name`

Il nome del container. Se il nome non è specificato, viene utilizzato il nome predefinito, "`Default`". Ogni container in un pod deve avere un nome univoco.

-Tipo: stringa

Campo obbligatorio: no

`resources`

Il tipo e la quantità di risorse da assegnare a un container. Le risorse supportate includono `memory`, `cpu` e `nvidia.com/gpu`. Per ulteriori informazioni, consulta Gestione delle risorse per pod e contenitori nella Kubernetesdocumentazione.

Tipo: oggetto EksContainerResourceRequirements

Campo obbligatorio: no

`limits`

Il tipo e la quantità di risorse specificate da prenotare per il container. I valori variano in base al `name` specificato. Le risorse possono essere chieste utilizzando oggetti `limits` o `requests`.

memory

Il limite rigido di memoria (in MiB) per il container, espresso in numeri interi con il suffisso "Mi". Se il container prova a superare la memoria qui specificata, sarà terminato. Devi specificare almeno 4 MiB di memoria per un processo. La `memory` può essere specificata in `limits` e/o `requests`. Se la `memory` è specificata in entrambi, il valore specificato in `limits` deve essere uguale al valore specificato in `requests`.

> **Note**
>
> Per ottimizzare l'utilizzo delle risorse, fornisci ai processi quanta più memoria possibile per il tipo di istanza specifica in uso. Per scoprire come, consulta Gestione della memoria delle risorse di calcolo.

cpu

Il numero di CPUs questo è riservato al contenitore. I valori devono essere un multiplo pari di `0.25`. La `cpu` può essere specificata in `limits` e/o `requests`. Se la `cpu` è specificata in entrambi, il valore specificato in `limits` deve essere almeno pari al valore specificato in `requests`.

nvidia.com/gpu

Il numero di GPUs quello è riservato al contenitore. I valori devono essere un numero intero. La `memory` può essere specificata in `limits` e/o `requests`. Se la `memory` è specificata in entrambi, il valore specificato in `limits` deve essere uguale al valore specificato in `requests`.

Tipo: mappatura stringa a stringa

Valore dei vincoli di lunghezza: lunghezza minima di 1. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

`requests`

Il tipo e la quantità di risorse specificate da chiedere per il container. I valori variano in base al `name` specificato. Le risorse possono essere chieste utilizzando oggetti `limits` o `requests`.

memory

Il limite rigido di memoria (in MiB) per il container, espresso in numeri interi con il suffisso "Mi". Se il container prova a superare la memoria qui specificata, sarà terminato. Devi specificare almeno 4 MiB di memoria per un processo. La `memory` può essere specificata in `limits` e/o `requests`. Se la `memory` è specificata in entrambi, il valore specificato in `limits` deve essere uguale al valore specificato in `requests`.

> **Note**
>
> Se stai cercando di massimizzare l'utilizzo delle risorse fornendo ai job quanta più memoria possibile per un particolare tipo di istanza, consulta. Gestione della memoria delle risorse di calcolo

cpu

Il numero di questi CPUs è riservato al contenitore. I valori devono essere un multiplo pari di `0.25`. La `cpu` può essere specificata in `limits` e/o `requests`. Se la `cpu` è specificata in entrambi, il valore specificato in `limits` deve essere uguale al valore specificato in `requests`.

nvidia.com/gpu

Il numero di GPUs quelli è riservato al contenitore. I valori devono essere un numero intero. La `nvidia.com/gpu` può essere specificata in `limits` e/o `requests`. Se la `nvidia.com/gpu` è specificata in entrambi, il valore specificato in `limits` deve essere uguale al valore specificato in `requests`.

Tipo: mappatura stringa a stringa

Valore dei vincoli di lunghezza: lunghezza minima di 1. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

`securityContext`

Il contesto di sicurezza per un processo. Per ulteriori informazioni, consulta Configurare un contesto di sicurezza per un pod o un contenitore nella Kubernetesdocumentazione.

Tipo: oggetto EksContainerSecurityContext

Campo obbligatorio: no

`privileged`

Quando questo parametro è `true`, al container sono assegnate autorizzazioni elevati nell'istanza di container host. Il livello delle autorizzazioni è simile alle autorizzazioni degli `root` utenti. Il valore predefinito è `false`. Questo parametro corrisponde alle *`privileged`* politiche di sicurezza del pod Privileged nella documentazione. Kubernetes

Tipo: Booleano

Campo obbligatorio: no

`readOnlyRootFilesystem`

Quando questo parametro è `true`, al container viene assegnato l'accesso in sola lettura al file system root. Il valore predefinito è `false`. Questo parametro corrisponde alle `ReadOnlyRootFilesystem` politiche di sicurezza del pod Volumes and file system presenti nella Kubernetesdocumentazione.

Tipo: Booleano

Campo obbligatorio: no

`runAsGroup`

Quando questo parametro è specificato, il container viene eseguito come ID gruppo specificato (`gid`). Se questo parametro non è specificato, l'impostazione predefinita è il gruppo specificato nei medatati dell'immagine. Questo parametro corrisponde alle

`RunAsGroup` `MustRunAs` politiche di sicurezza del pod Utenti e gruppi presenti nella Kubernetesdocumentazione.

Tipo: long

Campo obbligatorio: no

`runAsNonRoot`

Quando questo parametro è specificato, il container viene eseguito come utente con un `uid` diverso da 0. Se questo parametro non è specificato, viene applicata tale regola. Questo parametro corrisponde alle `RunAsUser` `MustRunAsNonRoot` politiche di sicurezza del pod Utenti e gruppi presenti nella Kubernetesdocumentazione.

Tipo: long

Campo obbligatorio: no

`runAsUser`

Quando questo parametro è specificato, il container viene eseguito come ID utente specificato (`uid`). Se questo parametro non è specificato, l'impostazione predefinita è l'utente specificato nei medatati dell'immagine. Questo parametro corrisponde alle `RunAsUser` `MustRanAs` politiche di sicurezza del pod Utenti e gruppi presenti nella Kubernetesdocumentazione.

Tipo: long

Campo obbligatorio: no

`volumeMounts`

Il volume viene montato per un container per un processo Amazon EKS. Per ulteriori informazioni sui volumi e sui montaggi dei volumi inKubernetes, consulta Volumes nella Kubernetesdocumentazione.

Tipo: matrice di oggetti EksContainerVolumeMount

Campo obbligatorio: no

`mountPath`

Il percorso sul container in cui è montato il volume.

-Tipo: stringa

Campo obbligatorio: no

`name`

Il nome del montaggio del volume. Deve corrispondere al nome di uno dei volumi nel pod.

-Tipo: stringa

Campo obbligatorio: no

`readOnly`

Se il valore è `true`, il container avrà accesso in sola lettura al volume. In caso contrario, il container può scrivere sul volume. Il valore predefinito è `false`.

Tipo: Booleano

Campo obbligatorio: no

`dnsPolicy`

La policy DNS per il pod. Il valore predefinito è `ClusterFirst`. Se il parametro `hostNetwork` non è specificato, l'impostazione predefinita è `ClusterFirstWithHostNet`. `ClusterFirst` indica che qualunque query DNS non corrispondente al suffisso del dominio del cluster configurato viene inoltrata al server nomi upstream ereditato dal nodo. Se non è stato specificato alcun valore per `dnsPolicy` l'operazione RegisterJobDefinitionAPI, non viene restituito alcun valore `dnsPolicy` da nessuna delle DescribeJobDefinitionsoperazioni DescribeJobsAPI. L'impostazione delle specifiche del pod conterrà `ClusterFirst` o `ClusterFirstWithHostNet`, a seconda del valore del parametro `hostNetwork`. Per ulteriori informazioni, consulta la politica DNS di Pod nella Kubernetesdocumentazione.

Valori validi: `Default` | `ClusterFirst` | `ClusterFirstWithHostNet`

-Tipo: stringa

Campo obbligatorio: no

`hostNetwork`

Indica se il pod utilizza l'indirizzo IP di rete degli host. Il valore predefinito è `true`. L'impostazione di questa opzione `false` abilita il modello di rete Kubernetes pod. La maggior parte dei AWS Batch carichi di lavoro è di sola uscita e non richiede il sovraccarico

dell'allocazione IP per ogni pod per le connessioni in ingresso. Per ulteriori informazioni, consulta Host namespaces e Pod networking nella documentazione. Kubernetes

Tipo: Booleano

Campo obbligatorio: no

`serviceAccountName`

Il nome dell'account di servizio utilizzato per l'esecuzione del pod. Per ulteriori informazioni, consulta Account di Kubernetes servizio e Configurazione di un account di Kubernetes servizio per assumere un ruolo IAM nella Amazon EKS User Guide e Configurare gli account di servizio per i pod nella Kubernetesdocumentazione.

-Tipo: stringa

Campo obbligatorio: no

`volumes`

Specifica i volumi per una definizione di processo che utilizza risorse Amazon EKS.

Tipo: matrice di oggetti EksVolume

Campo obbligatorio: no

EmptyDir

Specifica la configurazione di un volume. Kubernetes `emptyDir` Un volume `emptyDir` viene creato per la prima volta quando un pod viene assegnato a un nodo. Esiste finché il pod funziona su quel nodo. Il volume `emptyDir` inizialmente è vuoto. Tutti i container nel pod possono leggere e scrivere i file nel volume `emptyDir`. Il volume `emptyDir`,tuttavia, può essere montato sullo stesso percorso o su percorsi diversi in ogni container. Quando un pod viene rimosso da un nodo per un motivo qualunque, i dati nella `emptyDir` vengono eliminati definitivamente. Per ulteriori informazioni, consulta EmptyDir nella documentazione. Kubernetes

Tipo: oggetto EksEmptyDir

Campo obbligatorio: no

medium

Il supporto per memorizzare il volume. Il valore predefinito è una stringa vuota, che utilizza la memoria del nodo.

""

(Impostazione predefinita) Utilizza la memoria su disco del nodo.

"Memory"

Usa il volume `tmpfs` supportato dalla RAM del nodo. I contenuti del volume vengono persi al riavvio del nodo e tutta la memoria sul volume viene conteggiata in base al limite di memoria del container.

- Tipo: stringa

Campo obbligatorio: no

SizeLimit

La dimensione massima del volume. Per impostazione predefinita, non è definita alcuna dimensione massima.

- Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

Percorso dell'host

Specifica la configurazione di un Kubernetes `hostPath` volume. Un volume `hostPath` monta un file o una directory esistente dal file system del nodo host nel pod. Per ulteriori informazioni, vedete HostPath nella Kubernetes documentazione.

Tipo: oggetto EksHostPath

Campo obbligatorio: no

path

Il percorso del file o della directory sull'host da montare in container nel pod.

- Tipo: stringa

Campo obbligatorio: no

nome

Nome del volume. Il nome deve essere consentito come nome sottodominio DNS. Per ulteriori informazioni, consulta i nomi dei sottodomini DNS nella documentazione. Kubernetes

Tipo: stringa

Campo obbligatorio: sì

Secret

Specifica la configurazione di un volume. Kubernetes `secret` Per ulteriori informazioni, vedere secret nella Kubernetesdocumentazione.

Tipo: oggetto EksSecret

Campo obbligatorio: no

facoltativo

Speciifica se è necessario definire il segreto o le chiavi del segreto.

Tipo: Booleano

Campo obbligatorio: no

NomeSegreto

Il nome del segreto. Il nome deve essere consentito come nome sottodominio DNS. Per ulteriori informazioni, consulta i nomi dei sottodomini DNS nella documentazione. Kubernetes

Tipo: stringa

Campo obbligatorio: sì

Funzionalità della piattaforma

`platformCapabilities`

Le funzionalità della piattaforma richieste dalla definizione del processo. Se non viene specificato alcun valore, il valore predefinito è `EC2`. Per i lavori eseguiti su risorse Fargate, `FARGATE` è specificato.

> **Note**
>
> Se il job viene eseguito su risorse Amazon EKS, non devi specificare`platformCapabilities`.

-Tipo: stringa

Valori validi: `EC2` | `FARGATE`

Campo obbligatorio: no

Propaga i tag

`propagateTags`

Specifica se propagare i tag dal processo o dalla definizione del processo all'attività Amazon ECS corrispondente. Se non viene specificato alcun valore, i tag non vengono propagati. I tag possono essere propagati alle attività solo al momento della creazione dell'attività. Per i tag con lo stesso nome, viene data la priorità ai tag di processo anziché ai tag delle definizioni di processo. Se il numero totale di tag combinati del processo e della definizione del processo è superiore a 50, il lavoro viene `FAILED` spostato nello stato.

> **Note**
>
> Se il job viene eseguito su risorse Amazon EKS, non devi specificare`propagateTags`.

Tipo: Booleano

Campo obbligatorio: no

Proprietà del nodo

`nodeProperties`

Quando si registra una definizione di processo parallelo multinodo, è necessario specificare un elenco di proprietà del nodo. Queste proprietà del nodo definiscono il numero di nodi da utilizzare nel job, l'indice del nodo principale e i diversi intervalli di nodi da utilizzare. Se il lavoro

viene eseguito su risorse Fargate, non è possibile specificare. `nodeProperties` Utilizza invece `containerProperties`. Di seguito sono indicate le proprietà del nodo consentite in una definizione di processo. Per ulteriori informazioni, consulta Lavori paralleli multinodo.

> **Note**
>
> Se il job viene eseguito su risorse Amazon EKS, non devi specificare`nodeProperties`.

Tipo: oggetto NodeProperties

Campo obbligatorio: no

`mainNode`

Specifica l'indice del nodo per il nodo principale di un processo parallelo a più nodi. Il valore dell'indice del nodo deve essere inferiore al numero di nodi.

Tipo: integer

Campo obbligatorio: sì

`numNodes`

Il numero di nodi associati a un processo parallelo multinodo.

Tipo: integer

Campo obbligatorio: sì

`nodeRangeProperties`

Un elenco di intervalli di nodi e le relative proprietà associate a un processo parallelo multinodo.

> **Note**
>
> Un gruppo di nodi è un gruppo identico di nodi di processi che condividono tutti le stesse proprietà del container. È possibile utilizzare AWS Batch per specificare fino a cinque gruppi di nodi distinti per ogni job.

Tipo: matrice di oggetti NodeRangeProperty

Campo obbligatorio: sì

`targetNodes`

L'intervallo di nodi, utilizzando i valori di indice del nodo. Un intervallo di `0:3` indica i nodi con i valori di indice compresi tra `0` e `3`. Se il valore dell'intervallo iniziale viene omesso (`:n`), viene utilizzato 0 per iniziare l'intervallo. Se il valore di chiusura dell'intervallo (`n:`) viene omesso, viene utilizzato l'indice di nodo più alto possibile per chiudere l'intervallo. Gli intervalli di nodo cumulativi devono comprendere tutti i nodi (`0:n`). È possibile annidare intervalli di nodi, ad esempio `0:10` e`4:5`. In questo caso, le proprietà dell'`4:5`intervallo hanno la precedenza sulle `0:10` proprietà.

-Tipo: stringa

Campo obbligatorio: no

`container`

I dettagli del container per l'intervallo di nodo. Per ulteriori informazioni, consulta Proprietà del contenitore.

Tipo: oggetto ContainerProperties

Campo obbligatorio: no

Strategia per nuovi tentativi

`retryStrategy`

Quando si registra una definizione di processo, è possibile specificare una strategia per il numero di tentativi da utilizzare per i processi non riusciti inviati con questa definizione di processo. Qualsiasi strategia di nuovo tentativo specificata durante un'SubmitJoboperazione ha la precedenza sulla strategia di nuovo tentativo definita qui. Per impostazione predefinita, ogni processo viene tentato una sola volta. Se si specifica più di un tentativo, il processo viene ritentato se fallisce. Esempi di tentativo fallito includono il processo che restituisce un codice di uscita diverso da zero o l'istanza del contenitore viene terminata. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

Tipo: oggetto RetryStrategy

Campo obbligatorio: no

`attempts`

Il numero di volte per cui spostare un processo nello stato `RUNNABLE`. Puoi specificare da 1 a 10 tentativi. Se il valore di `attempts` è maggiore di uno, in caso di errore il processo viene ritentato il numero di volte specificato, fino a quando viene spostato in `RUNNABLE`.

```
"attempts": integer
```

Tipo: integer

Campo obbligatorio: no

`evaluateOnExit`

Matrice di un massimo di 5 oggetti che specificano le condizioni in base alle quali il processo viene riprovato o fallito. Se viene specificato questo parametro, è necessario specificare anche il parametro `attempts`. Se `evaluateOnExit` viene specificato ma nessuna delle voci corrisponde, il processo viene ritentato.

```
"evaluateOnExit": [
   {
      "action": "string",
      "onExitCode": "string",
      "onReason": "string",
      "onStatusReason": "string"
   }
]
```

Tipo: matrice di oggetti EvaluateOnExit

Campo obbligatorio: no

`action`

Specifica l'azione da eseguire se vengono soddisfatte tutte le condizioni specificate (`onStatusReason`, `onReason` e `onExitCode`). I valori non fanno distinzione tra maiuscole e minuscole.

Tipo: stringa

Campo obbligatorio: sì

Valori validi: `RETRY` | `EXIT`

`onExitCode`

Contiene uno schema a globo da confrontare con la rappresentazione decimale di `ExitCode` ciò che viene restituito per un lavoro. Il modello può contenere fino a 512 caratteri. Può contenere solo numeri. Non può contenere lettere o caratteri speciali. Facoltativamente può terminare con un asterisco (*) in modo che solo l'inizio della stringa debba essere una corrispondenza esatta.

- Tipo: stringa

Campo obbligatorio: no

`onReason`

Contiene uno schema a globo da confrontare con `Reason` quello restituito per un lavoro. Il modello può contenere fino a 512 caratteri. Può contenere lettere, numeri, punti (.), due punti (:) e spazi bianchi (spazi, tabulazioni). Facoltativamente può terminare con un asterisco (*) in modo che solo l'inizio della stringa debba essere una corrispondenza esatta.

- Tipo: stringa

Campo obbligatorio: no

`onStatusReason`

Contiene uno schema a globo da confrontare con `StatusReason` quello restituito per un lavoro. Il modello può contenere fino a 512 caratteri. Può contenere lettere, numeri, punti (.), due punti (:) e spazi bianchi (spazi, tabulazioni). Facoltativamente può terminare con un asterisco (*) in modo che solo l'inizio della stringa debba essere una corrispondenza esatta.

- Tipo: stringa

Campo obbligatorio: no

Priorità di pianificazione

`schedulingPriority`

La priorità di pianificazione per i lavori inviati con questa definizione di processo. Ciò riguarda solo i lavori in coda di lavoro con una politica di equa ripartizione. I processi con una priorità di pianificazione più alta vengono pianificati prima dei lavori con una priorità di pianificazione inferiore.

Il valore minimo supportato è 0 e il valore massimo supportato è 9999.

Tipo: integer

Campo obbligatorio: no

Tag

`tags`

Tag di coppia chiave-valore da associare alla definizione del processo. Per ulteriori informazioni, consulta Tagga le tue AWS Batch risorse.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Timeout

`timeout`

È possibile configurare una durata di timeout per i processi in modo che, se un lavoro dura più a lungo, lo AWS Batch interrompa. Per ulteriori informazioni, consulta Job timeout. Se un lavoro viene interrotto a causa di un timeout, non viene ritentato. Qualsiasi configurazione di timeout specificata durante un'SubmitJoboperazione ha la precedenza sulla configurazione di timeout definita qui. Per ulteriori informazioni, consulta Job timeout.

Tipo: oggetto JobTimeout

Campo obbligatorio: no

`attemptDurationSeconds`

La durata in secondi (misurata in base al `startedAt` timestamp del tentativo di lavoro) dopo la fine dei lavori non completati. AWS Batch Il valore minimo per il timeout è 60 secondi.

Per i processi dell'array, il timeout si applica ai processi figlio, non al processo dell'array padre.

Per processi paralleli multinodo (MNP), il timeout si applica all'intero processo, non ai singoli nodi.

Tipo: integer

Campo obbligatorio: no

Crea definizioni di lavoro utilizzando EcsProperties

Utilizzando AWS Batch Job Definitions `EcsProperties`, è possibile modellare hardware, sensori, ambienti 3D e altre simulazioni in contenitori separati. È possibile utilizzare questa funzionalità per organizzare in modo logico i componenti del carico di lavoro e separarli dall'applicazione principale. Questa funzionalità può essere utilizzata con AWS Batch Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS) e. AWS Fargate

`ContainerProperties`rispetto alle definizioni delle mansioni **`EcsProperties`**

È possibile scegliere di utilizzare `ContainerProperties`o utilizzare le `EcsProperties`definizioni in base al caso d'uso. A un livello elevato, l'esecuzione di AWS Batch job with `EcsProperties` è simile all'esecuzione di job con a. `ContainerProperties`

La struttura di definizione dei processi esistente, in uso`ContainerProperties`, rimane supportata. Se attualmente disponi di flussi di lavoro che utilizzano questa struttura, puoi continuare a eseguirli.

La differenza principale è che è stato aggiunto un nuovo oggetto alla definizione del processo per contenere le definizioni `EcsProperties` basate.

Ad esempio, una definizione di lavoro che viene utilizzata `ContainerProperties` su Amazon ECS e Fargate ha la seguente struttura:

```
{
   "containerProperties": {
      ...
      "image": "my_ecr_image1",
      ...
  },
...
}
```

Una definizione di lavoro che viene utilizzata `EcsProperties` su Amazon ECS e Fargate ha la seguente struttura:

```
{
  "ecsProperties": {
    "taskProperties": [{
      "containers": [
        {
          ...
          "image": "my_ecr_image1",
          ...
        },
        {
          ...
          "image": "my_ecr_image2",
          ...
        },
```

Modifiche generali al AWS Batch APIs

Di seguito vengono illustrate ulteriormente alcune delle principali differenze nell'utilizzo dei tipi di dati `ContainerProperties` e delle `EcsProperties` API:

- Molti dei parametri utilizzati all'interno vengono `ContainerProperties` visualizzati all'interno`TaskContainerProperties`. Alcuni esempi includono`command`,`image`,`privileged`,`secrets`, e`users`. Si possono trovare tutti all'interno TaskContainerProperties.
- Alcuni `TaskContainerProperties` parametri non hanno equivalenti funzionali nella struttura precedente. Alcuni esempi includono,`dependsOn`,`essential`, `nameipcMode`, `epidMode`. Per ulteriori informazioni, consultare EcsTaskDetails e TaskContainerProperties.

  Inoltre, alcuni `ContainerProperties` parametri non hanno equivalenti o applicazioni nella `EcsProperties` struttura. In `taskProperties`, `container` è stato sostituito con in `containers` modo che il nuovo oggetto possa accettare fino a dieci elementi. Per ulteriori informazioni, vedi:containerProperties e:containersRegisterJobDefinition. EcsTaskProperties
- `taskRoleArn`è funzionalmente equivalente `jobRoleArn` a. Per ulteriori informazioni, vedere EcsTaskProperties: taskRoleArn e ContainerProperties: jobRoleArn.
- È possibile includere da uno (1) a dieci (10) contenitori nella `EcsProperties` struttura. Per ulteriori informazioni, vedi:containersEcsTaskProperties.
- Gli oggetti `taskProperties` e InstanceTypes sono matrici, ma attualmente accettano solo un elemento. Ad esempio, :taskProperties e:instanceTypesEcsProperties. NodeRangeProperty

Definizioni di processi multi-container per Amazon ECS

Per adattarsi alla struttura multi-contenitore per Amazon ECS, alcuni tipi di dati delle API sono diversi. Ad esempio,

- `ecsProperties`è lo stesso livello della definizione di `containerProperties` contenitore singolo. Per ulteriori informazioni, consulta EcsProperties nella guida di riferimento delle API AWS Batch .
- `taskProperties`contiene le proprietà definite per il task Amazon ECS. Per ulteriori informazioni, consulta EcsProperties nella guida di riferimento delle API AWS Batch .
- `containers`include informazioni simili a quelle contenute `containerProperties` nella definizione di contenitore singolo. La differenza principale è che `containers` consente di definire fino a dieci contenitori. Per ulteriori informazioni, consulta ECSTaskProperties:Containers nella API Reference Guide.AWS Batch
- `essential`il parametro indica in che modo il contenitore influisce sul lavoro. Tutti i contenitori essenziali devono essere completati correttamente (uscire come 0) per far avanzare il lavoro. Se un contenitore contrassegnato come essenziale fallisce (esce come diverso da 0), il processo fallisce.

  Il valore predefinito è `true` e almeno un contenitore deve essere contrassegnato come. `essential` Per ulteriori informazioni, consulta `essential` nella guida di riferimento delle API AWS Batch .
- Con il `dependsOn`parametro, è possibile definire un elenco di dipendenze del contenitore. Per ulteriori informazioni, consulta `dependsOn` nella guida di riferimento delle API AWS Batch .

> **Note**
>
> La complessità dell'`dependsOn`elenco e il relativo runtime del contenitore possono influire sull'ora di inizio del processo. Se l'esecuzione delle dipendenze impiega molto tempo, il processo rimarrà in uno `STARTING` stato fino al completamento.

Per ulteriori informazioni sulla struttura `ecsProperties` and, consulta la sintassi della RegisterJobDefinitionrichiesta per ECSProperties.

Definizioni di processi multi-container per Amazon EKS

Per adattarsi alla struttura multi-contenitore per Amazon EKS, alcuni tipi di dati delle API sono diversi. Ad esempio,

- `name`è un identificatore univoco per il contenitore. Questo oggetto non è richiesto per un singolo contenitore, ma è necessario quando si definiscono più contenitori in un contenitore. Quando `name` non è definito per singoli contenitori, viene applicato il nome predefinito`default`,.
- `initContainers`sono definiti all'interno del tipo di eksPodPropertiesdati. Vengono eseguiti prima dei contenitori delle applicazioni, vengono sempre eseguiti fino al completamento e devono essere completati correttamente prima dell'avvio del contenitore successivo.

  Questi contenitori sono registrati con l'agente Amazon EKS Connector e mantengono le informazioni di registrazione nell'archivio dati di backend di Amazon Elastic Kubernetes Service. L'`initContainers`oggetto può accettare fino a dieci (10) elementi. Per ulteriori informazioni, vedete Init Containers nella Kubernetesdocumentazione.

  > **Note**
  >
  > L'`initContainers`oggetto può influire sull'ora di inizio del lavoro. Se l'esecuzione `initContainers` richiede molto tempo, il processo rimarrà in uno `STARTING` stato fino al completamento.

- `shareProcessNamespace`indica se i contenitori nel contenitore possono condividere lo stesso spazio dei nomi del processo. I valori predefiniti sono. `false` Impostando questa impostazione `true` per consentire ai contenitori di visualizzare e segnalare i processi in altri contenitori che si trovano nello stesso contenitore.
- Ogni contenitore ha un'importanza. Affinché il lavoro abbia esito positivo, tutti i contenitori devono essere completati correttamente (uscire come 0). Se un contenitore fallisce (esce con un valore diverso da 0), il processo fallisce.

Per ulteriori informazioni sulla struttura `eksProperties` and, vedere la sintassi della RegisterJobDefinitionrichiesta per EksProperties.

Riferimento: scenari AWS Batch di lavoro che utilizzano EcsProperties

Per illustrare come le definizioni di AWS Batch job utilizzate `EcsProperties` possono essere strutturate in base alle esigenze dell'utente, questo argomento presenta i seguenti payload.

RegisterJobDefinition È possibile copiare questi esempi in un file, personalizzarli in base alle proprie esigenze e quindi utilizzare il AWS Command Line Interface (AWS CLI) per chiamare. `RegisterJobDefinition`

AWS Batch lavoro per Amazon ECS su Amazon EC2

Di seguito è riportato un esempio di AWS Batch lavoro per Amazon Elastic Container Service su Amazon Elastic Compute Cloud:

```
{
    "jobDefinitionName": "multicontainer-ecs-ec2",
    "type": "container",
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
              {
                "name": "c1",
                "essential": false,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              },
              {
                "name": "c2",
                "essential": false,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
```

```
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              },
              {
                "name": "c3",
                "essential": true,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "firelensConfiguration": {
                  "type": "fluentbit",
                  "options": {
                    "enable-ecs-log-metadata": "true"
                  }
                 },
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "6"
                  },
                  {
                    "type": "MEMORY",
                    "value": "12288"
                  }
                ]
              }
            ]
          }
        ]
  }
}
```

AWS Batch lavoro per Amazon ECS su Fargate

Di seguito è riportato un esempio di AWS Batch lavoro per Amazon Elastic Container Service su AWS Fargate:

```
{
    "jobDefinitionName": "multicontainer-ecs-fargate",
    "type": "container",
    "platformCapabilities": [
        "FARGATE"
    ],
    "ecsProperties": {
        "taskProperties": [
          {
            "containers": [
              {
                "name": "c1",
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              },
              {
                "name": "c2",
                "essential": true,
                "command": [
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
```

```
                    "type": "VCPU",
                    "value": "6"
                  },
                  {
                    "type": "MEMORY",
                    "value": "12288"
                  }
                ]
              }
            ],
            "executionRoleArn": "arn:aws:iam::1112223333:role/ecsTaskExecutionRole"
          }
        ]
  }
}
```

AWS Batch lavoro per Amazon EKS

Di seguito è riportato un esempio di AWS Batch job per Amazon Elastic Kubernetes Service:

```
{
  "jobDefinitionName": "multicontainer-eks",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "shareProcessNamespace": true,
      "initContainers": [
        {
          "name": "init-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo"
          ],
          "args": [
            "hello world"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
```

```
        {
          "name": "init-container-2",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo",
            "my second init container"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ],
      "containers": [
        {
          "name": "c1",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "echo world"
         ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        },
        {
          "name": "sleep-container",
          "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
          "command": [
            "sleep",
            "20"
          ],
          "resources": {
            "requests": {
              "cpu": "1",
              "memory": "512Mi"
            }
          }
        }
      ]
```

```
    }
  }
}
```

AWS Batch Job MNP con più contenitori per nodo

Di seguito è riportato un esempio di AWS Batch job multi-node parallel (MNP) con più contenitori per nodo:

```
{
  "jobDefinitionName": "multicontainer-mnp",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 6,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:5",
        "ecsProperties": {
          "taskProperties": [
            {
              "containers": [
                {
                  "name": "range05-c1",
                  "command": [
                    "echo",
                    "hello world"
                  ],
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                  "resourceRequirements": [
                    {
                      "type": "VCPU",
                      "value": "2"
                    },
                    {
                      "type": "MEMORY",
                      "value": "4096"
                    }
                  ]
                },
                {
                  "name": "range05-c2",
                  "command": [
```

```
                  "echo",
                  "hello world"
                ],
                "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
                "resourceRequirements": [
                  {
                    "type": "VCPU",
                    "value": "2"
                  },
                  {
                    "type": "MEMORY",
                    "value": "4096"
                  }
                ]
              }
            ]
          }
        ]
      }
    ]
  }
}
```

Usa il driver di registro awslogs

Per impostazione predefinita, AWS Batch consente al driver di `awslogs` registro di inviare informazioni di registro a CloudWatch Logs. È possibile utilizzare questa funzionalità per visualizzare diversi registri dei contenitori in un'unica comoda posizione ed evitare che i registri dei contenitori occupino spazio su disco sulle istanze del contenitore. Questo argomento consente di configurare il driver di `awslogs` registro nelle definizioni dei processi.

> **Note**
>
> Nella AWS Batch console, è possibile configurare il driver di `awslogs` registro nella sezione Configurazione della registrazione quando si crea una definizione di processo.

> **Note**
>
> Il tipo di informazioni registrate dai contenitori del processo dipende principalmente dal loro `ENTRYPOINT` comando. Per impostazione predefinita, i log acquisiti mostrano l'output del comando che normalmente si vede in un terminale interattivo se il contenitore viene eseguito localmente, ovvero i flussi `STDOUT` and `STDERR` I/O . 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 del contenitore a Logs, vedi. CloudWatch Utilizzo dei CloudWatch registri con AWS Batch Per ulteriori informazioni sui CloudWatch log, consulta Monitoring Log Files e CloudWatch Logs quote nella Amazon CloudWatch Logs User Guide.

awslogs registra le opzioni del driver nel tipo di dati AWS Batch JobDefiniton

Il driver di `awslogs` registro supporta le seguenti opzioni nelle AWS Batch definizioni dei processi. Per ulteriori informazioni, consulta CloudWatch Logs logging driver nella documentazione Docker.

`awslogs-region`

Campo obbligatorio: no

Specificate la regione in cui il driver di `awslogs` registro deve inviare i log Docker. Per impostazione predefinita, la regione utilizzata è la stessa di quella per il lavoro. Puoi scegliere di inviare tutti i log dei lavori in diverse regioni a una singola regione in CloudWatch Logs. In questo modo, saranno visibili tutti da un'unica posizione. In alternativa, puoi separarli per regione per un approccio più granulare. Tuttavia, quando scegliete questa opzione, assicuratevi che i gruppi di log specificati esistano nella regione specificata.

`awslogs-group`

Obbligatorio: facoltativo

Con l'`awslogs-group`opzione, è possibile specificare il gruppo di log a cui il driver di `awslogs` log invia i propri flussi di log. Se questo non è specificato, `aws/batch/job` viene utilizzato.

`awslogs-stream-prefix`

Obbligatorio: facoltativo

Con l'`awslogs-stream-prefix`opzione, puoi associare un flusso di log al prefisso specificato e all'ID attività Amazon ECS del AWS Batch lavoro a cui appartiene il contenitore. Se specifichi un prefisso con questa opzione, il flusso di log assume il formato seguente:

```
prefix-name/default/ecs-task-id
```

`awslogs-datetime-format`

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea nel formato `strftime` Python. Un messaggio di log è costituito da una riga che corrisponde allo schema e da tutte le righe successive che non corrispondono allo schema. In questo modo la riga associata è il delimitatore tra i messaggi di log.

Un esempio di un caso d'uso per l'utilizzo di questo formato è per l'analisi di output, ad esempio uno dump dello stack, che potrebbe altrimenti essere registrato in più voci. Il modello corretto consente di acquisirlo in una sola voce.

Per ulteriori informazioni, consulta awslogs-datetime-format.

Questa opzione è sempre prioritaria nel caso in cui siano configurati sia `awslogs-datetime-format` che `awslogs-multiline-pattern`.

> **Note**
>
> Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

`awslogs-multiline-pattern`

Campo obbligatorio: no

Questa opzione definisce un modello di inizio multilinea utilizzando un'espressione regolare. Un messaggio di registro è costituito da una riga che corrisponde allo schema e da tutte le righe

successive che non corrispondono allo schema. Pertanto, la riga corrispondente è il delimitatore tra i messaggi di registro.

Per ulteriori informazioni, consulta la documentazione di awslogs-multiline-patternDocker.

Questa opzione viene ignorata se anche `awslogs-datetime-format` è configurato.

> **Note**
>
> Il logging multilinea esegue un'espressione regolare per l'analisi e il confronto di tutti i messaggi di log. L'operazione potrebbe avere ripercussioni negative sulle prestazioni del logging.

`awslogs-create-group`

Campo obbligatorio: no

Specifica se desideri che il gruppo di log venga creato automaticamente. Se questa opzione non è specificata, viene impostata in modo predefinito su `false`.

> **Warning**
>
> Questa opzione non è consigliata. Si consiglia di creare il gruppo di log in anticipo utilizzando l'azione CloudWatch Logs CreateLogGroupAPI ogni volta che ogni job tenta di creare il gruppo di log, aumentando la probabilità che il processo non riesca.

> **Note**
>
> La policy IAM per il tuo ruolo di esecuzione deve includere l'`logs:CreateLogGroup`autorizzazione prima di tentare di `awslogs-create-group` utilizzarla.

Specificate una configurazione di registro nella definizione del lavoro

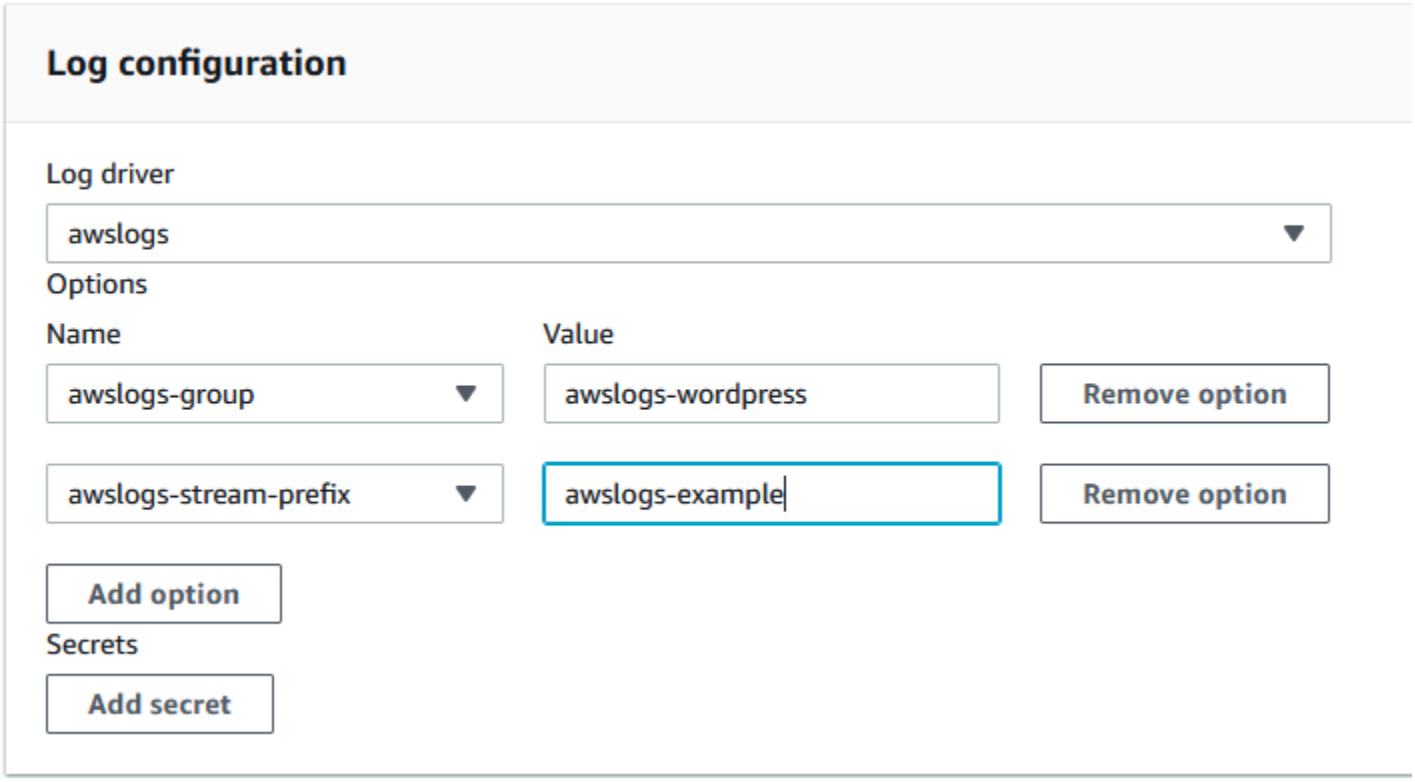
Per impostazione predefinita, AWS Batch abilita il driver di `awslogs` registro. Questa sezione descrive come personalizzare la configurazione del `awslogs` registro per un lavoro. Per ulteriori informazioni, consulta Creare una definizione di processo a nodo singolo .

I seguenti frammenti JSON di configurazione del registro hanno un `logConfiguration` oggetto specificato per ogni processo. Uno è per un WordPress processo che invia i log a un gruppo di log chiamato `awslogs-wordpress` e un altro è per un contenitore 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.

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-wordpress",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

```
"logConfiguration": {
    "logDriver": "awslogs",
    "options": {
        "awslogs-group": "awslogs-mysql",
        "awslogs-stream-prefix": "awslogs-example"
    }
}
```

Nella AWS Batch console, la configurazione del registro per la definizione del `wordpress` processo viene specificata come mostrato nell'immagine seguente.



Dopo aver registrato una definizione di attività con il driver di `awslogs` registro in una configurazione del registro delle definizioni del processo, è possibile inviare un lavoro con tale definizione di processo per iniziare a inviare i log ai CloudWatch registri. Per ulteriori informazioni, consulta Tutorial: invia un lavoro.

Specificare dati sensibili

Con AWS Batch, puoi inserire dati sensibili nei tuoi lavori archiviandoli in Gestione dei segreti AWS segreti o nei parametri di AWS Systems Manager Parameter Store e quindi facendo riferimento ad essi nella definizione del processo.

I segreti possono essere esposti a un job nei seguenti modi:

- Per inserire dati sensibili nei contenitori come variabili di ambiente, utilizzate il parametro di definizione del `secrets` processo.
- Per fare riferimento a informazioni riservate nella configurazione di registro di un lavoro, utilizzate il parametro di definizione del `secretOptions` lavoro.

Argomenti

- Specificare i dati sensibili con Secrets Manager
- Specificare i dati sensibili con Systems Manager Parameter Store

Specificare i dati sensibili con Secrets Manager

Con AWS Batch, puoi inserire dati sensibili nei tuoi lavori archiviandoli in Gestione dei segreti AWS modo riservato e quindi facendone riferimento nella definizione del lavoro. I dati sensibili archiviati nei segreti di Secrets Manager possono essere esposti a un job come variabili di ambiente o come parte della configurazione del registro.

Quando si inserisce un segreto come variabile di ambiente, è possibile specificare una chiave JSON o una versione di un segreto da inserire. Questo processo consente di controllare i dati sensibili esposti al lavoro. Per ulteriori informazioni sul controllo delle versioni dei segreti, consulta i termini e concetti chiave per Gestione dei segreti AWS nella Guida per l'utente di Gestione dei segreti AWS .

Considerazioni quando si specificano dati sensibili utilizzando Secrets Manager

Quando si utilizza Secrets Manager per specificare dati sensibili per i lavori, è necessario considerare quanto segue.

- Per inserire un segreto utilizzando una chiave JSON specifica o una versione di un segreto, nell'istanza del contenitore nel tuo ambiente di calcolo deve essere installata la versione 1.37.0 o successiva dell'agente contenitore Amazon ECS. 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 Updating the Amazon ECS Container Agent nella Amazon Elastic Container Service Developer Guide.

  Per inserire l'intero contenuto di un segreto come variabile di ambiente o per inserire un segreto in una configurazione di registro, l'istanza del contenitore deve avere la versione 1.23.0 o successiva dell'agente contenitore.
- 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.
- Quando utilizzi una definizione di lavoro che fa riferimento ai segreti di Secrets Manager per recuperare dati sensibili per i tuoi lavori, se utilizzi anche endpoint VPC di interfaccia, devi creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di Secrets Manager con endpoint VPC nella Guida per l'utente di Gestione dei segreti AWS .

- I dati sensibili vengono inseriti nel lavoro all'avvio iniziale del lavoro. Se il segreto viene successivamente aggiornato o ruotato, il lavoro non riceve automaticamente il valore aggiornato. È necessario avviare un nuovo lavoro per forzare il servizio a lanciare un nuovo lavoro con il valore segreto aggiornato.

Autorizzazioni IAM richieste per i segreti AWS Batch

Per utilizzare questa funzionalità, è necessario disporre del ruolo di esecuzione e farvi riferimento nella definizione del processo. Ciò consente all'agente del container di recuperare le risorse di Secrets Manager necessarie. Per ulteriori informazioni, consulta AWS Batch Ruolo di esecuzione IAM.

Per fornire l'accesso ai segreti di Secrets Manager che crei, aggiungi manualmente le seguenti autorizzazioni come policy in linea al ruolo di esecuzione. Per informazioni, consulta Aggiunta e rimozione delle policy IAM nella Guida per l'utente di IAM.

- `secretsmanager:GetSecretValue`: obbligatorio se si fa riferimento a un segreto di Secrets Manager.
- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave KMS personalizzata e non quella di default. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa.

La policy inline dell'esempio seguente aggiunge le autorizzazioni necessarie:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-
east-2:777777777777:secret:<secret_name>",
                "arn:aws:kms:us-east-2:777777777777:key/<key_id>"
            ]
```

```
        }
    ]
}
```

Inserimento di dati sensibili come una variabile di ambiente

È possibile specificare quanto segue all'interno della definizione del lavoro:

- L'`secrets`oggetto contenente il nome della variabile di ambiente da impostare nel lavoro
- L'Amazon Resource Name (ARN) del segreto di Secrets Manager
- Parametri aggiuntivi che contengono i dati sensibili da presentare al lavoro

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 opzionali. Tuttavia, se non li utilizzate, dovete 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 per AWS Secrets Manager nella Guida per l'Gestione dei segreti AWS utente.

`version-id`

Specifica l'identificatore univoco della versione del segreto che intendi utilizzare. Se viene specificato un ID versione, non è possibile specificare un'etichetta di gestione temporanea della versione. Se non viene specificato alcun ID versione, il comportamento predefinito consiste nel recuperare il segreto con l'etichetta `AWSCURRENT` di gestione temporanea.

IDs Le versioni vengono utilizzate per tenere traccia delle diverse versioni di un segreto quando vengono aggiornate o ruotate. Ogni versione di un segreto ha un ID. Per ulteriori informazioni, consulta Termini e concetti chiave per AWS Secrets Manager nella Guida per l'Gestione dei segreti AWS utente.

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.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
    }]
  }]
}
```

Example fare riferimento a una chiave specifica all'interno di un segreto

Di seguito viene illustrato un esempio di output di un get-secret-value comando > che visualizza il contenuto di un segreto insieme all'etichetta di staging della versione e all'ID di versione ad esso associati.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
```

```
    "VersionId": "871d9eca-18aa-46a9-8785-981dd39ab30c",
    "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.

```
{
  "containerProperties": [{
    "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 esempio di output di 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-46a9-8785-981dd39ab30c": [
            "AWSCURRENT"
        ],
        "9d4cb84b-ad69-40c0-a0ab-cead36b967e8": [
            "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.

```
{
  "containerProperties": [{
    "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.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

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.

```
{
  "containerProperties": [{
    "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.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-
AbCdEf:username1::9d4cb84b-ad69-40c0-a0ab-cead36b967e8"
    }]
  }]
}
```

Inietta dati sensibili in una configurazione di registro

Quando si specifica a `logConfiguration` all'interno della definizione del processo, è possibile specificare `secretOptions` il nome dell'opzione del driver di registro da impostare nel contenitore e l'ARN completo del segreto di Secrets Manager contenente i dati sensibili da presentare al contenitore.

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.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "splunk",
      "options": {
        "splunk-url": "https://cloud.splunk.com:8080"
      },
      "secretOptions": [{
        "name": "splunk-token",
        "valueFrom": "arn:aws:secretsmanager:region:aws_account_id:secret:secret_name-
AbCdEf"
      }]
    }]
  }]
}
```

Crea un segreto Gestione dei segreti AWS

Puoi utilizzare la console Secrets Manager per creare un segreto per i dati sensibili. Per ulteriori informazioni, consulta Creazione di un segreto di base nella Guida per l'utente di Gestione dei segreti AWS .

Come creare un segreto di base

Utilizza Secrets Manager per creare un segreto per i dati sensibili.

1. Apri la console Secrets Manager all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. In Select secret type (Seleziona tipo di segreto), scegliere Other type of secrets (Altro tipo di segreti).
4. Specifica i dettagli del tuo segreto personalizzato come Key (Chiave) e Value (Valore). Ad esempio, puoi specificare una chiave `UserName` e fornire il nome utente appropriato come valore. Aggiungi una seconda chiave con il nome di `Password` e il testo della password come valore. È inoltre possibile aggiungere voci per il nome di un database, l'indirizzo del server o la porta TCP. Puoi aggiungere tutte le coppie di cui hai bisogno per archiviare le informazioni necessarie.

   In alternativa, puoi scegliere la scheda Plaintext (Testo normale) e immettere il valore del segreto come desideri.
5. Scegliete la chiave di AWS KMS crittografia che desiderate utilizzare per crittografare il testo protetto nel segreto. Se non scegli una chiave, Secrets Manager verifica se esiste una chiave di default per l'account e, nel caso, la utilizza. Se una chiave di default non esiste, Secrets Manager ne crea automaticamente una. Puoi anche scegliere Aggiungi nuova chiave per creare un nuova chiave KMS personalizzata specifica per questo segreto. Per creare la tua chiave KMS, devi disporre delle autorizzazioni per creare le chiavi KMS nel tuo account.
6. Scegli Next (Successivo).
7. Per Secret name (Nome segreto), digitare un percorso e nome opzionali, come **`production/MyAwesomeAppSecret`** o **`development/TestSecret`** e selezionare Next (Successivo). Opzionalmente, è possibile aggiungere una descrizione che aiuti a ricordare lo scopo di questo segreto in futuro.

   Il nome del segreto deve essere costituito da lettere ASCII, cifre o uno dei seguenti caratteri:/_ **+=.@-**

8. (Opzionale) A questo punto, puoi configurare la rotazione per il tuo segreto. Per questa procedura, lasciare l'impostazione Disable automatic rotation (Disabilita rotazione automatica) e selezionare Next (Successivo).

   Per informazioni su come configurare la rotazione su segreti nuovi o esistenti, consulta Rotating Your Gestione dei segreti AWS Secrets.
9. Verifica le tue impostazioni e poi scegli Archivia segreto per salvare tutti i dati immessi come nuovo segreto in Secrets Manager.

Specificare i dati sensibili con Systems Manager Parameter Store

Con AWS Batch, puoi inserire dati sensibili nei tuoi contenitori memorizzando i dati sensibili nei parametri di AWS Systems Manager Parameter Store e quindi facendo riferimento ad essi nella definizione del contenitore.

Argomenti

- Considerazioni relative alla specificazione di dati sensibili utilizzando Systems Manager Parameter Store
- Autorizzazioni IAM richieste per i segreti AWS Batch
- Inietta dati sensibili come variabile di ambiente
- Inietta dati sensibili in una configurazione di registro
- Creare un parametro Parameter Store AWS Systems Manager

Considerazioni relative alla specificazione di dati sensibili utilizzando Systems Manager Parameter Store

Quando specifichi i dati sensibili per i container utilizzando i parametri dell'archivio parametri di Systems Manager, è necessario considerare quanto segue.

- Questa funzionalità richiede che l'istanza del contenitore disponga della versione 1.23.0 o successiva dell'agente contenitore. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni sulla verifica della versione dell'agente e sull'aggiornamento alla versione più recente, consulta Updating the Amazon ECS Container Agent nella Amazon Elastic Container Service Developer Guide.
- I dati sensibili vengono iniettati nel contenitore per il tuo lavoro quando il contenitore viene inizialmente avviato. Se il segreto o il parametro dell'archivio parametri viene in seguito aggiornato

o ruotato, il container non riceverà automaticamente il valore aggiornato. È necessario avviare un nuovo lavoro per forzare il lancio di un nuovo lavoro con segreti aggiornati.

Autorizzazioni IAM richieste per i segreti AWS Batch

Per utilizzare questa funzionalità, è necessario disporre del ruolo di esecuzione e farvi riferimento nella definizione del processo. Ciò consente all'agente container Amazon ECS di reperire le AWS Systems Manager risorse necessarie. Per ulteriori informazioni, consulta AWS Batch Ruolo di esecuzione IAM.

Per fornire l'accesso ai parametri di AWS Systems Manager Parameter Store che crei, aggiungi manualmente le seguenti autorizzazioni come policy in linea al ruolo di esecuzione. Per informazioni, consulta Aggiunta e rimozione delle policy IAM nella Guida per l'utente di IAM.

- `ssm:GetParameters`: obbligatorio se fai riferimento a un parametro dell'archivio parametri di Systems Manager in una definizione di attività.
- `secretsmanager:GetSecretValue`: obbligatorio se fai riferimento direttamente a un segreto di Secrets Manager o se il parametro dell'archivio parametri di Systems Manager fa riferimento a un segreto di Secrets Manager in una definizione di attività.
- `kms:Decrypt`: obbligatorio solo se il segreto utilizza una chiave KMS personalizzata e non quella di default. L'ARN per la chiave personalizzata deve essere aggiunto come risorsa.

La policy inline dell'esempio seguente aggiunge le autorizzazioni necessarie:

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ssm:GetParameters",
                "secretsmanager:GetSecretValue",
                "kms:Decrypt"
            ],
            "Resource": [
                "arn:aws:ssm:us-east-2:999999999999:parameter/<parameter_name>",
```

```
                "arn:aws:secretsmanager:us-
east-2:999999999999:secret:<secret_name>",
                "arn:aws:kms:us-east-2:999999999999:key/<key_id>"
            ]
        }
    ]
}
```

Inietta dati sensibili come variabile di ambiente

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.

Di seguito è riportato un frammento di una definizione di attività che mostra il formato quando si fa riferimento a un parametro Systems Manager Parameter Store. Se il parametro Systems Manager Parameter Store esiste nella stessa regione dell'attività che si sta avviando, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro esiste in una Regione diversa, deve essere specificato l'ARN completo.

```
{
  "containerProperties": [{
    "secrets": [{
      "name": "environment_variable_name",
      "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
    }]
  }]
}
```

Inietta dati sensibili in una configurazione di registro

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 Systems Manager Parameter Store esiste nella stessa regione dell'attività che si sta avviando, è possibile utilizzare l'ARN completo o il nome del parametro. Se il parametro esiste in una Regione diversa, deve essere specificato 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 Systems Manager Parameter Store.

```
{
  "containerProperties": [{
    "logConfiguration": [{
      "logDriver": "fluentd",
      "options": {
        "tag": "fluentd demo"
      },
      "secretOptions": [{
        "name": "fluentd-address",
        "valueFrom": "arn:aws:ssm:region:aws_account_id:parameter/parameter_name"
      }]
    }]
  }]
}
```

Creare un parametro Parameter Store AWS Systems Manager

È possibile utilizzare la AWS Systems Manager console per creare un parametro Systems Manager Parameter Store per i dati sensibili. Per ulteriori informazioni consulta Spiegazione passo per passo: creazione e utilizzo di un parametro in un comando (console) nella Guida per l'utente di AWS Systems Manager .

Per creare un parametro dell'Archivio parametri

1. Aprire la AWS Systems Manager console all'indirizzo https://console.aws.amazon.com/systems-manager/.
2. Nel pannello di navigazione, scegli Archivio parametri, Crea parametro.
3. In Name (Nome) immetti una gerarchia e un nome di parametro. Ad esempio, digita `test/database_password`.

4. In Description (Descrizione), digita una descrizione opzionale.
5. Per Tipo, scegliete String StringList, o SecureString.

> **Note**
>
> - Se scegli SecureString, viene visualizzato il campo KMS Key ID. Se non specifichi un ID chiave KMS, un ARN della chiave KMS, un nome alias o un ARN alias, il sistema utilizzerà `alias/aws/ssm`. Questa è la chiave KMS predefinita per Systems Manager. Per evitare l'utilizzo di questa chiave, scegli una chiave personalizzata. Per ulteriori informazioni, consulta Utilizzo dei parametri di stringa sicura nella Guida per l'utente di AWS Systems Manager .
> - Quando crei un parametro di stringa sicura nella console utilizzando il parametro `key-id` con un nome alias della chiave KMS personalizzata o un ARN dell'alias, devi specificare il prefisso `alias/` prima dell'alias. Di seguito è riportato un ARN di esempio:
>
>   ```
>   arn:aws:kms:us-east-2:123456789012:alias/MyAliasName
>   ```
>
>   Di seguito è riportato un nome alias di esempio:
>
>   ```
>   alias/MyAliasName
>   ```

6. In Value (Valore), digita un valore. Ad esempio, `MyFirstParameter`. Se hai scelto SecureString, il valore viene mascherato esattamente come lo hai inserito.
7. Scegli Create parameter (Crea parametro).

Autenticazione del registro privato per i lavori

L'autenticazione del registro privato per i lavori Gestione dei segreti AWS consente di archiviare le credenziali in modo sicuro e quindi di farvi riferimento nella definizione del processo. In questo modo è possibile fare riferimento alle immagini dei contenitori presenti in registri privati che non richiedono l'autenticazione nelle AWS definizioni dei processi. Questa funzionalità è supportata dai lavori ospitati su EC2 istanze Amazon e Fargate.

> **Important**
>
> Se la definizione del processo fa riferimento a un'immagine archiviata in Amazon ECR, questo argomento non è pertinente. Per ulteriori informazioni, consulta Utilizzo di immagini Amazon ECR con Amazon ECS nella Guida per l'utente di Amazon Elastic Container Registry.

Per i lavori ospitati su EC2 istanze Amazon, questa funzionalità richiede una versione `1.19.0` o successiva dell'agente container. Tuttavia, ti consigliamo di utilizzare la versione più recente dell'agente container. Per informazioni su come verificare la versione dell'agente e aggiornarla alla versione più recente, consulta Updating the Amazon ECS container agent nella Amazon Elastic Container Service Developer Guide.

Per i lavori ospitati su Fargate, questa funzionalità richiede una versione della piattaforma `1.2.0` o successiva. Per informazioni, consulta le versioni della piattaforma AWS Fargate Linux nella Amazon Elastic Container Service Developer Guide.

All'interno della definizione del container, specifica l'oggetto `repositoryCredentials` con i dettagli del segreto che hai creato. Il segreto a cui fai riferimento può provenire da un account diverso Regione AWS o diverso da quello del lavoro che lo utilizza.

> **Note**
>
> Quando si utilizza l' AWS Batch API o l' AWS SDK, se il segreto esiste nello stesso Regione AWS processo che si sta avviando, è possibile utilizzare l'ARN completo o il nome del segreto. AWS CLI Se il segreto esiste in un altro account, occorre specificare l'ARN completo del segreto. Quando si utilizza Console di gestione AWS, è necessario specificare sempre l'ARN completo del segreto.

Di seguito è riportato un frammento di una definizione di processo che mostra i parametri richiesti:

```
"containerProperties": [
  {
    "image": "private-repo/private-image",
    "repositoryCredentials": {
      "credentialsParameter":
 "arn:aws:secretsmanager:region:123456789012:secret:secret_name"
```

```
    }
  }
]
```

Autorizzazioni IAM obbligatorie per l'autenticazione di registri privati

Il ruolo di esecuzione è necessario per utilizzare questa funzionalità. In questo modo l'agente container può recuperare l'immagine del container. Per ulteriori informazioni, consulta AWS Batch Ruolo di esecuzione IAM.

Per fornire l'accesso ai segreti che crei, aggiungi le seguenti autorizzazioni come policy in linea al ruolo di esecuzione. 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.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Decrypt",
                "secretsmanager:GetSecretValue"
            ],
            "Resource": [
                "arn:aws:secretsmanager:us-
east-1:123456789012:secret:secret_name",
                "arn:aws:kms:us-east-1:123456789012:key/key_id"
            ]
        }
    ]
}
```

Tutorial: crea un segreto per l'autenticazione del registro privato

Completa i seguenti passaggi per creare un segreto per le credenziali del tuo registro privato con Gestione dei segreti AWS.

Crea un segreto di base

1. Apri la Gestione dei segreti AWS console all'indirizzo https://console.aws.amazon.com/secretsmanager/.
2. Scegli Archivia un nuovo segreto.
3. In Select secret type (Seleziona tipo di segreto), scegliere Other type of secrets (Altro tipo di segreti).
4. Selezionare Plaintext (Testo normale) e inserire le credenziali del registro privato utilizzando il formato seguente:

   ```
   {
     "username" : "privateRegistryUsername",
     "password" : "privateRegistryPassword"
   }
   ```

5. Scegli Next (Successivo).
6. Per Secret name (Nome segreto), digita un percorso e un nome facoltativi come **`production/MyAwesomeAppSecret`** o **`development/TestSecret`** e seleziona Next (Successivo). Opzionalmente, è possibile aggiungere una descrizione che aiuti a ricordare lo scopo di questo segreto in futuro.

   Il nome del segreto deve essere costituito da lettere ASCII, cifre o uno dei seguenti caratteri: `/_+=.@-`.
7. (Opzionale) A questo punto, puoi configurare la rotazione per il tuo segreto. Per questa procedura, lasciare l'impostazione Disable automatic rotation (Disabilita rotazione automatica) e selezionare Next (Successivo).

   Per istruzioni su come configurare la rotazione su segreti nuovi o esistenti, vedi Rotating Your Gestione dei segreti AWS Secrets.
8. Verifica le tue impostazioni e poi scegli Store secret (Archivia segreto) per salvare tutti i dati immessi come nuovo segreto in Secrets Manager.

Registra una definizione di lavoro e in Registro privato, attiva l'autenticazione del registro privato. Quindi, in ARN o nome di Gestione dei segreti, inserisci il nome della risorsa Amazon (ARN) del segreto. Per ulteriori informazioni, consulta Autorizzazioni IAM obbligatorie per l'autenticazione di registri privati.

Volumi Amazon EFS

Amazon Elastic File System (Amazon EFS) offre uno storage di file semplice e scalabile da utilizzare con i tuoi AWS Batch lavori. Con Amazon EFS, la capacità di storage è elastica. Si ridimensiona automaticamente man mano che aggiungi e rimuovi file. Le tue applicazioni possono disporre dell'archiviazione di cui hanno bisogno al momento del bisogno.

Puoi utilizzare i file system Amazon EFS AWS Batch per esportare i dati del file system nella tua flotta di istanze di container. In questo modo, i tuoi job hanno accesso allo stesso storage persistente. Tuttavia, devi configurare l'AMI per la tua istanza di container per montare il file system Amazon EFS prima dell'avvio del daemon Docker. Inoltre, le definizioni dei processi devono fare riferimento ai montaggi dei volumi sull'istanza del contenitore per utilizzare il file system. Le seguenti sezioni ti aiutano a iniziare a usare Amazon EFS con AWS Batch.

Considerazioni sui volumi Amazon EFS

Quando usi i volumi Amazon EFS, tieni presente le considerazioni seguenti:

- Per i lavori che utilizzano EC2 risorse, il supporto del file system Amazon EFS è stato aggiunto come anteprima pubblica con la versione AMI ottimizzata per Amazon ECS `20191212` con agente container versione 1.35.0. Tuttavia, il supporto del file system di Amazon EFS è entrato in disponibilità generale con la versione AMI ottimizzata per Amazon ECS `20200319` con agente container versione 1.38.0, che conteneva il punto di accesso Amazon EFS e le funzionalità di autorizzazione IAM. Ti consigliamo di utilizzare una versione AMI ottimizzata per Amazon ECS `20200319` o successiva per sfruttare queste funzionalità. Per ulteriori informazioni, consulta le versioni AMI ottimizzate per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

  > **Note**
  >
  > Se crei la tua AMI, devi utilizzare l'agente container 1.38.0 o successiva, `ecs-init` versione 1.38.0-1 o successiva ed eseguire i seguenti comandi sulla tua istanza Amazon.

EC2 Tutto questo serve per abilitare il plugin di volume Amazon ECS. I comandi dipendono dal fatto che si stia usando Amazon Linux 2 o Amazon Linux come immagine di base.

Amazon Linux 2

```
$ yum install amazon-efs-utils
systemctl enable --now amazon-ecs-volume-plugin
```

Amazon Linux

```
$ yum install amazon-efs-utils
sudo shutdown -r now
```

- Per i lavori che utilizzano le risorse Fargate, è stato aggiunto il supporto del file system Amazon EFS quando si utilizza la versione 1.4.0 o successiva della piattaforma. Per ulteriori informazioni, consulta le versioni della piattaforma AWS Fargate nella Amazon Elastic Container Service Developer Guide.
- Quando si specificano i volumi Amazon EFS nei lavori che utilizzano risorse Fargate, Fargate crea un contenitore supervisore responsabile della gestione del volume Amazon EFS. Il contenitore supervisor utilizza una piccola quantità di memoria del lavoro. Il container supervisor è visibile quando si sottopone a query l'endpoint dei metadati dell'attività versione 4. Per ulteriori informazioni, consulta Endpoint metadati dei processi versione 4 nella Guida per l'utente di Amazon Elastic Container Service per AWS Fargate.

Usa i punti di accesso Amazon EFS

I punti di accesso Amazon EFS sono punti di ingresso specifici dell'applicazione in un file system EFS che consentono di gestire l'accesso delle applicazioni 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 radice diversa per il file system in modo che i client possano accedere solo ai dati nella directory specificata o nelle sue 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 del punto di accesso nella definizione del AWS Batch processo, la directory principale deve essere omessa o impostata su `/` Ciò impone il percorso impostato sul punto di accesso EFS.

È possibile utilizzare un ruolo AWS Batch Job IAM per far sì che applicazioni specifiche utilizzino un punto di accesso specifico. Combinando le policy IAM con i punti di accesso, puoi fornire facilmente accesso sicuro a set di dati specifici per le applicazioni. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, consulta Ruoli IAM per le attività nella Guida per sviluppatori di Amazon Elastic Container Service.

Specificare un file system Amazon EFS nella definizione del processo

Per utilizzare i volumi del file system Amazon EFS per i tuoi contenitori, devi specificare le configurazioni del volume e del punto di montaggio nella definizione del processo. Il seguente frammento JSON di definizione del processo mostra la sintassi per gli oggetti `volumes` e `mountPoints` per un contenitore:

```
{
    "containerProperties": [
        {
            "image": "amazonlinux:2",
            "command": [
                "ls",
                "-la",
                "/mount/efs"
            ],
            "mountPoints": [
                {
                    "sourceVolume": "myEfsVolume",
                    "containerPath": "/mount/efs",
                    "readOnly": true
                }
            ],
            "volumes": [
                {
                    "name": "myEfsVolume",
```

```
                    "efsVolumeConfiguration": {
                        "fileSystemId": "fs-12345678",
                        "rootDirectory": "/path/to/my/data",
                        "transitEncryption": "ENABLED",
                        "transitEncryptionPort": integer,
                        "authorizationConfig": {
                            "accessPointId": "fsap-1234567890abcdef1",
                            "iam": "ENABLED"
                        }
                    }
                }
            ]
        }
    ]
}
```

`efsVolumeConfiguration`

Tipo: oggetto

Campo obbligatorio: no

Questo parametro viene specificato quando si utilizzano volumi Amazon EFS.

`fileSystemId`

Tipo: stringa

Campo obbligatorio: sì

L'ID del file system Amazon EFS da utilizzare.

`rootDirectory`

Tipo: string

Campo obbligatorio: no

La directory all'interno del file system Amazon EFS da montare come directory principale all'interno dell'host. Se questo parametro viene omesso, viene utilizzata la radice del volume Amazon EFS. La specifica di `/` avrà lo stesso effetto dell'omissione di questo parametro. Può contenere fino a 4.096 caratteri.

> **Important**
>
> Se un punto di accesso EFS è specificato in `authorizationConfig`, il parametro della directory principale deve essere omesso o impostato `/` su. Ciò impone il percorso impostato sul punto di accesso EFS.

`transitEncryption`

Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Determina se abilitare la crittografia per i dati Amazon EFS in transito tra l' AWS Batch host e il server Amazon EFS. Se si utilizza l'autorizzazione Amazon EFS IAM, 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' AWS Batch host 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. Il valore deve essere compreso tra 0 e 65.535. Per ulteriori informazioni, consulta Assistente per il montaggio di EFS nella Guida per l'utente di Amazon Elastic File System.

`authorizationConfig`

Tipo: oggetto

Campo obbligatorio: no

I dettagli di configurazione dell'autorizzazione per il file system Amazon EFS.

`accessPointId`

Tipo: string

Campo obbligatorio: no

L'ID del punto di accesso da utilizzare. Se viene specificato un punto di accesso, il valore della directory principale in `efsVolumeConfiguration` deve essere omesso o impostato `/` su. Ciò impone 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

Determina se utilizzare il ruolo IAM del AWS Batch lavoro definito in una definizione di processo 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 sul ruolo di esecuzione IAM, consultaAWS Batch Ruolo di esecuzione IAM.

Esempi di definizione di Job

Gli esempi di definizione dei processi riportati nei seguenti argomenti illustrano come utilizzare modelli comuni come le variabili di ambiente, la sostituzione dei parametri e gli aumenti di volume.

Indice

Variabili di ambiente

L'esempio seguente di definizione del processo utilizza variabili di ambiente per specificare un tipo di file e un URL Amazon S3. Questo esempio particolare è tratto dal post sul blog Creating a Simple

«Fetch & Run» AWS Batch Job compute. Lo `fetch_and_run.sh`script descritto nel post del blog utilizza queste variabili di ambiente per scaricare `myjob.sh` lo script da S3 e dichiararne il tipo di file.

Anche se le variabili di comando e ambiente sono codificate nella definizione del processo in questo esempio, puoi specificare le sostituzioni delle variabili di comando e ambiente per rendere la definizione del processo più versatile.

```
{
    "jobDefinitionName": "fetch_and_run",
    "type": "container",
    "containerProperties": {
        "image": "123456789012.dkr.ecr.us-east-1.amazonaws.com/fetch_and_run",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "myjob.sh",
            "60"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/AWSBatchS3ReadOnly",
        "environment": [
            {
                "name": "BATCH_FILE_S3_URL",
                "value": "s3://amzn-s3-demo-source-bucket/myjob.sh"
            },
            {
                "name": "BATCH_FILE_TYPE",
                "value": "script"
            }
        ],
        "user": "nobody"
    }
}
```

Sostituzione dei parametri

La seguente definizione di processo di esempio illustra come permettere la sostituzione dei parametri e impostare valori predefiniti.

Le dichiarazioni `Ref::` nella sezione `command` vengono utilizzate per impostare segnaposto per la sostituzione dei parametri. Quando si invia un processo con questa definizione, è necessario specificare le sostituzioni di parametro per popolare tali valori, ad esempio `inputfile` e `outputfile`. La `parameters` sezione che segue imposta un parametro predefinito per`codec`, ma è possibile sovrascriverlo in base alle esigenze.

Per ulteriori informazioni, consulta Parametri.

```
{
    "jobDefinitionName": "ffmpeg_parameters",
    "type": "container",
    "parameters": {"codec": "mp4"},
    "containerProperties": {
        "image": "my_repo/ffmpeg",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "2000"
            },
            {
                "type": "VCPU",
                "value": "2"
            }
        ],
        "command": [
            "ffmpeg",
            "-i",
            "Ref::inputfile",
            "-c",
            "Ref::codec",
            "-o",
            "Ref::outputfile"
        ],
        "jobRoleArn": "arn:aws:iam::123456789012:role/ECSTask-S3FullAccess",
        "user": "nobody"
    }
}
```

Verifica la funzionalità della GPU

Il seguente esempio di definizione di processo verifica se l'AMI per i carichi di lavoro su GPU descritta in Usa un'AMI per carichi di lavoro GPU è stata impostata correttamente. Questa definizione di lavoro di esempio esegue l'esempio del classificatore TensorFlow Deep MNIST da. GitHub

```
{
    "containerProperties": {
        "image": "tensorflow/tensorflow:1.8.0-devel-gpu",
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32000"
            },
            {
                "type": "VCPU",
                "value": "8"
            }
        ],
        "command": [
            "sh",
            "-c",
            "cd /tensorflow/tensorflow/examples/tutorials/mnist; python mnist_deep.py"
        ]
    },
    "type": "container",
    "jobDefinitionName": "tensorflow_mnist_deep"
}
```

È possibile creare un file con il testo JSON precedente chiamato `tensorflow_mnist_deep.json` e quindi registrare una definizione di AWS Batch processo con il seguente comando:

```
aws batch register-job-definition --cli-input-json file://tensorflow_mnist_deep.json
```

Job parallelo multinodo

La definizione del processo di esempio seguente illustra un processo parallelo multi-nodo. Per ulteriori informazioni, consulta Creazione di un flusso di lavoro di dinamica molecolare strettamente accoppiato con lavori paralleli a più nodi AWS BatchAWS nel blog Compute.

```
{
```

```
  "jobDefinitionName": "gromacs-jobdef",
  "jobDefinitionArn": "arn:aws:batch:us-east-2:123456789012:job-definition/gromacs-
jobdef:1",
  "revision": 6,
  "status": "ACTIVE",
  "type": "multinode",
  "parameters": {},
  "nodeProperties": {
    "numNodes": 2,
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes": "0:1",
        "container": {
          "image": "123456789012.dkr.ecr.us-east-2.amazonaws.com/gromacs_mpi:latest",
          "resourceRequirements": [
              {
                  "type": "MEMORY",
                  "value": "24000"
              },
              {
                  "type": "VCPU",
                  "value": "8"
              }
          ],
          "command": [],
          "jobRoleArn": "arn:aws:iam::123456789012:role/ecsTaskExecutionRole",
          "ulimits": [],
          "instanceType": "p3.2xlarge"
        }
      }
    ]
  }
}
```

Jobs

I lavori sono l'unità di lavoro da AWS Batch cui è iniziata. I job possono essere richiamati come applicazioni containerizzate eseguite su istanze di container Amazon ECS in un cluster ECS.

I processi containerizzati possono fare riferimento a un'immagine, a un comando e ai parametri di un container. Per ulteriori informazioni, consulta JobDefinition.

È possibile inviare un numero elevato di processi semplici e indipendenti.

Argomenti

Tutorial: invia un lavoro

Dopo aver registrato una definizione di processo, è possibile inviarla come processo a una coda di AWS Batch lavoro. È possibile sovrascrivere molti dei parametri specificati nella definizione del processo in fase di esecuzione.

Come inviare un processo

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona quello Regione AWS da usare.
3. Nel riquadro di navigazione scegliere Jobs (Processi).
4. Scegli Invia nuovo lavoro.
5. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può avere una lunghezza massima di 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
6. Per Job definition, scegli una definizione di job esistente per il tuo job. Per ulteriori informazioni, consulta Creare una definizione di processo a nodo singolo .
7. Per Job queue, scegli una coda lavori esistente. Per ulteriori informazioni, consulta Creare una coda di lavoro.
8. Per Job dependencies, scegli Add Job dependencies.
   - Per Job id, inserisci l'ID del lavoro per eventuali dipendenze. Quindi scegli Aggiungi dipendenze lavorative. Un lavoro può avere fino a 20 dipendenze. Per ulteriori informazioni, consulta Dipendenze dal lavoro.
9. (Solo processi in array) Per Array size (Dimensione array), specifica una dimensione dell'array compresa tra 2 e 10.000.
10. (Facoltativo) Espandi Tag, quindi scegli Aggiungi tag per aggiungere tag alla risorsa. Inserisci una chiave e un valore opzionale, quindi scegli Aggiungi tag.
11. Scegli Pagina successiva.
12. Nella sezione Job overrides:
    a. (Facoltativo) Per Priorità di pianificazione, immettete un valore di priorità di pianificazione compreso tra 0 e 100. Ai valori più alti viene data una priorità maggiore.

b. (Facoltativo) In Tentativi di lavoro, immettete il numero massimo di volte in cui si AWS Batch tenta di spostare il lavoro a uno `RUNNABLE` stato. È possibile inserire un numero compreso tra 1 e 10. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

c. (Facoltativo) Per il timeout di esecuzione, immettete il valore di timeout (in secondi). Il timeout di esecuzione è il periodo di tempo prima che un lavoro incompiuto venga terminato. Se un tentativo supera la durata del timeout, viene interrotto e passa a uno stato. `FAILED` Per ulteriori informazioni, consulta Job timeout. Il valore minimo è 60 secondi.

> **Important**
>
> Non fare affidamento sul fatto che i lavori eseguiti sulle risorse di Fargate durino per più di 14 giorni. Dopo 14 giorni, le risorse di Fargate potrebbero non essere più disponibili e il lavoro potrebbe essere interrotto.

d. (Facoltativo) Attiva i tag Propagate per propagare i tag dal processo e dalla definizione del processo al task Amazon ECS.

13. Espandere Additional configuration (Configurazione aggiuntiva).
14. (Facoltativo) Per le condizioni della strategia Retry, scegli Aggiungi valutazione all'uscita. Inserisci almeno un valore di parametro, quindi scegli un'azione. Per ogni set di condizioni, l'azione deve essere impostata su Riprova o Esci. Queste azioni significano quanto segue:

    - Riprova: AWS Batch riprova fino al raggiungimento del numero di tentativi di lavoro specificato.
    - Esci: AWS Batch interrompe l'esecuzione di un nuovo tentativo.

    > **Important**
    >
    > Se scegli Aggiungi valutazione all'uscita, configura almeno un parametro e scegli un'azione oppure scegli Rimuovi valutazione all'uscita.

15. Per Parametri, scegli Aggiungi parametri per aggiungere segnaposto di sostituzione dei parametri. Quindi, inserite una chiave e un valore opzionale.
16. Nella sezione Container overrides:

    a. Per Command, inserisci i comandi nel campo come equivalenti all'array di stringhe JSON.

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 `CMD` parametro Docker, vedere https://docs.docker.com/engine/reference/builder/ #cmd.

> **Note**
>
> Questo parametro non può contenere una stringa vuota.

b. Per v CPUs, inserisci il numero di v CPUs da riservare per il contenitore. Questo parametro è mappato a `CpuShares` nella sezione Create a container di Docker Remote API e l'opzione `--cpu-shares` a docker run. Ogni vCPU equivale a 1.024 condivisioni di CPU. Devi specificare almeno un vCPU.

c. Per Memoria, inserisci il limite di memoria disponibile per il contenitore. Se il container tenta di superare la memoria specificata qui, viene arrestato. Questo parametro è mappato a `Memory` nella sezione Create a container di Docker Remote API e l'opzione `--memory` a docker run. Per un processo, è necessario specificare almeno 4 MiB di memoria.

> **Note**
>
> Per massimizzare l'utilizzo delle risorse, dai la priorità alla memoria per i lavori di un tipo di istanza specifico. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

d. (Facoltativo) Per Numero di GPUs, scegli il numero GPUs da riservare per il contenitore.

e. (Facoltativo) Per le variabili di ambiente, scegliete Aggiungi variabile di ambiente per aggiungere variabili di ambiente come coppie nome-valore. Queste variabili vengono passate al contenitore.

f. Scegli Pagina successiva.

g. Per Job review, rivedi i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Quando hai finito, scegli Crea definizione del lavoro.

Lavori di assistenza in AWS Batch

AWS Batch i lavori di assistenza consentono di inviare richieste ai AWS servizi tramite code di AWS Batch lavoro. Attualmente, AWS Batch supporta SageMaker Training jobs as service jobs. A differenza dei job containerizzati che AWS Batch gestiscono l'esecuzione del container sottostante,

i service job consentono di AWS Batch fornire funzionalità di pianificazione e accodamento dei job mentre il AWS servizio di destinazione (come l' SageMaker IA) gestisce l'esecuzione effettiva del lavoro.

AWS Batch for SageMaker Training jobs consente ai data scientist di inviare lavori di formazione con priorità a code configurabili, garantendo che i carichi di lavoro vengano eseguiti senza intervento non appena le risorse sono disponibili. Questa funzionalità affronta sfide comuni come il coordinamento delle risorse, la prevenzione di spese eccessive accidentali, il rispetto dei vincoli di budget, l'ottimizzazione dei costi con istanze riservate e l'eliminazione della necessità di coordinamento manuale tra i membri del team.

I lavori di assistenza differiscono dai lavori containerizzati in diversi modi principali:

- Invio del lavoro: i lavori di assistenza devono essere inviati utilizzando l'SubmitServiceJobAPI. I lavori di assistenza non possono essere inviati tramite la AWS Batch console.
- Esecuzione del lavoro: AWS Batch pianifica e mette in coda i lavori di servizio, ma il AWS servizio di destinazione esegue il carico di lavoro effettivo.
- Identificatori di risorse: i job di servizio utilizzano quelli ARNs che contengono «service-job» anziché «job» per distinguerli dai job containerizzati.

Per iniziare con AWS Batch Service Jobs for Training, consulta. SageMaker the section called “Guida introduttiva AWS Batch all' SageMaker IA”

Argomenti

- Carichi utili per lavori di assistenza in AWS Batch
- Invia un lavoro di assistenza in AWS Batch
- Mappatura dello stato AWS Batch del lavoro del servizio allo stato dell' SageMaker IA
- Strategie di riprova di lavoro nell'assistenza in AWS Batch
- Monitora i lavori di assistenza in coda AWS Batch

Carichi utili per lavori di assistenza in AWS Batch

Quando si inviano lavori di assistenza utilizzando SubmitServiceJob, si forniscono due parametri chiave che definiscono il lavoro:`serviceJobType`, `eserviceRequestPayload`.

- `serviceJobType`Specificano quale AWS servizio eseguirà il lavoro. Per i lavori di SageMaker formazione, questo valore è`SAGEMAKER_TRAINING`.
- `serviceRequestPayload`è una stringa con codifica JSON che contiene la richiesta completa che normalmente verrebbe inviata direttamente al servizio di destinazione. Per i lavori di SageMaker formazione, questo payload contiene gli stessi parametri che utilizzeresti con l'API AI. SageMaker CreateTrainingJob

Per un elenco completo di tutti i parametri disponibili e delle relative descrizioni, consulta il riferimento all'CreateTrainingJobAPI SageMaker AI. Tutti i parametri supportati da `CreateTrainingJob` possono essere inclusi nel payload del job di servizio.

Per esempi di altre configurazioni dei lavori di formazione APIs, consulta la CLI SDKs e SageMaker la AI Developer Guide.

Si consiglia di utilizzare PySDK per la creazione di lavori di assistenza perché PySDK dispone di classi e utilità di supporto. Per un esempio di utilizzo di PySDK, vedi esempi di intelligenza artificiale su. SageMaker GitHub

Esempio di service job payload

L'esempio seguente mostra un semplice payload di lavoro di servizio per un processo di SageMaker formazione che esegue uno script di formazione «hello world»:

Questo payload verrebbe passato come stringa JSON al `serviceRequestPayload` parametro durante la chiamata. `SubmitServiceJob`

```
{
  "TrainingJobName": "my-simple-training-job",
  "RoleArn": "arn:aws:iam::123456789012:role/SageMakerExecutionRole",
  "AlgorithmSpecification": {
    "TrainingInputMode": "File",
    "TrainingImage": "763104351884.dkr.ecr.us-west-2.amazonaws.com/pytorch-
training:2.0.0-cpu-py310",
    "ContainerEntrypoint": [
      "echo",
      "hello world"
    ]
  },
  "ResourceConfig": {
    "InstanceType": "ml.c5.xlarge",
    "InstanceCount": 1,
```

```
    "VolumeSizeInGB": 1
  },
  "OutputDataConfig": {
    "S3OutputPath": "s3://your-output-bucket/output"
  },
  "StoppingCondition": {
    "MaxRuntimeInSeconds": 30
  }
}
```

Invia un lavoro di assistenza in AWS Batch

Per inviare lavori di assistenza a AWS Batch, si utilizza l'SubmitServiceJobAPI. Puoi inviare lavori utilizzando il AWS CLI o SDK.

Se non disponi già di un ruolo di esecuzione, devi crearne uno prima di poter inviare il lavoro di servizio. Per creare il ruolo di esecuzione SageMaker AI, consulta Come utilizzare i ruoli di esecuzione SageMaker AI nella guida per sviluppatori SageMaker AI.

Flusso di lavoro per l'invio dei lavori di assistenza

Quando invii un lavoro di assistenza, segui questo AWS Batch flusso di lavoro:

1. AWS Batch riceve la `SubmitServiceJob` richiesta e convalida i parametri AWS Batch specifici. `serviceRequestPayload`Viene passato senza convalida.
2. Il lavoro entra nello `SUBMITTED` stato e viene inserito nella coda dei lavori specificata
3. AWS Batch valuta se c'è capacità disponibile nell'ambiente di servizio per i `RUNNABLE` lavori in prima fila
4. Se la capacità è disponibile, il lavoro passa `SCHEDULED` e il lavoro è stato passato all'IA SageMaker
5. Quando la capacità è stata acquisita e l' SageMaker IA ha scaricato i dati del lavoro di servizio, il job di servizio inizierà l'inizializzazione e il lavoro verrà modificato in`STARTING`.
6. Quando l' SageMaker IA inizia a eseguire il lavoro, il suo stato viene modificato in`RUNNING`.
7. Mentre l' SageMaker IA esegue il lavoro, ne AWS Batch monitora l'avanzamento e mappa gli stati del servizio in base agli stati del AWS Batch lavoro. Per informazioni dettagliate su come vengono mappati gli stati di lavoro del servizio, consulta ???
8. Quando il lavoro di assistenza è completato, passa a `SUCCEEDED` e qualsiasi output è pronto per essere scaricato.

Prerequisiti

Prima di inviare un lavoro di assistenza, assicurati di avere:

- Ambiente di servizio: un ambiente di servizio che definisce i limiti di capacità. Per ulteriori informazioni, consulta Crea un ambiente di servizio in AWS Batch.
- SageMaker job queue: una coda di SageMaker lavoro per fornire la pianificazione dei lavori. Per ulteriori informazioni, consulta Crea una coda SageMaker di lavoro di formazione in AWS Batch.
- Autorizzazioni IAM: autorizzazioni per creare e gestire code di AWS Batch lavoro e ambienti di servizio. Per ulteriori informazioni, consulta AWS Batch Politiche, ruoli e autorizzazioni IAM.

Invia un lavoro di assistenza con la AWS CLI

Di seguito viene illustrato come inviare un lavoro di servizio utilizzando la AWS CLI:

```
aws batch submit-service-job \
    --job-name "my-sagemaker-training-job" \
    --job-queue "my-sagemaker-job-queue" \
    --service-job-type "SAGEMAKER_TRAINING" \
    --service-request-payload '{\"TrainingJobName\": \"sagemaker-training-job-
example\", \"AlgorithmSpecification\": {\"TrainingImage\": \"123456789012.dkr.ecr.us-
east-1.amazonaws.com/pytorch-inference:1.8.0-cpu-py3\", \"TrainingInputMode
\": \"File\", \"ContainerEntrypoint\":  [\"sleep\", \"1\"]}, \"RoleArn\":
\"arn:aws:iam::123456789012:role/SageMakerExecutionRole\", \"OutputDataConfig\":
 {\"S3OutputPath\": \"s3://example-bucket/model-output/\"}, \"ResourceConfig\":
 {\"InstanceType\": \"ml.m5.large\", \"InstanceCount\": 1, \"VolumeSizeInGB\": 1}}'
    --client-token "unique-token-12345"
```

Per ulteriori informazioni sui `serviceRequestPayload` parametri, vederethe section called “Carichi utili per lavori di assistenza”.

Mappatura dello stato AWS Batch del lavoro del servizio allo stato dell' SageMaker IA

Quando si inviano lavori a una coda di SageMaker lavoro utilizzando SubmitServiceJob, AWS Batch gestisce il ciclo di vita del lavoro e mappa gli stati dei lavori su stati di AWS Batch lavoro di SageMaker formazione equivalenti. I lavori di assistenza, come i lavori di SageMaker formazione, seguono un ciclo di vita a stati diverso rispetto ai tradizionali lavori in container. Sebbene i lavori di

assistenza condividano la maggior parte degli stati con i lavori relativi ai container, essi introducono lo SCHEDULED stato e presentano comportamenti diversi per quanto riguarda i tentativi di ripetizione, in particolare per quanto riguarda la gestione di errori di capacità insufficiente del servizio di destinazione.

La tabella seguente mostra lo stato del AWS Batch lavoro e lo Status/ corrispondente: SageMaker SecondaryStatus

Stato del Batch	SageMaker Stato primario dell'IA	SageMaker Status secondario dell'IA	Description
SUBMITTED	N/D	N/D	Job inviato in coda, in attesa di valutazione da parte dello scheduler.
RUNNABLE	N/D	N/D	Job è in coda e pronto per la pianificazione. I lavori in questo stato vengono avviati non appena sono disponibili risorse sufficienti nell'ambiente di servizio. I posti di lavoro possono rimanere in questo stato a tempo indeterminato quando non sono disponibili risorse sufficienti.
SCHEDULED	InProgress	Pending	Processo di assistenza inviato con successo a AI SageMaker
STARTING	InProgress	Downloading	SageMaker Lavoro di formazione nel download di dati e immagini. La capacità lavorativa di formazione è stata acquisita e l'inizializzazione del lavoro è iniziata.
RUNNING	InProgress	Training	SageMaker Algoritmo di esecuzione dei lavori di formazione
RUNNING	InProgress	Uploading	SageMaker Processo di formazione: caricamento degli artefatti di output dopo il completamento dell'addestramento

Stato del Batch	SageMaker Stato primario dell'IA	SageMaker Status secondario dell'IA	Description
`SUCCEEDED`	`Completed`	`Completed`	SageMaker Lavoro di formazione completato con successo. Caricamento terminato degli artefatti di output.
`FAILED`	`Failed`	`Failed`	SageMaker Il processo di formazione ha riscontrato un errore irreversibile.
`FAILED`	`Stopped`	`Stopped`	SageMaker Il processo di formazione è stato interrotto manualmente. `StopTrainingJob`

Strategie di riprova di lavoro nell'assistenza in AWS Batch

Le strategie di ripetizione dei lavori di assistenza consentono AWS Batch di riprovare automaticamente i lavori di servizio non riusciti in condizioni specifiche.

I lavori di assistenza possono richiedere più tentativi per diversi motivi:

- Problemi temporanei di servizio: errori interni del servizio, rallentamenti o interruzioni temporanee possono causare il fallimento dei lavori durante l'invio o l'esecuzione.
- Errori di inizializzazione dell'addestramento: i problemi durante l'avvio del processo, ad esempio problemi di acquisizione delle immagini o errori di inizializzazione, possono essere risolti riprovando.

Configurando strategie di riprova appropriate, è possibile migliorare le percentuali di successo del lavoro e ridurre la necessità di interventi manuali, in particolare per carichi di lavoro di formazione di lunga durata.

> **Note**
>
> I job di assistenza riprovano automaticamente determinati tipi di errori, ad esempio errori di capacità insufficiente, senza consumare i tentativi di riprova configurati. La strategia di riprova

consente di gestire principalmente altri tipi di errori, ad esempio errori di algoritmo o problemi di servizio.

Configurazione delle strategie di ripetizione dei tentativi

Le strategie di ripetizione dei processi di servizio sono configurate utilizzando ServiceJobRetryStrategy, che supporta sia il semplice numero di tentativi che la logica di ripetizione condizionale.

Configurazione del nuovo tentativo

La strategia di riprova più semplice specifica il numero di tentativi di nuovo tentativo da effettuare se un job di servizio fallisce:

```
{
  "retryStrategy": {
    "attempts": 3
  }
}
```

Questa configurazione consente di ritentare il job di servizio fino a 3 volte in caso di errore.

Important

Il `attempts` valore rappresenta il numero totale di volte in cui il lavoro può essere collocato nello `RUNNABLE` stato, incluso il tentativo iniziale. Il valore 3 indica che il processo verrà tentato inizialmente una volta, quindi ritentato fino a 2 volte in caso di esito negativo.

Riprova la configurazione con evaluateOnExit

È possibile utilizzare il `evaluateOnExit` parametro per specificare le condizioni in base alle quali i lavori devono essere ritentati o lasciati fallire. Ciò è utile quando diversi tipi di errori richiedono una gestione diversa.

L'`evaluateOnExit`array può contenere fino a 5 strategie di ripetizione, ognuna delle quali specifica un'azione (`RETRYoEXIT`) e condizioni basate sui motivi dello stato:

```
{
  "retryStrategy": {
```

```
    "attempts": 5,
    "evaluateOnExit": [
      {
        "action": "RETRY",
        "onStatusReason": "Received status from SageMaker: InternalServerError*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "Received status from SageMaker: ValidationException*"
      },
      {
        "action": "EXIT",
        "onStatusReason": "*"
      }
    ]
  }
}
```

Questa configurazione:

- Riprova i lavori che falliscono a causa di errori interni del server SageMaker AI
- Fallisce immediatamente i lavori che presentano eccezioni di convalida (errori del client che non verranno risolti con un nuovo tentativo)
- Include una regola generale per uscire da qualsiasi altro tipo di errore

Corrispondenza del motivo dello stato

Il `onStatusReason` parametro supporta la corrispondenza dei modelli con un massimo di 512 caratteri. I pattern possono utilizzare caratteri jolly (*) e corrispondere ai motivi di stato restituiti dall' SageMaker IA.

Per i lavori di assistenza, i messaggi di stato dell' SageMaker IA hanno il prefisso «Stato ricevuto da SageMaker:» per distinguerli dai messaggi AWS Batch generati. I modelli più comuni includono:

- `Received status from SageMaker: InternalServerError*`- Rispondi agli errori interni del servizio
- `Received status from SageMaker: ValidationException*`- Corrisponde agli errori di convalida del client
- `Received status from SageMaker: ResourceLimitExceeded*`- Rispondi agli errori relativi al limite delle risorse

- `*CapacityError*`- Risolvi gli errori relativi alla capacità

> **Tip**
>
> Utilizza un pattern matching specifico per gestire in modo appropriato diversi tipi di errore. Ad esempio, riprovate a commettere errori interni del server, ma non riuscite immediatamente in caso di errori di convalida che indicano problemi con i parametri del lavoro.

Monitora i lavori di assistenza in coda AWS Batch

Puoi monitorare lo stato dei lavori nella coda dei lavori di SageMaker Training utilizzando`list-service-jobs`, e`get-job-queue-snapshot`.

Visualizza i lavori in esecuzione nella tua coda:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status RUNNING
```

Visualizza i lavori in attesa in coda:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status RUNNABLE
```

Visualizza i lavori che sono stati inviati SageMaker ma non ancora in esecuzione:

```
aws batch list-service-jobs \
  --job-queue my-sm-training-fifo-jq \
  --job-status SCHEDULED
```

Ottieni un'istantanea dei lavori in primo piano:

```
aws batch get-job-queue-snapshot --job-queue my-sm-training-fifo-jq
```

Questo comando mostra l'ordine dei lavori di assistenza imminenti nella coda.

Ottieni informazioni dettagliate su un lavoro di assistenza specifico:

```
aws batch describe-service-job --jobId job-id
```

Job stati

Quando si invia un lavoro a una coda di AWS Batch lavoro, il lavoro entra nello `SUBMITTED` stato. Dopodiché passa per gli stati successivi, fino a quando non ha esito positivo (codice di uscita `0`) o negativo (codice di uscita diverso da zero). I processi di AWS Batch hanno gli stati seguenti:

`SUBMITTED`

Un lavoro che è stato inviato alla coda e non è stato ancora valutato dallo scheduler. Il pianificatore valuta il processo per stabilire se presenta dipendenze in sospeso relative al corretto completamento di altri processi. Se sono presenti dipendenze, il processo passa allo stato `PENDING`. Se non sono presenti dipendenze, il processo passa allo stato `RUNNABLE`.

`PENDING`

Un lavoro che si trova in coda e non è ancora in grado di essere eseguito a causa della dipendenza da un altro lavoro o risorsa. Quando le dipendenze sono state soddisfatte, il processo passa allo stato `RUNNABLE`.

`RUNNABLE`

Un processo che si trova nella coda, che non presenta dipendenze in sospeso e che è quindi pronto per essere pianificato per un host. I lavori in questo stato vengono avviati non appena sono disponibili risorse sufficienti in uno degli ambienti di calcolo mappati alla coda del lavoro. Tuttavia, i processo possono rimanere in questo stato in modo indefinito se le risorse sufficienti non sono disponibili.

> **Note**
>
> Se i tuoi lavori non procedono`STARTING`, consulta la sezione relativa alla risoluzione dei Lavori bloccati in uno `RUNNABLE` status problemi.

`STARTING`

Questi processi sono stati pianificati per un host e le operazioni di avvio del container pertinente sono in corso. Una volta che l'immagine del container è stata estratta e il container è in esecuzione, il processo passa allo stato `RUNNING`.

La durata del pull dell'immagine, la durata del completamento di Amazon EKS InitContainer e la durata della risoluzione di Amazon ECS ContainerDependency si verificano nello stato STARTING. La quantità di tempo necessaria per estrarre un'immagine per il tuo lavoro è equivalente alla quantità di tempo in cui il lavoro rimarrà nello stato INIZIALE.

Ad esempio, se occorrono tre minuti per estrarre l'immagine del lavoro, quest'ultimo rimarrà nello stato INIZIALE per tre minuti. Se InitContainers impiega un totale di dieci minuti per essere completato, il processo Amazon EKS rimarrà in modalità STARTING per dieci minuti. Se hai impostato Amazon ECS ContainerDependencies nel tuo job Amazon ECS, il processo rimarrà in STARTING fino a quando tutte le dipendenze del contenitore (il loro runtime) non saranno risolte. STARTING non è incluso nei timeout; la durata inizia da RUNNING. Per ulteriori informazioni, consulta Job states.

`RUNNING`

Il processo viene eseguito come processo container su un'istanza di container Amazon ECS all'interno di un ambiente di calcolo. Al momento dell'uscita del container, il codice di uscita del processo determina l'esito positivo o negativo di quest'ultimo. Il codice di uscita `0` indica che il processo ha avuto esito positivo, mentre un codice di uscita diverso da zero indica che ha avuto esito negativo. Se il processo associato a un tentativo non riuscito presenta tentativi rimanenti nella sua configurazione opzionale della strategia relativa ai nuovi tentativi, il processo passa nuovamente allo stato `RUNNABLE`. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

> **Note**
>
> I registri dei `RUNNING` lavori sono disponibili in Logs. CloudWatch Il gruppo di log è`/aws/batch/job`, e il formato del nome del flusso di log è il seguente:. *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`* Questo formato potrebbe cambiare in futuro.
> Dopo che un processo raggiunge lo `RUNNING` stato, è possibile recuperare a livello di codice il nome del flusso di log con l'DescribeJobsoperazione API. Per ulteriori informazioni, consulta Visualizza i dati di log inviati ai CloudWatch registri nella Amazon CloudWatch Logs User Guide. Per impostazione predefinita, questi log non scadono mai. Tuttavia, è possibile modificare il periodo di conservazione. Per ulteriori informazioni, consulta Change Log Data Retention in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

`SUCCEEDED`

Il processo è stato completato correttamente e ha ricevuto il codice di uscita `0`. Lo stato del `SUCCEEDED` lavoro rimane invariato AWS Batch per almeno 7 giorni.

> **Note**
>
> I registri dei `SUCCEEDED` lavori sono disponibili in CloudWatch Registri. Il gruppo di log è`/aws/batch/job`, e il formato del nome del flusso di log è il seguente:. *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`* Questo formato potrebbe cambiare in futuro.
> Dopo che un processo raggiunge lo `RUNNING` stato, è possibile recuperare a livello di codice il nome del flusso di log con l'DescribeJobsoperazione API. Per ulteriori informazioni, consulta Visualizza i dati di log inviati ai CloudWatch registri nella Amazon CloudWatch Logs User Guide. Per impostazione predefinita, questi log non scadono mai. Tuttavia, è possibile modificare il periodo di conservazione. Per ulteriori informazioni, consulta Change Log Data Retention in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

`FAILED`

Il processo ha ottenuto un esito negativo per tutti i tentativi disponibili. Lo stato del `FAILED` lavoro rimane invariato AWS Batch per almeno 7 giorni.

> **Note**
>
> I registri dei `FAILED` lavori sono disponibili in CloudWatch Registri. Il gruppo di log è`/aws/batch/job`, e il formato del nome del flusso di log è il seguente:. *`first200CharsOfJobDefinitionName`*`/default/`*`ecs_task_id`* Questo formato potrebbe cambiare in futuro.
> Dopo che un lavoro raggiunge lo `RUNNING` stato, è possibile recuperarne il flusso di log a livello di codice con l'DescribeJobsoperazione API. Per ulteriori informazioni, consulta Visualizza i dati di log inviati ai CloudWatch registri nella Amazon CloudWatch Logs User Guide. Per impostazione predefinita, questi log non scadono mai. Tuttavia, è possibile modificare il periodo di conservazione. Per ulteriori informazioni, consulta Change Log Data Retention in CloudWatch Logs nella Amazon CloudWatch Logs User Guide.

AWS Batch variabili dell'ambiente di lavoro

AWS Batch imposta variabili di ambiente specifiche nei job dei container. Queste variabili di ambiente forniscono un'introspezione per i contenitori all'interno dei job. È possibile utilizzare i valori di queste variabili nella logica delle applicazioni. Tutte le variabili AWS Batch impostate iniziano con il `AWS_BATCH_` prefisso. Si tratta di un prefisso di variabile di ambiente protetto. Non è possibile utilizzare questo prefisso per le proprie variabili nelle definizioni o nelle sostituzioni dei processi.

Nei container dei processi sono disponibili le variabili di ambiente seguenti:

`AWS_BATCH_CE_NAME`

Questa variabile è impostata sul nome dell'ambiente di calcolo in cui si trova il lavoro.

`AWS_BATCH_JOB_ARRAY_INDEX`

Questa variabile viene impostata solo nei processi figlio in array. L'indice dei processi in array inizia da 0 e a ciascun processo figlio viene assegnato un numero di indice univoco. Ad esempio, i valori di indice di un processo in array con 10 elementi figlio sono compresi tra 0 e 9. È possibile utilizzare questo valore di indice per controllare il modo in cui vengono indicati i diversi elementi figlio dei processi in array. Per ulteriori informazioni, consulta Utilizza l'array job index per controllare la differenziazione dei job.

`AWS_BATCH_JOB_ARRAY_SIZE`

Questa variabile è impostata sulla dimensione del job dell'array principale. La dimensione del job dell'array principale viene passata al job dell'array secondario in questa variabile.

`AWS_BATCH_JOB_ATTEMPT`

Questa variabile è impostata sul numero di tentativi del processo. Al primo tentativo viene assegnato il numero 1. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

`AWS_BATCH_JOB_ID`

Questa variabile è impostata sull'ID del AWS Batch lavoro.

`AWS_BATCH_JOB_KUBERNETES_NODE_UID`

Questa variabile è impostata come Kubernetes UID dell'oggetto nodo che si trova nel cluster Kubernetes su cui viene eseguito il pod. Questa variabile è impostata solo per i job eseguiti su risorse Amazon EKS. Per ulteriori informazioni, UIDsconsulta la Kubernetesdocumentazione.

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Questa variabile viene impostata solo nei processi paralleli a più nodi. Questa variabile è impostata sul numero d'indice del nodo principale del processo. Il codice dell'applicazione può essere confrontato con `AWS_BATCH_JOB_MAIN_NODE_INDEX` quello di un singolo nodo per determinare se si tratta del nodo principale. `AWS_BATCH_JOB_NODE_INDEX`

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Questa variabile è impostata solo nei nodi figlio del lavoro parallelo a più nodi. Questa variabile non è presente nel nodo principale, ma è impostata sull' IPv4 indirizzo privato del nodo principale del processo. Il codice di applicazione del nodo figlio può utilizzare questo indirizzo per comunicare con il nodo principale.

`AWS_BATCH_JOB_NODE_INDEX`

Questa variabile viene impostata solo nei processi paralleli a più nodi. Questa variabile è impostata sul numero d'indice del nodo. L'indice del nodo inizia con 0 e a ciascun nodo viene assegnato un numero d'indice univoco. Ad esempio, un processo parallelo a più nodi con 10 figli ha valori d'indice compresi tra 0 e 9.

`AWS_BATCH_JOB_NUM_NODES`

Questa variabile viene impostata solo nei processi paralleli a più nodi. Questa variabile è impostata sul numero di nodi richiesti per il processo parallelo multinodo.

`AWS_BATCH_JQ_NAME`

Questa variabile è impostata sul nome della coda dei processi a cui viene inviato il processo.

Ritentativi di lavoro automatizzati

Puoi applicare ai tuoi processi e alle definizioni di processo una strategia di nuovi tentativi che consenta la ripetizione automatica dei processi in caso di esito negativo. I possibili scenari di errore includono quanto segue:

- Un codice di uscita diverso da zero ricevuto dal processo di un container
- Errore o chiusura dell' EC2 istanza Amazon
- Errore o AWS interruzione del servizio interno

Quando un lavoro viene inviato a una coda di lavoro e inserito nello `RUNNING` stato considerato un tentativo. Per impostazione predefinita, a ogni processo viene assegnato un tentativo per passare allo stato `SUCCEEDED` o `FAILED`. Tuttavia, sia il flusso di lavoro di definizione che quello di invio del lavoro possono essere utilizzati per specificare una strategia di nuovo tentativo con un numero di tentativi compreso tra 1 e 10 tentativi. Se evaluateOnExitspecificato, può contenere fino a 5 strategie di nuovo tentativo. Se evaluateOnExitviene specificato, ma nessuna delle strategie di nuovo tentativo corrisponde, il processo viene riprovato. Per i lavori che non corrispondono a exit, aggiungi una voce finale che termina per qualsiasi motivo. Ad esempio, questo `evaluateOnExit` oggetto ha due voci con azioni di `RETRY` e una voce finale con un'azione di`EXIT`.

```
"evaluateOnExit": [
    {
        "action": "RETRY",
        "onReason": "AGENT"
    },
    {
        "action": "RETRY",
        "onStatusReason": "Task failed to start"
    },
    {
        "action": "EXIT",
        "onReason": "*"
    }
]
```

In fase di runtime, la variabile di ambiente `AWS_BATCH_JOB_ATTEMPT` è impostata sul numero del tentativo corrispondente del processo del container. Il primo tentativo è numerato e `1` i tentativi successivi sono in ordine crescente (ad esempio, 2, 3, 4).

Si supponga, ad esempio, che un tentativo di lavoro abbia esito negativo per qualsiasi motivo e che il numero di tentativi specificato nella configurazione dei nuovi tentativi sia maggiore del numero. `AWS_BATCH_JOB_ATTEMPT` Quindi, il lavoro viene rimesso nello `RUNNABLE` stato. Per ulteriori informazioni, consulta Job stati.

> **Note**
>
> I lavori annullati o terminati non vengono ritentati. Inoltre, i lavori che falliscono a causa di una definizione di processo non valida non vengono ritentati.

Per ulteriori informazioni, vedere Strategia per nuovi tentativiCreare una definizione di processo a nodo singolo , Tutorial: invia un lavoro e Codici di errore delle attività interrotte.

Dipendenze dal lavoro

Quando invii un AWS Batch lavoro, puoi specificare il lavoro da IDs cui dipende il lavoro. In tal caso, lo AWS Batch scheduler garantisce che il processo venga eseguito solo dopo che le dipendenze specificate sono state completate con successo. Dopo il completamento delle dipendenze, il processo passerà dallo stato `PENDING` a `RUNNABLE`, quindi a `STARTING` e a `RUNNING`. Nel caso in cui qualsiasi dipendenza del processo abbia esito negativo, il processo passerà automaticamente dallo stato `PENDING` a `FAILED`.

Ad esempio, l'esecuzione del processo A può dipendere da un massimo di altri 20 processi che devono avere esito positivo. Dopodiché puoi inviare processi aggiuntivi che dipendono dal processo A e da un massimo di altri 19 processi.

Per i processi in array, puoi specificare una dipendenza di tipo `SEQUENTIAL` senza specificare un ID del processo, in modo che ogni processo figlio nell'array venga completato in maniera sequenziale a partire dall'indice 0. È anche possibile specificare una dipendenza tipo `N_TO_N` con un ID processo. In questo modo, prima di iniziare, ciascun figlio nell'indice di questo processo deve attendere il completamento del figlio nell'indice corrispondente di ciascuna dipendenza. Per ulteriori informazioni, consulta Lavori di matrice.

Per inviare un AWS Batch lavoro con dipendenze, vedi. Tutorial: invia un lavoro

Pianificazione basata sulle risorse consente di pianificare i lavori in base alle risorse consumabili necessarie per eseguire i lavori. Si specificano le risorse consumabili necessarie per l'esecuzione di un processo e Batch tiene conto di queste dipendenze tra le risorse quando pianifica un lavoro. È possibile ridurre il sottoutilizzo delle risorse di elaborazione allocando solo i lavori che dispongono di tutte le risorse necessarie. La pianificazione basata sulle risorse è disponibile sia per le politiche di pianificazione FIFO che per quelle di condivisione equa e può essere utilizzata con tutte le piattaforme di elaborazione supportate da Batch, tra cui EKS, ECS e Fargate. Può essere utilizzato con i job Array, i job Multi-node parallel (MNP) e con i normali job Batch.

Job timeout

Puoi configurare una durata del timeout per i tuoi processi, in modo che se un processo viene eseguito per più tempo, AWS Batch termina l'operazione. Ad esempio, potresti avere un lavoro

che sai dovrebbe richiedere solo 15 minuti per essere completato. Talvolta, l'applicazione si blocca durante un ciclo e viene eseguita sempre, perciò è possibile impostare un timeout di 30 minuti per terminare il processo bloccato.

> **Important**
>
> Per impostazione predefinita, AWS Batch non prevede un timeout di lavoro. Se non si definisce un timeout per il lavoro, il processo viene eseguito fino alla chiusura del contenitore.

Puoi specificare un parametro `attemptDurationSeconds` che deve essere pari almeno a 60 secondi, nella definizione del processo oppure quando lo invii. Trascorso questo numero di secondi dal `startedAt` timestamp del tentativo di lavoro, AWS Batch termina il processo. Nella risorsa di calcolo, il container del processo riceve un segnale `SIGTERM` per offrire alla tua applicazione la possibilità di arresto normale. Se il container è ancora in esecuzione dopo 30 secondi, viene inviato un segnale `SIGKILL` per forzare l'arresto del container.

Le chiusure di timeout sono gestite nel miglior modo possibile. Non dovresti aspettarti che la fine del timeout avvenga esattamente allo scadere del tentativo di lavoro (potrebbero essere necessari alcuni secondi in più). Se la tua applicazione richiede l'esecuzione di timeout precisi, è necessario implementare questa logica all'interno dell'applicazione. Se disponi di un numero elevato di processi con un timeout simultaneo, le cancellazioni dei timeout si comportano come una coda FIFO, in cui i processi vengono terminati in batch.

> **Note**
>
> Non esiste un valore di timeout massimo per un lavoro. AWS Batch

Se un lavoro viene interrotto per aver superato la durata del timeout, non viene ritentato. Se il tentativo di un processo ha esito negativo, è possibile riprovare (se sono stati abilitati altri tentativi) e viene avviato il conteggio del timeout per il nuovo tentativo.

> **Important**
>
> I lavori eseguiti con risorse Fargate non possono aspettarsi che durino più di 14 giorni. Se la durata del timeout supera i 14 giorni, le risorse di Fargate potrebbero non essere più disponibili e il lavoro verrà interrotto.

Per i processi in array, i processi figlio hanno la stessa configurazione di timeout del processo padre.

Per informazioni sull'invio di un AWS Batch lavoro con una configurazione di timeout, vedere. Tutorial: invia un lavoro

Offerte di lavoro Amazon EKS

Un lavoro è l'unità di lavoro più piccola in AWS Batch. Un AWS Batch lavoro su Amazon EKS prevede una one-to-one mappatura su un Kubernetes pod. Una definizione di AWS Batch lavoro è un modello per un AWS Batch lavoro. Quando si invia un AWS Batch lavoro, si fa riferimento a una definizione di processo, si sceglie come target una coda di lavoro e si fornisce un nome per un lavoro. Nella definizione di un AWS Batch processo su Amazon EKS, il parametro EksProperties definisce l'insieme di parametri supportati da un processo AWS Batch su Amazon EKS. In una SubmitJobrichiesta, il eksPropertiesOverrideparametro consente di sostituire alcuni parametri comuni. In questo modo, è possibile utilizzare modelli di definizioni di lavoro per più lavori. Quando un lavoro viene inviato al tuo cluster Amazon EKS, AWS Batch trasforma il lavoro in un `podspec` (). `Kind: Pod` `podspec`Utilizza alcuni AWS Batch parametri aggiuntivi per garantire che i lavori siano scalati e pianificati correttamente. AWS Batch combina etichette e taint per garantire che i job vengano eseguiti solo su nodi AWS Batch gestiti e che altri pod non vengano eseguiti su tali nodi.

> **Important**
>
> - Se il `hostNetwork` parametro non è impostato in modo esplicito in una definizione di processo Amazon EKS, la modalità di rete pod è AWS Batch predefinita in modalità host. Più specificamente, vengono applicate le seguenti impostazioni: e. `hostNetwork=true` `dnsPolicy=ClusterFirstWithHostNet`
> - AWS Batch pulisce i job pod subito dopo che un pod ha completato il suo lavoro. Per visualizzare i log delle applicazioni dei pod, configura un servizio di registrazione per il tuo cluster. Per ulteriori informazioni, consulta Usa CloudWatch Logs per monitorare i lavori AWS Batch su Amazon EKS.

Argomenti

- Tutorial: mappa un job in esecuzione su un pod e un nodo
- Tutorial: Riporta un pod in esecuzione al suo lavoro

Tutorial: mappa un job in esecuzione su un pod e un nodo

I `nodeName` parametri `podProperties` di un processo in esecuzione sono stati `podName` impostati per il tentativo di lavoro corrente. Utilizzate l'operazione DescribeJobsAPI per visualizzare questi parametri.

Di seguito è riportato un output di esempio.

```
$ aws batch describe-jobs --job 2d044787-c663-4ce6-a6fe-f2baf7e51b04
{
 "jobs": [
  {
   "status": "RUNNING",
   "jobArn": "arn:aws:batch:us-east-1:123456789012:job/2d044787-c663-4ce6-a6fe-
f2baf7e51b04",
   "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/
MyJobOnEks_SleepWithRequestsOnly:1",
   "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/My-Eks-JQ1",
   "jobId": "2d044787-c663-4ce6-a6fe-f2baf7e51b04",
   "eksProperties": {
    "podProperties": {
     "nodeName": "ip-192-168-55-175.ec2.internal",
     "containers": [
      {
       "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
       "resources": {
        "requests": {
         "cpu": "1",
         "memory": "1024Mi"
        }
       }
      }
     ],
     "podName": "aws-batch.b0aca953-ba8f-3791-83e2-ed13af39428c"
    }
   }
  }
 ]
}
```

Per un lavoro con nuovi tentativi abilitati, `podName` la fine `nodeName` di ogni tentativo completato è inclusa nel parametro `eksAttempts` list dell'operazione DescribeJobsAPI. La `podName` fine `nodeName` del tentativo di esecuzione corrente si trova nell'`podProperties`oggetto.

Tutorial: Riporta un pod in esecuzione al suo lavoro

Un pod ha delle etichette che indicano `jobId` la fine `uuid` dell'ambiente di calcolo a cui appartiene. AWS Batch inserisce variabili di ambiente in modo che il runtime del lavoro possa fare riferimento alle informazioni sul lavoro. Per ulteriori informazioni, consulta AWS Batch variabili dell'ambiente di lavoro. È possibile visualizzare queste informazioni eseguendo il comando seguente. L'output è il seguente.

```
$ kubectl describe pod aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1 -n my-aws-batch-
namespace
Name:         aws-batch.14638eb9-d218-372d-ba5c-1c9ab9c7f2a1
Namespace:    my-aws-batch-namespace
Priority:     0
Node:         ip-192-168-45-88.ec2.internal/192.168.45.88
Start Time:   Wed, 26 Oct 2022 00:30:48 +0000
Labels:       batch.amazonaws.com/compute-environment-uuid=5c19160b-
d450-31c9-8454-86cf5b30548f
              batch.amazonaws.com/job-id=f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
              batch.amazonaws.com/node-uid=a4be5c1d-9881-4524-b967-587789094647
...
Status:       Running
IP:           192.168.45.88
IPs:
  IP:  192.168.45.88
Containers:
  default:
    Image:         public.ecr.aws/amazonlinux/amazonlinux:2
    ...
    Environment:
      AWS_BATCH_JOB_KUBERNETES_NODE_UID:  a4be5c1d-9881-4524-b967-587789094647
      AWS_BATCH_JOB_ID:                   f980f2cf-6309-4c77-a2b2-d83fbba0e9f0
      AWS_BATCH_JQ_NAME:                  My-Eks-JQ1
      AWS_BATCH_JOB_ATTEMPT:              1
      AWS_BATCH_CE_NAME:                  My-Eks-CE1

...
```

Funzionalità supportate da AWS Batch Amazon EKS jobs

Queste sono le caratteristiche AWS Batch specifiche comuni anche ai Kubernetes job eseguiti su Amazon EKS:

- Dipendenze dal lavoro
- Lavori di matrice
- Job timeout
- Ritentativi di lavoro automatizzati
- Usa la pianificazione con condivisione equa per aiutarti a pianificare i lavori

Kubernetes**`Secrets`** e **`ServiceAccounts`**

AWS Batch supporta riferimenti Kubernetes `Secrets` e. `ServiceAccounts` Puoi configurare i pod per utilizzare i ruoli IAM di Amazon EKS per gli account di servizio. Per ulteriori informazioni, consulta Configurazione dei pod per l'utilizzo di un account di Kubernetes servizio nella Amazon EKS User Guide.

Documenti correlati

- Considerazioni su AWS Batch memoria e vCPU per Amazon EKS
- Esegui lavori GPU
- Lavori bloccati in uno `RUNNABLE` status

Lavori paralleli multinodo

Puoi utilizzare processi paralleli multinodo per eseguire singoli processi che si estendono su più istanze Amazon EC2 . Con i processi paralleli AWS Batch multinodo (noti anche come pianificazione di gruppo), puoi eseguire applicazioni di elaborazione su larga scala e ad alte prestazioni e addestrare modelli GPU distribuiti senza la necessità di avviare, configurare e gestire direttamente le risorse Amazon. EC2 Un job parallelo AWS Batch multinodo è compatibile con qualsiasi framework che supporti la comunicazione tra nodi basata su IP. Gli esempi includono Apache MXNet TensorFlow, Caffe2 o Message Passing Interface (MPI).

I lavori paralleli multinodo vengono inviati come processo singolo. Tuttavia, la definizione del processo (o sostituzioni del nodo di invio del processo) specifica il numero di nodi da creare per

il processo e quali gruppi di nodo creare. Ogni processo parallelo a più nodi contiene un nodo principale, che viene avviato prima. Dopo che il nodo principale è attivo, i nodi secondari vengono avviati e avviati. Il processo è terminato solo se il nodo principale viene chiuso. Tutti i nodi secondari vengono quindi interrotti. Per ulteriori informazioni, consulta Gruppi di nodi.

I nodi di lavoro paralleli multinodo sono single-tenant. Ciò significa che su ogni EC2 istanza Amazon viene eseguito un solo contenitore di job.

Lo stato del processo finale (`SUCCEEDED` o `FAILED`) è determinato dallo stato del processo finale del nodo principale. Per conoscere lo stato di un processo parallelo a più nodi, descrivi il lavoro utilizzando l'ID del lavoro restituito al momento dell'invio del lavoro. Se hai bisogno dei dettagli per i nodi secondari, descrivi ogni nodo figlio singolarmente. È possibile indirizzare i nodi utilizzando la *#N* notazione (a partire da 0). Ad esempio, per accedere ai dettagli del secondo nodo di un job, descrivi *aws_batch_job_id* #1 utilizzando l'operazione AWS Batch DescribeJobsAPI. Le informazioni `started`, `stoppedAt`, `statusReason` e `exit` per un processo parallelo a più nodi, vengono popolate dal nodo principale.

Se si specificano nuovi tentativi di lavoro, un errore del nodo principale causa un altro tentativo. Gli errori dei nodi secondari non causano ulteriori tentativi. Ogni nuovo tentativo di un processo parallelo a più nodi aggiorna il tentativo corrispondente dei suoi nodi figlio associati.

Per eseguire lavori paralleli a più nodi AWS Batch, il codice dell'applicazione deve contenere i framework e le librerie necessari per la comunicazione distribuita.

Argomenti

- Variabili di ambiente
- Gruppi di nodi
- Ciclo di vita del lavoro per i lavori MNP
- Considerazioni sull'ambiente di calcolo per MNP con AWS Batch

Variabili di ambiente

In fase di esecuzione, ogni nodo è configurato in base alle variabili di ambiente standard ricevute da tutti AWS Batch i job. Inoltre, i nodi sono configurati con le seguenti variabili di ambiente specifiche per i lavori paralleli a più nodi:

`AWS_BATCH_JOB_MAIN_NODE_INDEX`

Questa variabile è impostata sul numero d'indice del nodo principale del processo. Il codice dell'applicazione può essere confrontato `AWS_BATCH_JOB_MAIN_NODE_INDEX` con quello di un singolo nodo per determinare se si tratta del nodo principale. `AWS_BATCH_JOB_NODE_INDEX`

`AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS`

Questa variabile è impostata solo nei nodi figlio del lavoro parallelo a più nodi. Questa variabile non è presente nel nodo principale. Questa variabile è impostata sull' IPv4 indirizzo privato del nodo principale del lavoro. Il codice di applicazione del nodo figlio può utilizzare questo indirizzo per comunicare con il nodo principale.

`AWS_BATCH_JOB_NODE_INDEX`

Questa variabile è impostata sul numero d'indice del nodo. L'indice del nodo inizia con 0 e a ciascun nodo viene assegnato un numero d'indice univoco. Ad esempio, un processo parallelo a più nodi con 10 figli ha valori d'indice compresi tra 0 e 9.

`AWS_BATCH_JOB_NUM_NODES`

Questa variabile è impostata sul numero di nodi che hai richiesto per il tuo processo parallelo a più nodi.

Gruppi di nodi

Un gruppo di nodi è un gruppo identico di nodi di lavoro che condividono tutti le stesse proprietà del contenitore. È possibile utilizzare AWS Batch per specificare fino a cinque gruppi di nodi distinti per ogni job.

Ogni gruppo può avere immagini container, comandi, variabili d'ambiente propri. Ad esempio, è possibile inviare un processo che richiede una singola `c5.xlarge` istanza per il nodo principale e cinque nodi figlio di `c5.xlarge` istanza. Ciascuno di questi gruppi di nodi distinti può specificare diverse immagini o comandi del contenitore da eseguire per ogni processo.

In alternativa, tutti i nodi del job possono utilizzare un singolo gruppo di nodi. Inoltre, il codice dell'applicazione può differenziare i ruoli dei nodi, come il nodo principale e il nodo secondario. A tale scopo, confronta la variabile di `AWS_BATCH_JOB_MAIN_NODE_INDEX` ambiente con il proprio valore di`AWS_BATCH_JOB_NODE_INDEX`. È possibile avere fino a 1.000 nodi in un singolo processo. Questo è il limite predefinito per le istanze in un cluster Amazon ECS. Puoi richiedere di aumentare questo limite.

> **Note**
>
> Attualmente tutti i gruppi di nodi in un processo parallelo multinodo devono utilizzare lo stesso tipo di istanza.

Ciclo di vita del lavoro per i lavori MNP

Quando si invia un processo parallelo multinodo, il lavoro entra nello `SUBMITTED` stato. Quindi, il lavoro attende il completamento di eventuali dipendenze tra i lavori. Il lavoro passa anche allo stato. `RUNNABLE` Infine, AWS Batch fornisce la capacità dell'istanza necessaria per eseguire il job e avvia queste istanze.

Ogni processo parallelo a più nodi contiene un nodo principale. Il nodo principale è una singola sottoattività che AWS Batch monitora per determinare l'esito del processo multinodo inviato. Il nodo principale viene avviato prima e passa allo stato `STARTING`. Il valore di timeout specificato nel `attemptDurationSeconds` parametro si applica all'intero processo e non ai nodi.

Quando il nodo principale raggiunge lo `RUNNING` stato dopo l'esecuzione del contenitore del nodo, i nodi secondari vengono avviati e anch'essi passano allo `STARTING` stato. I nodi figlio si presentano in ordine casuale. Non ci sono garanzie sui tempi e sull'ordine di avvio del nodo secondario. Per garantire che tutti i nodi dei job abbiano `RUNNING` lo stato dopo l'esecuzione del contenitore del nodo, il codice dell'applicazione può interrogare l' AWS Batch API per ottenere le informazioni sul nodo principale e sul nodo secondario. In alternativa, il codice dell'applicazione può attendere che tutti i nodi siano online prima di avviare qualsiasi attività di elaborazione distribuita. L'indirizzo IP privato del nodo principale è disponibile come la variabile d'ambiente `AWS_BATCH_JOB_MAIN_NODE_PRIVATE_IPV4_ADDRESS` in ogni nodo figlio. Il tuo codice dell'applicazione può utilizzare queste informazioni per coordinare e comunicare i dati tra ciascuna operazione.

Quando i nodi singoli escono, passano allo stato `SUCCEEDED` o `FAILED`, a seconda del loro codice di uscita. Se il nodo principale esce, il processo viene considerato completo e tutti i nodi figlio vengono arrestati. Se un nodo figlio muore, AWS Batch non esegue alcuna azione sugli altri nodi del job. Se non vuoi che il tuo lavoro continui con un numero ridotto di nodi, devi tenerne conto nel codice dell'applicazione. In questo modo si interrompe o si annulla il lavoro.

Considerazioni sull'ambiente di calcolo per MNP con AWS Batch

Durante la configurazione di ambienti di calcolo per eseguire processi paralleli a più nodi con AWS Batch, è necessario tenere presenti diversi aspetti.

- I lavori paralleli multinodo non sono supportati negli ambienti di `UNMANAGED` elaborazione.
- Se desideri inviare lavori paralleli multinodo a un ambiente di elaborazione, crea un gruppo di posizionamento del cluster in una singola zona di disponibilità e associalo alle tue risorse di elaborazione. In questo modo i processi paralleli multinodo su un raggruppamento logico di istanze restano vicini con un elevato potenziale di flusso di rete. Per ulteriori informazioni, consulta Placement Groups nella Amazon EC2 User Guide.
- I lavori paralleli multinodo non sono supportati negli ambienti di elaborazione che utilizzano istanze Spot.
- AWS Batch i lavori paralleli a più nodi utilizzano la modalità di `awsvpc` rete Amazon ECS, che offre ai contenitori di lavori paralleli multinodo le stesse proprietà di rete delle istanze Amazon EC2. Ogni container di processo parallelo a più nodi ottiene la propria interfaccia di rete elastica, un indirizzo IP primario privato e un nome host DNS interno. L'interfaccia di rete viene creata nella stessa sottorete VPC della risorsa di calcolo host.
- Il tuo ambiente di elaborazione potrebbe non avere più di cinque gruppi di sicurezza associati. Le interfacce di rete elastiche create e collegate a un'attività MNP utilizzeranno i gruppi di sicurezza specificati nell'ambiente di calcolo. Se non si specifica un gruppo di sicurezza, viene utilizzato il gruppo di sicurezza predefinito per il VPC.
- La modalità `awsvpc` di rete non fornisce le interfacce di rete elastiche per lavori paralleli a più nodi con indirizzi IP pubblici. Per accedere a Internet, le risorse di calcolo 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. La comunicazione tra nodi deve utilizzare l'indirizzo IP privato o il nome host DNS del nodo. I lavori paralleli multinodo eseguiti su risorse di calcolo all'interno di sottoreti pubbliche non hanno accesso alla rete in uscita. Per creare un VPC con sottoreti private e un gateway NAT, consulta Crea un cloud privato virtuale .
- Le interfacce di rete elastiche create e collegate alle risorse di calcolo non possono essere scollegate manualmente o modificate dal tuo account. Questo serve a prevenire l'eliminazione accidentale di un'interfaccia elastica di rete associata a un processo in esecuzione. Per rilasciare le interfacce di rete elastiche per un'attività, interrompere il processo.
- L'ambiente di elaborazione deve avere un valore massimo di v sufficiente CPUs per supportare il processo parallelo multinodo.

- La tua quota di EC2 istanze Amazon include il numero di istanze necessarie per eseguire il tuo processo. Ad esempio, supponiamo che il tuo processo richieda 30 istanze, ma che il tuo account possa eseguire solo 20 istanze in una regione. Quindi, il tuo lavoro rimarrà bloccato. `RUNNABLE`
- Se si specifica un tipo di istanza per un gruppo di nodi in un processo parallelo a più nodi, l'ambiente di calcolo deve avviare quel tipo di istanza.

Lavori paralleli multinodo su Amazon EKS

Puoi utilizzarlo AWS Batch su Amazon Elastic Kubernetes Service per eseguire processi paralleli a più nodi (MNP) (noti anche come pianificazione di gruppo) sui cluster gestiti. Kubernetes Questa opzione viene comunemente utilizzata per lavori di grandi dimensioni, strettamente accoppiati e ad alte prestazioni che non possono essere eseguiti su una singola istanza di Amazon Elastic Compute Cloud. Per ulteriori informazioni, consulta Lavori paralleli multinodo.

Puoi utilizzare questa funzionalità per eseguire applicazioni di elaborazione ad alte prestazioni gestite e Kubernetes specifiche di Amazon EKS, formazione su modelli linguistici di grandi dimensioni e altri lavori di intelligenza artificiale (AI) /machine learning (ML).

Argomenti

Esecuzione di processi MNP

AWS Batch supporta i lavori MNP su Amazon Elastic Container Service e Amazon EKS tramite Amazon EC2. Di seguito vengono fornite informazioni più specifiche sui parametri dell'istanza e del contenitore per la funzionalità.

Quote di istanze per MNP su Amazon EKS

- È possibile utilizzare fino a 1000 istanze per un singolo processo MNP.
- Fino a 5000 istanze possono unirsi a un singolo cluster Amazon EKS.
- È possibile raggruppare e collegare fino a 5 ambienti di elaborazione in cluster a una coda di lavoro.

Ad esempio, è possibile scalare fino a 5 ambienti di elaborazione in cluster in una coda di lavoro e 1000 istanze in ogni ambiente di elaborazione.

Oltre ai parametri dell'istanza, è importante notare che non è possibile utilizzare i lavori Fargate for MNP tramite nessuno dei due servizi.

È possibile utilizzare un solo tipo di istanza in ogni job MNP. È possibile modificare il tipo di istanza aggiornando l'ambiente di calcolo o definendo un nuovo ambiente di calcolo. È inoltre possibile specificare il tipo di istanza e fornire i requisiti di vCPU e memoria durante la creazione della definizione del processo.

Quote di container per MNP su Amazon EKS

- Un job parallelo multinodo supporta un pod per nodo.
- Fino a 10 contenitori (o 10 contenitori init). Per ulteriori informazioni, consulta Init Containers nella documentazione di Kubernetes.) in ogni pod.
- Fino a 5 intervalli di nodi in ogni job MNP.
- Fino a 10 immagini di contenitori distinte in ogni intervallo di nodi.

Ad esempio, è possibile eseguire fino a un massimo di 10.000 container in un singolo processo MNP che contiene 5 intervalli di nodi e un totale di 50 immagini uniche.

Esecuzione di processi MNP in un Amazon VPC privato e in un cluster Amazon EKS

I job MNP possono essere eseguiti su qualsiasi cluster Amazon EKS, indipendentemente dal fatto che disponga di una rete Internet pubblica o meno. Quando utilizzi un cluster Amazon EKS con solo accesso alla rete privata, assicurati che AWS Batch possa accedere al piano di controllo di Amazon EKS e al server Kubernetes API gestito. Puoi concedere l'accesso necessario tramite gli endpoint Amazon Virtual Private Cloud. Per ulteriori informazioni, consulta Configurare un servizio endpoint.

I cluster Pod Amazon EKS non possono scaricare un'immagine da una fonte pubblica poiché il VPC privato non dispone di accesso a Internet. Il tuo cluster Amazon EKS deve estrarre immagini da un registro di container che si trova all'interno del tuo Amazon VPC. Puoi creare un Amazon Elastic Container Registry (Amazon ECR) nel tuo Amazon VPC e copiarvi le immagini dei container per l'accesso ai nodi.

Puoi anche creare una regola pull through cache con Amazon ECR. Una volta creata una regola pull through cache per un registro pubblico esterno, puoi semplicemente estrarre un'immagine da quel registro pubblico esterno utilizzando l'URI del tuo registro privato Amazon ECR. Quindi Amazon ECR

crea un repository e memorizza l'immagine nella cache. Quando un'immagine memorizzata nella cache viene estratta utilizzando l'URI del registro privato di Amazon ECR, Amazon ECR controlla il registro remoto per verificare se esiste una nuova versione dell'immagine e aggiornerà il registro privato fino a una volta ogni 24 ore. Per ulteriori informazioni, consulta Creazione di una regola pull through cache in Amazon ECR.

Notifica di errore

Se i tuoi lavori MNP sono bloccati, puoi ricevere notifiche tramite Amazon Console di gestione AWS e Amazon EventBridge. Ad esempio, se un lavoro MNP è bloccato in testa alla coda, puoi ricevere una notifica sul problema insieme alle informazioni sulla sua causa, in modo da poter intervenire tempestivamente per sbloccare la coda dei lavori. Facoltativamente, è possibile terminare automaticamente il processo MNP se non viene intrapresa alcuna azione entro un determinato periodo di tempo, che può essere definito nel modello di coda dei lavori. Per ulteriori informazioni, consulta Eventi bloccati in Job queue

Crea una definizione di processo Amazon EKS MNP

Per definire ed eseguire processi MNP su Amazon EKS, sono disponibili nuovi parametri all'interno delle operazioni `RegisterJobDefinition`e delle `SubmitJob`API.

- Utilizzali nella `eksPropertiesnodeProperties`sezione per definire la definizione del processo MNP.
- Utilizza nella `eksPropertiesOverridenodePropertyOverrides`sezione per sovrascrivere i parametri definiti nella definizione del processo quando si invia un lavoro MNP.

Queste azioni possono essere definite tramite le operazioni API e il. Console di gestione AWS

Riferimento: registrare il payload della richiesta di definizione del processo Amazon EKS MNP

L'esempio seguente illustra come registrare una definizione di processo MNP di Amazon EKS con due nodi.

```
{
  "jobDefinitionName": "MyEksMnpJobDefinition",
  "type": "multinode",
  "nodeProperties": {
    "numNodes": 2,
```

```
    "mainNode": 0,
    "nodeRangeProperties": [
      {
        "targetNodes" : "0:",
        "eksProperties": {
          "podProperties": {
            "containers": [
              {
                "name": "test-eks-container-1",
                "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                "command": [
                  "sleep",
                  "60"
                ],
                "resources": {
                  "limits": {
                    "cpu": "1",
                    "memory": "1024Mi"
                  }
                },
                "securityContext":{
                  "runAsUser":1000,
                  "runAsGroup":3000,
                  "privileged":true,
                  "readOnlyRootFilesystem":true,
                  "runAsNonRoot":true
               }
              }
            ],
            "initContainers": [
               {
                  "name":"init-ekscontainer",
                  "image": "public.ecr.aws/amazonlinux/amazonlinux:2",
                  "command": [
                     "echo",
                     "helloWorld"
                   ],
                   "resources": {
                     "limits": {
                       "cpu": "1",
                       "memory": "1024Mi"
                     }
                  }
               }
```

```
            ],
            "metadata": {
               "labels": {
                  "environment" : "test"
               }
            }
          }
        }
      }
    ]
  }
}
```

Per registrare la definizione del processo utilizzando AWS CLI, copia la definizione in un file locale denominato MyEksMnpJobDefinition.json ed esegui il comando seguente.

```
aws batch register-job-definition --cli-input-json file://MyEksMnpJobDefinition.json
```

Riceverai la seguente risposta JSON.

```
{
    "jobDefinitionName": "MyEksMnpJobDefinition",
    "jobDefinitionArn": "arn:aws:batch:us-east-1:0123456789:job-definition/
MyEksMnpJobDefinition:1",
    "revision": 1
}
```

Invia un lavoro Amazon EKS MNP

Per inviare un lavoro utilizzando la definizione di lavoro registrata, inserisci il seguente comando. Sostituisci il valore di <EKS_JOB_QUEUE_NAME>con il nome o l'ARN di una coda di lavoro preesistente associata a un ambiente di calcolo Amazon EKS.

```
aws batch submit-job --job-queue <EKS_JOB_QUEUE_NAME> \
    --job-definition MyEksMnpJobDefinition \
    --job-name myFirstEksMnpJob
```

Riceverai la seguente risposta JSON.

```
{
```

```
    "jobArn": "arn:aws:batch:region:account:job/9b979cce-9da0-446d-90e2-ffa16d52af68",
    "jobName": "myFirstEksMnpJob",
    "jobId": "<JOB_ID>"
}
```

È possibile controllare lo stato del lavoro utilizzando il JoBid restituito con il seguente comando.

```
aws batch describe-jobs --jobs <JOB_ID>
```

Sostituisci una definizione di processo MNP di Amazon EKS

Facoltativamente, puoi sovrascrivere i dettagli della definizione del lavoro (ad esempio modificare le dimensioni del lavoro MNP o i dettagli del lavoro secondario). Di seguito viene fornito un esempio di payload di richiesta JSON per inviare un job MNP a cinque nodi e modifiche al comando del contenitore. `test-eks-container-1`

```
{
  "numNodes": 5,
  "nodePropertyOverrides": [
    {
      "targetNodes": "0:",
      "eksPropertiesOverride": {
        "podProperties": {
          "containers": [
            {
              "name": "test-eks-container-1",
              "command": [
                "sleep",
                "150"
              ]
            }
          ]
        }
      }
    }
  ]
}
```

Per inviare un lavoro con queste sostituzioni, salvate l'esempio in un file locale, eks-mnp-job-nodeoverride.json, e utilizzate il per inviare il lavoro con AWS CLI le sostituzioni.

Lavori di matrice

Un array job è un job che condivide parametri comuni, come la definizione del jobCPUs, v e la memoria. Viene eseguito come una raccolta di processi di base correlati ma separati che potrebbero essere distribuiti su più host ed eseguiti contemporaneamente. I job di array sono il modo più efficiente per eseguire lavori estremamente paralleli come simulazioni Monte Carlo, sweep parametrici o lavori di rendering di grandi dimensioni.

AWS Batch i lavori di array vengono inviati proprio come i normali lavori. Tuttavia, devi specificare la dimensione dell'array (tra 2 e 10.000) per definire la quantità di processi figlio da eseguire nell'array. Se invii un processo con una dimensione dell'array di 1.000, un singolo processo viene eseguito e genera 1.000 processi figlio. Il processo in array è un riferimento o un puntatore per gestire tutti i processi figlio. In questo modo, puoi inviare carichi di lavoro di grandi dimensioni con una singola query. Il timeout specificato nel `attemptDurationSeconds` parametro si applica a ogni lavoro secondario. Il job dell'array principale non ha un timeout.

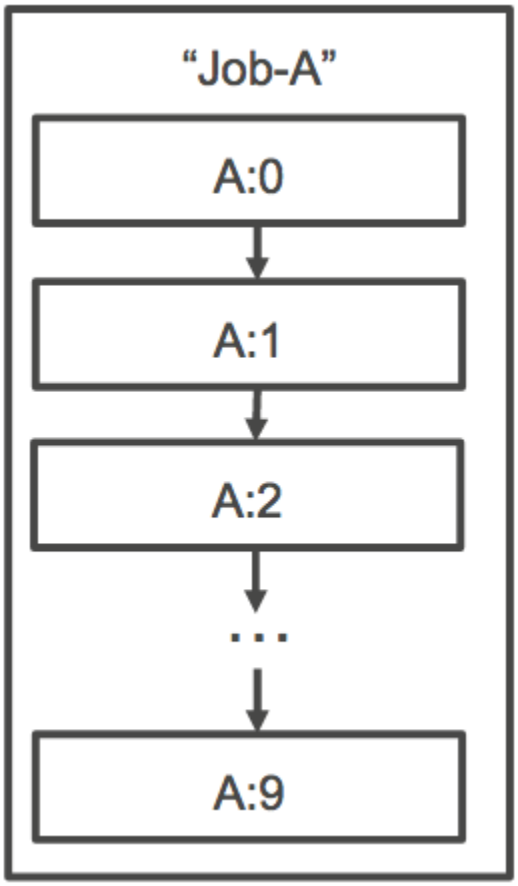
Quando si invia un processo di array, il job dell'array principale ottiene un ID di AWS Batch lavoro normale. Ogni job secondario ha lo stesso ID di base. Tuttavia, l'indice dell'array per il job secondario viene aggiunto alla fine dell'ID principale, ad esempio *`example_job_ID`*`:0` per il primo job figlio dell'array.

Il processo dell'array principale può immettere uno `SUCCEEDED` stato `SUBMITTED` `PENDINGFAILED`,,. Un processo principale dell'array viene aggiornato a `PENDING` quando viene aggiornato un processo figlio a`RUNNABLE`. Per ulteriori informazioni sulle dipendenze lavorative, vedereDipendenze dal lavoro.

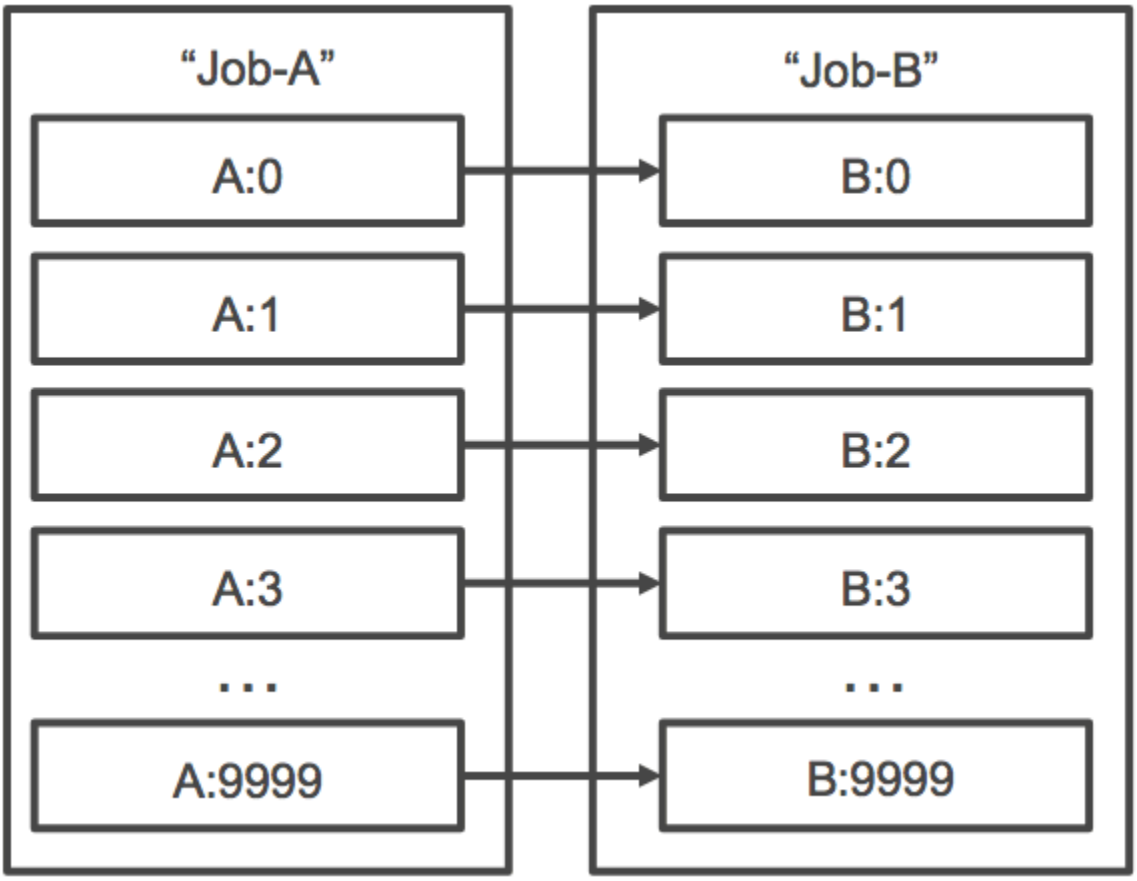
In fase di runtime, la variabile di ambiente `AWS_BATCH_JOB_ARRAY_INDEX` è impostata sul numero di indice dell'array del processo corrispondente del container. Il primo array job index è numerato e `0` i tentativi successivi sono in ordine crescente (ad esempio, 1, 2 e 3). È possibile utilizzare questo valore di indice per controllare il modo in cui vengono indicati i diversi elementi figlio dei processi in array. Per ulteriori informazioni, consulta Utilizza l'array job index per controllare la differenziazione dei job.

Per le dipendenze di processi in array, puoi specificare un tipo di dipendenza, ad esempio `SEQUENTIAL` o `N_TO_N`. Puoi specificare una dipendenza di tipo `SEQUENTIAL` (senza specificare un ID del processo) in modo che ogni processo figlio nell'array venga completato in maniera sequenziale a partire dall'indice 0. Ad esempio, se invii un processo in array con una dimensione dell'array di 100 e specifichi una dipendenza di tipo `SEQUENTIAL`, 100 processi figlio vengono generati in sequenza,

il primo dei quali deve avere esito positivo prima che il successivo processo figlio possa iniziare. L'illustrazione di seguito mostra il processo A, un processo in array con una dimensione dell'array di 10. Ciascun processo nell'indice dei figli del processo A dipende dal processo figlio precedente. Il processo A:1 non può iniziare fino a quando il processo A:0 non termina.



Puoi anche specificare una dipendenza tipo `N_TO_N` con un ID processo per processi dell'array. In questo modo, prima di iniziare, ciascun figlio nell'indice di questo processo deve attendere il completamento del figlio nell'indice corrispondente di ciascuna dipendenza. La figura seguente mostra Job A e Job B, due job di array con una dimensione di array di 10.000 ciascuno. Ciascun processo nell'indice dei figli del processo B dipende dal corrispondente indice nel processo A. Il processo B:1 non può iniziare fino al completamento del processo A:1.



Se si annulla o si termina un processo di array principale, tutti i job secondari vengono annullati o terminati con esso. È possibile annullare o interrompere singoli lavori per bambini (il che li sposta a uno `FAILED` status) senza influire sugli altri lavori secondari. Tuttavia, se un job child array fallisce (da solo o annullando o interrompendo manualmente il lavoro), anche il lavoro principale fallisce. In questo scenario, il lavoro principale passa al completamento di tutti `FAILED` i lavori secondari.

Argomenti

- Esempio di workflow Array Job
- Utilizza l'array job index per controllare la differenziazione dei job

Esempio di workflow Array Job

Un flusso di lavoro comune per AWS Batch i clienti consiste nell'eseguire un processo di configurazione con prerequisiti, eseguire una serie di comandi su un gran numero di attività di input e quindi concludere con un processo che aggrega i risultati e scrive dati di riepilogo su Amazon S3, DynamoDB, Amazon Redshift o Aurora.

Esempio:

- `JobA`: un processo standard, non basato su array, che esegue una rapida elencazione e convalida dei metadati degli oggetti in un bucket Amazon S3,. `BucketA` La sintassi SubmitJobJSON è la seguente.

```
{
    "jobName": "JobA",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobA-list-and-validate:1"
}
```

- `JobB`: Un lavoro di matrice con 10.000 copie da cui dipende che esegue comandi `JobA` che richiedono un uso intensivo della CPU su ogni oggetto in `BucketA` ingresso e carica i risultati su. `BucketB` La sintassi SubmitJobJSON è la seguente.

```
{
    "jobName": "JobB",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobB-CPU-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "4096"
            },
            {
                "type": "VCPU",
                "value": "32"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobA_job_ID"
  }
    ]
}
```

- `JobC`: Un altro processo di array di 10.000 copie che dipende `JobB` da un modello di `N_TO_N` dipendenza, che esegue comandi che richiedono molta memoria per ogni elemento in essoBucketB, scrive i metadati su DynamoDB e carica l'output risultante su. `BucketC` La sintassi JSON è la seguente. SubmitJob

```
{
    "jobName": "JobC",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobC-Memory-Intensive-Processing:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10000
    },
    "dependsOn": [
        {
            "jobId": "JobB_job_ID",
            "type": "N_TO_N"
        }
    ]
}
```

- `JobD`: Un processo di array che esegue 10 passaggi di convalida, ciascuno dei quali richiede una query su DynamoDB e che può interagire con uno qualsiasi dei bucket Amazon S3 di cui sopra. Ciascuno dei passaggi della procedura esegue lo stesso comando`JobD`. Tuttavia, il comportamento è diverso in base al valore della variabile di `AWS_BATCH_JOB_ARRAY_INDEX` ambiente all'interno del contenitore del lavoro. Questi passaggi di convalida vengono eseguiti in sequenza (ad esempio, `JobD:0` e poi`JobD:1`). La sintassi SubmitJobJSON è la seguente.

```
{
    "jobName": "JobD",
    "jobQueue": "ProdQueue",
```

```
    "jobDefinition": "JobD-Sequential-Validation:1",
    "containerOverrides": {
        "resourceRequirements": [
            {
                "type": "MEMORY",
                "value": "32768"
            },
            {
                "type": "VCPU",
                "value": "1"
            }
        ]
   }
    "arrayProperties": {
        "size": 10
    },
    "dependsOn": [
        {
            "jobId": "JobC_job_ID"
        },
        {
            "type": "SEQUENTIAL"
        },

    ]
}
```

- JobE: un processo finale, non basato su array, che esegue alcune semplici operazioni di pulizia e invia una notifica Amazon SNS con un messaggio che la pipeline è stata completata e un collegamento all'URL di output. La sintassi SubmitJobJSON è la seguente.

```
{
    "jobName": "JobE",
    "jobQueue": "ProdQueue",
    "jobDefinition": "JobE-Cleanup-and-Notification:1",
    "parameters": {
        "SourceBucket": "s3://amzn-s3-demo-source-bucket",
        "Recipient": "pipeline-notifications@mycompany.com"
    },
    "dependsOn": [
        {
            "jobId": "JobD_job_ID"
        }
```

```
    ]
}
```

Utilizza l'array job index per controllare la differenziazione dei job

Questo tutorial descrive come utilizzare la variabile di `AWS_BATCH_JOB_ARRAY_INDEX` ambiente per differenziare i lavori dei bambini. Ogni lavoro secondario viene assegnato a questa variabile. L'esempio utilizza il numero di indice del lavoro secondario per leggere una riga specifica in un file. Quindi, sostituisce il parametro associato a quel numero di riga con un comando all'interno del contenitore del lavoro. Il risultato è che puoi avere più AWS Batch job che eseguono la stessa immagine Docker e gli stessi argomenti di comando. Tuttavia, i risultati sono diversi perché l'array job index viene utilizzato come modificatore.

In questo tutorial creerai un file di testo contenente tutti i colori dell'arcobaleno, ciascuno su una riga. Quindi, si crea uno script di ingresso per un contenitore Docker che converte l'indice in un valore che può essere utilizzato per un numero di riga nel file a colori. L'indice inizia da zero, ma i numeri di riga iniziano da uno. Crea un Dockerfile che copia i file di colore e indice nell'immagine del contenitore e imposta `ENTRYPOINT` l'immagine nello script di ingresso. Il Dockerfile e le risorse sono creati su un'immagine Docker che viene inviata ad Amazon ECR. Quindi registri una definizione di processo che utilizzi la tua nuova immagine del contenitore, invii un AWS Batch array job con quella definizione di processo e visualizzi i risultati.

Argomenti

Prerequisiti

Questo flusso di lavoro tutorial presenta i seguenti prerequisiti:

- Un ambiente AWS Batch di calcolo. Per ulteriori informazioni, consulta Crea un ambiente di elaborazione.

- Una coda AWS Batch di lavoro e l'ambiente di elaborazione associato. Per ulteriori informazioni, consulta Creare una coda di lavoro.
- È AWS CLI installato sul sistema locale. Per ulteriori informazioni, vedere >Installazione di AWS Command Line Interface nella Guida per l'AWS Command Line Interface utente.
- Il Docker installato sul sistema locale. Per ulteriori informazioni, consulta la sezione relativa al Docker CE nella documentazione del Docker.

Crea un'immagine del contenitore

È possibile utilizzarlo `AWS_BATCH_JOB_ARRAY_INDEX` in una definizione di processo nel parametro di comando. Tuttavia, si consiglia di creare un'immagine del contenitore che utilizzi invece la variabile in uno script entrypoint. In questa sezione viene descritto come creare un'immagine del container di questo tipo.

Per creare l'immagine del container Docker

1. Crea una nuova directory da utilizzare come WorkSpace dell'immagine Docker e aprila.
2. Create un file denominato `colors.txt` nella directory del workspace e incollate quanto segue al suo interno.

```
red
orange
yellow
green
blue
indigo
violet
```

3. Crea un file denominato `print-color.sh` nella cartella del tuo spazio di lavoro e incolla quanto segue al suo interno.

> **Note**
>
> La variabile `LINE` è impostata su `AWS_BATCH_JOB_ARRAY_INDEX` +1 poiché l'indice dell'array inizia da 0, mentre i numeri di riga iniziano da 1. La `COLOR` variabile è impostata sul colore associato al numero di riga. `colors.txt`

```
#!/bin/sh
LINE=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
COLOR=$(sed -n ${LINE}p /tmp/colors.txt)
echo My favorite color of the rainbow is $COLOR.
```

4. Crea un file denominato `Dockerfile` nella cartella del tuo spazio di lavoro e incolla il seguente contenuto al suo interno. Questo file copia i file precedenti nel container e imposta lo script entrypoint in modo che venga eseguito all'avvio del container.

```
FROM busybox
COPY print-color.sh /tmp/print-color.sh
COPY colors.txt /tmp/colors.txt
RUN chmod +x /tmp/print-color.sh
ENTRYPOINT /tmp/print-color.sh
```

5. Creazione dell'immagine Docker.

```
$ docker build -t print-color .
```

6. Esegui il test del container utilizzando lo script riportato di seguito. Questo script imposta la `AWS_BATCH_JOB_ARRAY_INDEX` variabile su 0 localmente e poi la incrementa per simulare il funzionamento di un array job con sette figli.

```
$ AWS_BATCH_JOB_ARRAY_INDEX=0
while [ $AWS_BATCH_JOB_ARRAY_INDEX -le 6 ]
do
    docker run -e AWS_BATCH_JOB_ARRAY_INDEX=$AWS_BATCH_JOB_ARRAY_INDEX print-color
    AWS_BATCH_JOB_ARRAY_INDEX=$((AWS_BATCH_JOB_ARRAY_INDEX + 1))
done
```

Di seguito è riportato l'output.

```
My favorite color of the rainbow is red.
My favorite color of the rainbow is orange.
My favorite color of the rainbow is yellow.
My favorite color of the rainbow is green.
My favorite color of the rainbow is blue.
My favorite color of the rainbow is indigo.
My favorite color of the rainbow is violet.
```

Invia la tua immagine ad Amazon ECR

Ora che hai creato e testato il tuo contenitore Docker, inseriscilo in un archivio di immagini. Questo esempio utilizza Amazon ECR, ma è possibile utilizzare un altro registro, ad esempio DockerHub.

1. Crea un archivio di immagini Amazon ECR per archiviare l'immagine del contenitore. Questo esempio utilizza solo il AWS CLI, ma puoi anche usare il. Console di gestione AWS Per ulteriori informazioni, consulta Creating a repository nella Amazon Elastic Container Registry User Guide.

   ```
   $ aws ecr create-repository --repository-name print-color
   ```

2. Etichetta l'`print-color`immagine con l'URI del repository Amazon ECR restituito dal passaggio precedente.

   ```
   $ docker tag print-color aws_account_id.dkr.ecr.region.amazonaws.com/print-color
   ```

3. Accedi al tuo registro Amazon ECR. 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
   ```

4. Invia la tua immagine ad Amazon ECR.

   ```
   $ docker push aws_account_id.dkr.ecr.region.amazonaws.com/print-color
   ```

Crea e registra una definizione di lavoro

Ora che l'immagine Docker si trova in un registro di immagini, puoi specificarla in una definizione di AWS Batch lavoro. Quindi, puoi utilizzarlo in un secondo momento per eseguire un processo di array. In questo esempio viene utilizzato solo il AWS CLI. Tuttavia, è possibile utilizzare anche il Console di gestione AWS. Per ulteriori informazioni, consulta Creare una definizione di processo a nodo singolo .

Per creare una definizione del processo

1. Crea un file denominato `print-color-job-def.json` nella cartella del tuo spazio di lavoro e incolla quanto segue al suo interno. Sostituisci l'URI del repository di immagini con l'URI dell'immagine personalizzata.

```
{
  "jobDefinitionName": "print-color",
  "type": "container",
  "containerProperties": {
    "image": "aws_account_id.dkr.ecr.region.amazonaws.com/print-color",
    "resourceRequirements": [
        {
            "type": "MEMORY",
            "value": "250"
        },
        {
            "type": "VCPU",
            "value": "1"
        }
    ]
  }
}
```

2. Registra la definizione del lavoro con AWS Batch.

```
$ aws batch register-job-definition --cli-input-json file://print-color-job-
def.json
```

Inviate un lavoro AWS Batch di matrice

Dopo aver registrato la definizione del processo, è possibile inviare un lavoro di AWS Batch array che utilizzi la nuova immagine del contenitore.

Per inviare un lavoro AWS Batch di array

1. Crea un file denominato `print-color-job.json` nella cartella del tuo spazio di lavoro e incolla quanto segue al suo interno.

> **Note**
>
> Questo esempio utilizza la coda dei lavori menzionata nella the section called "Prerequisiti" sezione.

```
{
  "jobName": "print-color",
  "jobQueue": "existing-job-queue",
  "arrayProperties": {
    "size": 7
  },
  "jobDefinition": "print-color"
}
```

2. Invia il lavoro alla tua coda di AWS Batch lavoro. Annota l'ID del lavoro restituito nell'output.

```
$ aws batch submit-job --cli-input-json file://print-color-job.json
```

3. Descrivi lo stato del processo e attendi che il processo venga impostato su `SUCCEEDED`.

Visualizza i log dei lavori dell'array

Una volta raggiunto `SUCCEEDED` lo stato del lavoro, puoi visualizzare CloudWatch i log dal contenitore del lavoro.

Per visualizzare i log del lavoro in Logs CloudWatch

1. Apri la AWS Batch console all'indirizzo. https://console.aws.amazon.com/batch/
2. Nel riquadro di navigazione a sinistra, scegli Jobs (Processi).
3. In Job queue (Coda di processi), seleziona una coda.
4. Nella sezione Status (Stato), scegli succeeded (completato).
5. Per visualizzare tutti i processi figlio del processo in array, seleziona l'ID processo restituito nella sezione precedente.
6. Per visualizzare i log del container del processo, seleziona uno dei processi figlio e scegli View logs (Visualizza log).

7. Visualizza gli altri log del processo figlio. Ciascun job restituisce un colore diverso dell'arcobaleno.

Esegui lavori GPU

I job GPU consentono di eseguire lavori che utilizzano un'istanza. GPUs

Sono supportati i seguenti tipi di istanze EC2 basate su Amazon GPU. Per ulteriori informazioni, consulta Istanze Amazon EC2 G3, Istanze Amazon EC2 G4, Istanze Amazon G5, Istanze EC2 AmazonG6, Istanze EC2 Amazon P2, Istanze Amazon EC2 P3, Istanze EC2 Amazon EC2 P4d, Istanze Amazon P5, IstanzeAmazon P6, Istanze Amazon Trn1, IstanzeAmazon EC2 Trn2, Amazon EC2 EC2 EC2 https://aws.amazon.com/ec2/instance-types/inf1/ EC2 Istanze Inf1, istanze Amazon EC2 Inf2, istanzeAmazon Dl1 e istanze EC2 Amazon Dl2. EC2

Tipo di istanza	GPUs	Memoria GPU	v CPUs	Memoria	Larghezza di banda di rete
g3s.xlarge	1	8 GiB	4	30,5 GiB	10 Gb/s
g3.4xlarge	1	8 GiB	16	122 GiB	Fino a 10 Gb/s
g3.8xlarge	2	16 GiB	32	24 GiB	10 Gb/s
g3.16xlarge	4	32 GiB	64	488 GiB	25 Gb/s
g4dn.xlarge	1	16 GiB	4	16 GiB	Fino a 25 Gb/s
g4dn.2xlarge	1	16 GiB	8	32 GiB	Fino a 25 Gb/s
g4dn.4xlarge	1	16 GiB	16	64 GiB	Fino a 25 Gb/s

Tipo di istanza	GPUs	Memoria GPU	v CPUs	Memoria	Larghezza di banda di rete
g4dn.8xlarge	1	16 GiB	32	128 GiB	50 Gb/s
g4dn.12xlarge	4	64 GiB	48	192 GiB	50 Gb/s
g4dn.16xlarge	1	16 GiB	64	256 GiB	50 Gb/s
g5.xlarge	1	24 GiB	4	16 GiB	Fino a 10 Gb/s
g5.2xlarge	1	24 GiB	8	32 GiB	Fino a 10 Gb/s
g5.4xlarge	1	24 GiB	16	64 GiB	Fino a 25 Gb/s
g5.8xlarge	1	24 GiB	32	128 GiB	25 Gb/s
g5.16xlarge	1	24 GiB	64	256 GiB	25 Gb/s
g5.12xlarge	4	96 GiB	48	192 GiB	40 Gb/s
g5.24xlarge	4	96 GiB	96	384 GiB	50 Gb/s
g5.48xlarge	8	192 GiB	192	768 GiB	100 Gb/s
g5g.xlarge	1	16 GiB	4	8 GiB	Fino a 10 Gb/s
g5g.2xlarge	1	16 GiB	8	16 GiB	Fino a 10 Gb/s
g5g.4xlarge	1	16 GiB	16	32 GiB	Fino a 10 Gb/s
g5g.8xlarge	1	16 GiB	32	64 GiB	12 GBps
g5g.16xlarge	2	32 GiB	64	128 GiB	25 Gb/s
g5g.metal	2	32 GiB	64	128 GiB	25 Gb/s
g6.xlarge	1	24 GiB	4	16 GiB	Fino a 10 Gb/s
g6.2xlarge	1	24 GiB	8	32 GiB	Fino a 10 Gb/s
g6.4xlarge	1	24 GiB	16	64 GiB	Fino a 25 Gb/s

Tipo di istanza	GPUs	Memoria GPU	v CPUs	Memoria	Larghezza di banda di rete
g6.8xlarge	1	24 GiB	32	128 GiB	25 Gb/s
g6.16xlarge	1	24 GiB	64	256 GiB	25 Gb/s
g6.12xlarge	4	96 GiB	48	192 GiB	40 Gb/s
g6.24xlarge	4	96 GiB	96	384 GiB	50 Gb/s
g6.48xlarge	8	192 GiB	192	768 GiB	100 Gb/s
g6e.xlarge	1	48 GiB	4	32 GiB	Fino a 20 Gbps
g6e.2xlarge	1	48 GiB	8	64 GiB	Fino a 20 Gbps
g6e.4xlarge	1	48 GiB	16	128 GiB	20 Gb/s
g6e.8xlarge	1	48 GiB	32	256 GiB	25 Gb/s
g6e.16xlarge	1	48 GiB	64	512 GiB	35 Gbps
g6e.12xlarge	4	192 GiB	48	384 GiB	100 Gb/s
g6e.24xlarge	4	192 GiB	96	768 GiB	200 Gb/s
g6e.48xlarge	8	384 GiB	192	1536 GiB	400 Gb/s
gr6.4xlarge	1	24 GiB	16	128 GiB	Fino a 25 Gb/s
gr6.8xlarge	1	24 GiB	32	256 GiB	25 Gb/s
p2.xlarge	1	12 GiB	4	61 GiB	Elevata
p2.8xlarge	8	96 GiB	32	488 GiB	10 Gb/s
p2.16xlarge	16	192 GiB	64	732 GiB	20 Gb/s
p3.2xlarge	1	16 GiB	8	61 GiB	Fino a 10 Gb/s
p3.8xlarge	4	64 GiB	32	24 GiB	10 Gb/s

Tipo di istanza	GPUs	Memoria GPU	v CPUs	Memoria	Larghezza di banda di rete
p3.16xlarge	8	128 GiB	64	488 GiB	25 Gb/s
p3dn.24xlarge	8	256 GiB	96	768 GiB	100 Gb/s
p4d.24xlarge	8	320 GiB	96	1152 GiB	400 Gb/s
p4de.24xlarge	8	640 GiB	96	1152 GiB	400 Gb/s
p5.48xlarge	8	640 GiB	192	2 TiB	3200 Gbps
p5e.48xlarge	8	128 GiB	192	2 TiB	3200 Gbps
p5en.48xlarge	8	128 GiB	192	2 TiB	3200 Gbps
p6-b200.48xlarge	8	1440 GiB	192	2 TiB	100 Gb/s
trn1.2xlarge	1	32 GiB	8	32 GiB	Fino a 12,5 Gb/s
trn1.32xlarge	16	512 GiB	128	512 GiB	800 Gbps
trn1n.32xlarge	16	512 GiB	128	512 GiB	1600 Gbps
trn2.48xlarge	16	1,5 TiB	192	2 TiB	3,2 Tbps
inf1.xlarge	1	8 GiB	4	8 GiB	Fino a 25 Gb/s
inf1.2xlarge	1	8 GiB	8	16 GiB	Fino a 25 Gb/s
inf1.6xlarge	4	32 GiB	24	48 GiB	25 Gb/s
inf1.24xlarge	16	128 GiB	96	192 GiB	100 Gb/s
inf2.xlarge	1	32 GiB	4	16 GiB	Fino a 15 Gb/s
inf2.8xlarge	1	32 GiB	32	128 GiB	Fino a 25 Gb/s
inf2.24xlarge	6	192 GiB	96	384 GiB	50 Gb/s
inf2.48xlarge	12	384 GiB	192	768 GiB	100 Gb/s

Tipo di istanza	GPUs	Memoria GPU	v CPUs	Memoria	Larghezza di banda di rete
dl1.24xlarge	8	256 GiB	96	768 GiB	400 Gb/s
dl2q.24xlarge	8	128 GiB	96	768 GiB	100 Gb/s

> **Note**
>
> Per i lavori con GPU supporta AWS Batch solo i tipi di istanze con NVIDIA. GPUs Ad esempio, la `G4ad`famiglia non è supportata per la pianificazione tramite GPU. Puoi comunque utilizzare `G4ad`on AWS Batch definendo solo i requisiti di vcpu e memoria nella definizione del processo, quindi accedendo GPUs direttamente all'host tramite la personalizzazione in un modello di Amazon EC2 Launch, i dati utente con un'AMI ottimizzata per il calcolo Amazon ECS o Amazon EKS o un'AMI personalizzata per l'utilizzo di AMD. GPUs
> I tipi di istanza che utilizzano un' ARM64 architettura sono supportati per i lavori GPU su dati AMIs forniti su misura AWS Batch o su dati EC2 utente Amazon a cui accedere GPUs tramite codice e configurazioni personalizzati. Ad esempio, la famiglia di `G5g`istanze.

Il parametro ResourceRequirements per la definizione del processo specifica il numero di GPUs da aggiungere al contenitore. Questo numero di non GPUs è disponibile per nessun altro processo eseguito su quell'istanza per la durata del processo. Tutti i tipi di istanze in un ambiente di calcolo che esegue processi GPU devono appartenere alle `p3` famiglie di `g6` istanze `p4` `p5p6,g3,g3s,g4,,g5`, o. Se ciò non viene fatto, un job GPU potrebbe rimanere bloccato nello stato. `RUNNABLE`

I lavori che non utilizzano il GPUs possono essere eseguiti su istanze GPU. Tuttavia, l'esecuzione su istanze GPU potrebbe costare di più rispetto a istanze simili non GPU. A seconda della vCPU, della memoria e del tempo specifici necessari, questi job non GPU potrebbero bloccare l'esecuzione dei job GPU.

Argomenti

- Crea un Kubernetes cluster basato su GPU su Amazon EKS
- Crea una definizione di processo GPU Amazon EKS
- Esegui un job GPU nel tuo cluster Amazon EKS

Crea un Kubernetes cluster basato su GPU su Amazon EKS

Prima di creare un Kubernetes cluster basato su GPU su Amazon EKS, devi aver completato i passaggi indicati. Guida introduttiva ad AWS Batch Amazon EKS Inoltre, considera anche quanto segue:

- AWS Batch supporta i tipi di istanze con NVIDIA GPUs.
- Per impostazione predefinita, AWS Batch seleziona l'AMI accelerata Amazon EKS con la Kubernetes versione che corrisponde alla versione del piano di controllo del cluster Amazon EKS.

```
$ cat <<EOF > ./batch-eks-gpu-ce.json
{
  "computeEnvironmentName": "My-Eks-GPU-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:<account>:cluster/<cluster-name>",
    "kubernetesNamespace": "my-aws-batch-namespace"
  },
  "computeResources": {
    "type": "EC2",
    "allocationStrategy": "BEST_FIT_PROGRESSIVE",
    "minvCpus": 0,
    "maxvCpus": 1024,
    "instanceTypes": [
      "p3dn.24xlarge",
      "p4d.24xlarge"
    ],
    "subnets": [
        "<eks-cluster-subnets-with-access-to-internet-for-image-pull>"
    ],
    "securityGroupIds": [
        "<eks-cluster-sg>"
    ],
    "instanceRole": "<eks-instance-profile>"
  }
}
EOF

$ aws batch create-compute-environment --cli-input-json file://./batch-eks-gpu-ce.json
```

AWS Batch non gestisce il plug-in del dispositivo NVIDIA GPU per tuo conto. È necessario installare questo plug-in nel cluster Amazon EKS e consentirgli di indirizzare i AWS Batch nodi. Per ulteriori informazioni, consulta Enabling GPU Support in Kubernetes on GitHub.

Per configurare il plugin del NVIDIA dispositivo (`DaemonSet`) per indirizzare i AWS Batch nodi, esegui i seguenti comandi.

```
# pull nvidia daemonset spec
$ curl -O https://raw.githubusercontent.com/NVIDIA/k8s-device-plugin/v0.12.2/nvidia-device-plugin.yml
# using your favorite editor, add Batch node toleration
# this will allow the DaemonSet to run on Batch nodes
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"

$ kubectl apply -f nvidia-device-plugin.yml
```

Non è consigliabile combinare carichi di lavoro basati su calcolo (CPU e memoria) con carichi di lavoro basati su GPU nelle stesse combinazioni di ambiente di calcolo e coda di lavoro. Questo perché i processi di elaborazione possono utilizzare la capacità della GPU.

Per allegare code di lavoro, esegui i seguenti comandi.

```
$ cat <<EOF > ./batch-eks-gpu-jq.json
 {
    "jobQueueName": "My-Eks-GPU-JQ1",
    "priority": 10,
    "computeEnvironmentOrder": [
      {
        "order": 1,
        "computeEnvironment": "My-Eks-GPU-CE1"
      }
    ]
  }
EOF

$ aws batch create-job-queue --cli-input-json file://./batch-eks-gpu-jq.json
```

Crea una definizione di processo GPU Amazon EKS

Al momento `nvidia.com/gpu` è supportata solo questa opzione e il valore della risorsa impostato deve essere un numero intero. Non puoi usare frazioni di GPU. Per ulteriori informazioni, consulta Schedule GPUs nella Kubernetes documentazione.

Per registrare una definizione di processo GPU per Amazon EKS, esegui i seguenti comandi.

```
$ cat <<EOF > ./batch-eks-gpu-jd.json
{
    "jobDefinitionName": "MyGPUJobOnEks_Smi",
    "type": "container",
    "eksProperties": {
        "podProperties": {
            "hostNetwork": true,
            "containers": [
                {
                    "image": "nvcr.io/nvidia/cuda:10.2-runtime-centos7",
                    "command": ["nvidia-smi"],
                    "resources": {
                        "limits": {
                            "cpu": "1",
                            "memory": "1024Mi",
                            "nvidia.com/gpu": "1"
                        }
                    }
                }
            ]
        }
    }
}
EOF

$ aws batch register-job-definition --cli-input-json file://./batch-eks-gpu-jd.json
```

Esegui un job GPU nel tuo cluster Amazon EKS

La risorsa GPU non è comprimibile. AWS Batch crea una specifica del pod per i lavori GPU in cui il valore della richiesta è uguale al valore dei limiti. Questo è un requisito. Kubernetes

Per inviare un job GPU, esegui i seguenti comandi.

```
$ aws batch submit-job --job-queue My-Eks-GPU-JQ1 --job-definition MyGPUJobOnEks_Smi --
job-name My-Eks-GPU-Job

# locate information that can help debug or find logs (if using Amazon CloudWatch Logs
 with Fluent Bit)
$ aws batch describe-jobs --job <job-id> | jq '.jobs[].eksProperties.podProperties |
 {podName, nodeName}'
{
  "podName": "aws-batch.f3d697c4-3bb5-3955-aa6c-977fcf1cb0ca",
  "nodeName": "ip-192-168-59-101.ec2.internal"
}
```

Visualizza i AWS Batch lavori in una coda di lavoro

Puoi visualizzare e filtrare i tuoi lavori in AWS Batch. Questa funzione offre la possibilità di visualizzare una coda di lavori esistente e filtrarne i lavori in base a una delle tre opzioni.

La ricerca e il filtro sono in grado di recuperare lavori che non si trovano in uno stato terminale (SUCCEEDEDoFAILED). Una volta che lo stato di un lavoro è SUCCEEDED o FAILED dovresti essere in grado di recuperarlo per un massimo di sette giorni. Puoi comunque visualizzare i EventBridge log di un'offerta di lavoro CloudWatch o di Amazon.

Utilizza questa procedura per elencare tutti i lavori in una coda di lavori nella AWS Batch console. Facoltativamente, utilizzare il campo Filtra risultati per restringere i risultati in base ai criteri specificati.

1. Passare alla console AWS Batch.
2. Nel riquadro di navigazione, scegli Jobs.
3. Espandi l'elenco a discesa Job queue e scegli la coda dei lavori in cui desideri cercare.

   > **Note**
   >
   > Puoi cercare offerte di lavoro all'interno di una sola coda di lavoro alla volta.

4. Nel campo Filtra risultati, inserisci le parole chiave da includere nei risultati della ricerca. È possibile utilizzare questo campo per cercare per nome del lavoro, stato o ID lavoro. A seconda della proprietà, è possibile che sia necessario definire operatori aggiuntivi, ad esempio equals (=) o contains (:).

> **Note**
>
> SageMaker Le code dei lavori di formazione supportano solo il filtraggio in base al nome del lavoro e al Job ID

5. Selezionare Search (Cerca).

AWS Batch Cerca offerte di lavoro in una coda di lavoro

Puoi cercare e filtrare i tuoi lavori AWS Batch utilizzando Job search. Questa funzione offre la possibilità di cercare all'interno di una coda di lavoro esistente e filtrarne i lavori.

La ricerca e il filtro sono in grado di recuperare lavori che non si trovano in uno stato terminale (`SUCCEEDEDoFAILED`). Una volta che lo stato di un lavoro è `SUCCEEDED` o `FAILED` dovresti essere in grado di recuperarlo per un massimo di sette giorni. Puoi comunque visualizzare i EventBridge log di un'offerta di lavoro CloudWatch o di Amazon.

Per effettuare ricerche utilizzando più criteri contemporaneamente, utilizza la funzione di ricerca avanzata. Ad esempio, puoi includere uno o tutti i seguenti filtri: Stato, intervallo di date e criteri aggiuntivi (ad esempio nome, definizione del lavoro o ID del lavoro).

Cerca AWS Batch offerte di lavoro (AWS Console)

Utilizzare questa procedura per cercare i lavori in una coda di lavoro nella AWS Batch console.

1. Passare alla console AWS Batch.
2. Nel riquadro di navigazione, scegli Jobs.
3. Attiva la ricerca avanzata.
4. Espandi l'elenco a discesa Job queue e scegli la coda dei lavori in cui desideri cercare.

> **Note**
>
> Puoi cercare offerte di lavoro all'interno di una sola coda di lavoro alla volta.

5. Per le opzioni di ricerca:

a. Nell'elenco a discesa Stato puoi scegliere uno o più stati in base ai quali filtrare. Per ulteriori informazioni, consultare Job stati e Stato della mansione di servizio.

b. Scegli Intervallo di date per filtrare i risultati in base a un intervallo di data e ora.

- Scegli la modalità Relativa per cercare lavori con una data di creazione all'interno di un intervallo di tempo contando a ritroso rispetto alla data e all'ora correnti.
- Scegli la modalità Assoluta per cercare lavori con una data di creazione entro un intervallo di data e ora specificati.

c. Nel campo Criteri aggiuntivi, inserisci le parole chiave da includere nei risultati della ricerca. Ad esempio, è possibile utilizzare questo campo per effettuare una ricerca in base al nome del lavoro, alla definizione del lavoro o all'ID del lavoro. A seconda della proprietà, è possibile che sia necessario definire operatori aggiuntivi, ad esempio equals (=) o contains (:).

> **Note**
>
> SageMaker Le code dei lavori di formazione supportano solo il filtraggio in base al nome del lavoro e al Job ID

6. Selezionare Search (Cerca).

Cerca e filtra i AWS Batch lavori ()AWS CLI

Utilizzare questa procedura per elencare tutti i lavori in una coda di lavoro con. AWS CLI Facoltativamente, utilizzare il parametro -filters per restringere i risultati in base ai criteri specificati.

Search job queue (AWS CLI)

È possibile utilizzare il comando list-jobs per cercare e filtrare una coda di lavori.

Ad esempio, puoi cercare una coda di lavoro in base al nome del lavoro:

```
aws batch list-jobs \
    --job-queue my-job-queue \
    --filters name=JOB_NAME,values="my-job"
```

Nel comando precedente, apporta le modifiche seguenti:

- *my-job-queue*Sostituiscila con il nome della tua coda di lavoro.
- Sostituiscilo *my-job* con il nome del tuo lavoro.

Search service job queue (AWS CLI)

È possibile utilizzare il list-service-jobscomando per cercare e filtrare una coda di lavori di assistenza.

Ad esempio, è possibile cercare una coda di lavori di assistenza in base al nome del lavoro:

```
aws batch list-service-jobs \
    --job-queue my-sm-queue \
    --filters name=JOB_NAME,values="my-sm-job"
```

Nel comando precedente, apporta le modifiche seguenti:

- *my-sm-queue*Sostituiscilo con il nome della tua coda di lavoro di assistenza.
- *my-sm-job*Sostituiscilo con il nome della tua mansione di assistenza.

Modalità di rete per i AWS Batch lavori

La tabella seguente descrive le modalità di rete e l'utilizzo tipico dei tipi di AWS Batch job. Consulta i link nella colonna «Job type» per maggiori dettagli su considerazioni e comportamenti.

Tipo di processo	Modalità di rete supportate	Utilizzo tipico
ECS: lavoro EC2 semplice	`host`	Utilizzato per i carichi di lavoro batch imbarazzantemente paralleli più scalabili che richiedono solo l'uscita verso vpc definito in Compute Environment.
ECS- job EC2 parallelo multinodo	`awsvpc`	Utilizzato per carichi di lavoro distribuiti su più host (nodi) strettamente accoppiati, modellati come un singolo

Tipo di processo	Modalità di rete supportate	Utilizzo tipico
		lavoro con comunicazioni coordinate tra i nodi di attività.
Un lavoro semplice con ECS-Fargate	`awsvpc`	True serverless per carichi di lavoro in batch incredibilmente paralleli. In genere il TCO più basso e il modello di lavoro con il massimo isolamento dei container.
EKS: lavoro EC2 semplice	host e pod	Utilizzato per carichi di lavoro batch altamente scalabili e incredibilmente paralleli che richiedono solo l'uscita verso vpc definito in Compute Environment. L'impostazione predefinita è la rete host.
EKS- job parallelo EC2 multinodo	host e pod	Utilizzato per carichi di lavoro distribuiti su più host (nodi) strettamente accoppiati, modellati come un unico lavoro con comunicazioni coordinate tra i nodi del pod. L'impostazione predefinita è la rete host.

Visualizza i registri dei AWS Batch lavori in CloudWatch Logs

Puoi configurare i tuoi AWS Batch lavori per inviare informazioni di registro ad Amazon CloudWatch Logs. In questo modo, puoi visualizzare diversi log dei tuoi lavori in un'unica comoda posizione. Per ulteriori informazioni, consulta Utilizzo dei CloudWatch registri con AWS Batch.

Puoi anche utilizzare i Job logs nella AWS Batch console per monitorare o risolvere un processo. AWS Batch

1. Apri la AWS Batch console.
2. Scegliere Jobs (Processi). Per informazioni più dettagliate sull'ordinamento e il filtraggio dei lavori nella coda dei lavori, consulta e Visualizza i AWS Batch lavori in una coda di lavoro Cerca offerte di lavoro in una coda di lavoro
3. Per Job queue, scegli la coda lavori che desideri.

   > **Tip**
   >
   > Se ci sono diversi lavori nella coda dei lavori, puoi attivare Ricerca e filtro per trovarli più velocemente. Per ulteriori informazioni, consulta AWS Batch Cerca offerte di lavoro in una coda di lavoro.

4. In Status, scegli lo stato del lavoro che desideri.
5. Scegli il lavoro che desideri e si aprirà la pagina Dettagli.
6. Nella pagina Dettagli, scorri verso il basso fino a Log stream name e scegli il link. Il link apre la pagina Amazon CloudWatch Logs relativa al lavoro.
7. (Facoltativo) Se è la prima volta che visualizzi i log, ti potrebbe essere richiesta l'autorizzazione.

   Per Autorizzazione obbligatoria, inserisci**OK**, quindi scegli Autorizza per accettare le CloudWatch spese di Amazon.

   > **Note**
   >
   > Per revocare l'autorizzazione agli addebiti: CloudWatch
   >
   > 1. Nel riquadro di navigazione a sinistra, scegli Autorizzazioni.
   > 2. Per Job logs, scegliete Modifica.
   > 3. Deseleziona la CloudWatch casella di controllo Autorizza Batch da usare.
   > 4. Scegli Save changes (Salva modifiche).

Rivedi le informazioni sul AWS Batch lavoro

È possibile esaminare le informazioni sul AWS Batch lavoro come lo stato, la definizione del lavoro e le informazioni sul contenitore.

1. Apri la AWS Batch console.

2. Scegliere Jobs (Processi).
3. Per Job queue, scegli la coda lavori che desideri.

> **Tip**
>
> Se ci sono diversi lavori nella coda dei lavori, puoi attivare la ricerca e il filtro per trovarli più velocemente. Per ulteriori informazioni, consulta AWS Batch Cerca offerte di lavoro in una coda di lavoro.

4. Scegli il lavoro che desideri.

> **Note**
>
> Puoi anche usare AWS Command Line Interface (AWS CLI) per visualizzare i dettagli di un AWS Batch lavoro. Per ulteriori informazioni, vedere describe-jobs nel Command Reference.AWS CLI

Sicurezza in AWS Batch

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di data center e architetture di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra te AWS e te. Il modello di responsabilità condivisa descrive questo aspetto come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi in Cloud AWS. AWS fornisce inoltre servizi che è possibile utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei AWS Programmi di AWS conformità dei Programmi di conformità dei di . Per ulteriori informazioni sui programmi di conformità applicabili AWS Batch, consulta AWS Servizi nell'ambito del programma di conformitàAWS .
- Sicurezza nel cloud: la tua responsabilità è determinata dal servizio AWS che utilizzi. Inoltre, sei responsabile anche di altri fattori, tra cui la riservatezza dei dati, i requisiti dell'azienda e le leggi e le normative applicabili.

Questa documentazione aiuta a capire come applicare il modello di responsabilità condivisa durante l'utilizzo AWS Batch. Negli argomenti seguenti viene illustrato come eseguire la configurazione AWS Batch per soddisfare gli obiettivi di sicurezza e conformità. Imparerai anche a utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere AWS Batch le tue risorse.

Argomenti

- Risolvi i problemi AWS Batch di IAM

Identity and Access Management per AWS Batch

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 può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare le risorse. AWS Batch IAM è uno Servizio AWS strumento che puoi utilizzare senza costi aggiuntivi.

Argomenti

- Destinatari
- Autenticazione con identità
- Gestione dell'accesso tramite policy
- Come AWS Batch funziona con IAM
- Esempi di policy basate sull'identità per AWS Batch
- AWS politiche gestite per AWS Batch

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia in base al tuo ruolo:

- Utente del servizio: richiedi le autorizzazioni all'amministratore se non riesci ad accedere alle funzionalità (consulta Risolvi i problemi AWS Batch di IAM)
- Amministratore del servizio: determina l'accesso degli utenti e invia le richieste di autorizzazione (consulta Come AWS Batch funziona con IAM)
- Amministratore IAM: scrivi policy per gestire l'accesso (consulta Esempi di policy basate sull'identità per AWS Batch)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi autenticarti come utente IAM o assumendo un ruolo IAM. Utente root dell'account AWS

Puoi accedere come identità federata utilizzando credenziali provenienti da una fonte di identità come AWS IAM Identity Center (IAM Identity Center), autenticazione Single Sign-On o credenziali. Google/ Facebook Per ulteriori informazioni sull'accesso, consulta Come accedere all' Account AWS nella Guida per l'utente di Accedi ad AWS .

Per l'accesso programmatico, AWS fornisce un SDK e una CLI per firmare crittograficamente le richieste. Per ulteriori informazioni, consulta AWS Signature Version 4 per le richieste API nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un Account AWS, si inizia con un'identità di accesso denominata utente Account AWS root che ha accesso completo a tutte Servizi AWS le risorse. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Per le attività che richiedono le credenziali dell'utente root, consulta Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura ottimale, richiedi agli utenti umani di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente della directory aziendale, del provider di identità Web o Directory Service che accede Servizi AWS utilizzando le credenziali di una fonte di identità. Le identità federate assumono ruoli che forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, si consiglia di utilizzare AWS IAM Identity Center. Per ulteriori informazioni, consulta Che cos'è il Centro identità IAM? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è una identità che dispone di autorizzazioni specifiche per una singola persona o applicazione. Consigliamo di utilizzare credenziali temporanee invece di utenti IAM con credenziali a lungo termine. Per ulteriori informazioni, consulta Richiedere agli utenti umani di utilizzare la federazione con un provider di identità per accedere AWS utilizzando credenziali temporanee nella Guida per l'utente IAM.

Un gruppo IAM specifica una raccolta di utenti IAM e semplifica la gestione delle autorizzazioni per gestire gruppi di utenti di grandi dimensioni. Per ulteriori informazioni, consulta Casi d'uso per utenti IAM nella Guida per l'utente di IAM.

Ruoli IAM

Un ruolo IAM è un'identità con autorizzazioni specifiche che fornisce credenziali temporanee. Puoi assumere un ruolo passando da un ruolo utente a un ruolo IAM (console) o chiamando un'operazione AWS CLI o AWS API. Per ulteriori informazioni, consulta Metodi per assumere un ruolo nella Guida per l'utente IAM.

I ruoli IAM sono utili per l'accesso federato degli utenti, le autorizzazioni utente IAM temporanee, l'accesso tra account, l'accesso tra servizi e le applicazioni in esecuzione su Amazon. EC2 Per maggiori informazioni, consultare Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

Gestione dell'accesso tramite policy

Puoi controllare l'accesso AWS creando policy e collegandole a identità o risorse. AWS Una policy definisce le autorizzazioni quando è associata a un'identità o a una risorsa. AWS valuta queste politiche quando un preside effettua una richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per maggiori informazioni sui documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Utilizzando le policy, gli amministratori specificano chi ha accesso a cosa definendo quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Un amministratore IAM crea le policy IAM e le aggiunge ai ruoli, che gli utenti possono quindi assumere. Le policy IAM definiscono le autorizzazioni indipendentemente dal metodo utilizzato per eseguirle.

Policy basate sull'identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile collegare a un'identità (utente, gruppo o ruolo). Tali policy controllano le operazioni autorizzate per l'identità, nonché le risorse e le condizioni in cui possono essere eseguite. Per informazioni su come creare una policy basata su identità, consultare Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente IAM.

Le policy basate su identità possono essere policy in linea (con embedding direttamente in una singola identità) o policy gestite (policy autonome collegate a più identità). Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scegliere tra policy gestite e policy in linea nella Guida per l'utente di IAM.

Policy basate sulle risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi includono le policy di trust dei ruoli IAM e le policy dei bucket di Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. In una policy basata sulle risorse è obbligatorio specificare un'entità principale.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non è possibile utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi che possono impostare le autorizzazioni massime concesse dai tipi di policy più comuni:

- Limiti delle autorizzazioni: imposta il numero massimo di autorizzazioni che una policy basata su identità ha la possibilità di concedere a un'entità IAM. Per ulteriori informazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente di IAM.
- Politiche di controllo del servizio (SCPs): specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa in. AWS Organizations Per ulteriori informazioni, consultare Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Politiche di controllo delle risorse (RCPs): imposta le autorizzazioni massime disponibili per le risorse nei tuoi account. Per ulteriori informazioni, consulta Politiche di controllo delle risorse (RCPs) nella Guida per l'AWS Organizations utente.
- Policy di sessione: policy avanzate passate come parametro quando si crea una sessione temporanea per un ruolo o un utente federato. Per maggiori informazioni, consultare Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando a una richiesta si applicano più tipi di policy, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire o meno una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Come AWS Batch funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS Batch, scopri con quali funzionalità IAM è possibile utilizzare AWS Batch.

Funzionalità IAM che puoi utilizzare con AWS Batch

Funzionalità IAM	AWS Batch supporto
Policy basate sull'identità	Sì
Policy basate sulle risorse	No
Operazioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
ACLs	No
ABAC (tag nelle policy)	Sì
Credenziali temporanee	Sì
Autorizzazioni del principale	Sì
Ruoli di servizio	Sì
Ruoli collegati al servizio	Sì

Per avere una panoramica di alto livello su come AWS Batch e altri AWS servizi funzionano con la maggior parte delle funzionalità IAM, consulta AWS i servizi che funzionano con IAM nella IAM User Guide.

Politiche basate sull'identità per AWS Batch

Supporta le policy basate sull'identità: sì

Le policy basate sull'identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le operazioni che

utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Definizione di autorizzazioni personalizzate IAM con policy gestite dal cliente nella Guida per l'utente di IAM.

Con le policy basate sull'identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente IAM.

Esempi di policy basate su identità per AWS Batch

Per visualizzare esempi di politiche basate sull' AWS Batch identità, vedere. Esempi di policy basate sull'identità per AWS Batch

Azioni politiche per AWS Batch

Supporta le operazioni di policy: si

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le operazioni che è possibile utilizzare per consentire o negare l'accesso in una policy. Includere le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di AWS Batch azioni, vedere Azioni definite da AWS Batch nel Service Authorization Reference.

Le azioni politiche in AWS Batch uso utilizzano il seguente prefisso prima dell'azione:

```
batch
```

Per specificare più operazioni in una sola istruzione, occorre separarle con la virgola.

```
"Action": [
      "batch:action1",
      "batch: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": "batch:Describe*"
```

Per visualizzare esempi di politiche AWS Batch basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS Batch

Risorse politiche per AWS Batch

Supporta le risorse relative alle policy: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Per le azioni che non supportano le autorizzazioni a livello di risorsa, si utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di AWS Batch risorse e relativi ARNs, vedere Resources Defined by AWS Batch nel Service Authorization Reference. Per informazioni sulle operazioni con cui è possibile specificare l'ARN di ogni risorsa, consulta Operazioni definite da AWS Batch.

Chiavi di condizione delle policy per AWS Batch

Supporta le chiavi di condizione delle policy specifiche del servizio: sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale entità principale può eseguire operazioni su quali risorse e in quali condizioni.

L'elemento `Condition` specifica quando le istruzioni vengono eseguite in base a criteri definiti. È possibile compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

Per visualizzare un elenco di chiavi di AWS Batch condizione, consulta Condition Keys for AWS Batch nel Service Authorization Reference. Per sapere con quali azioni e risorse puoi utilizzare una chiave di condizione, vedi Azioni definite da AWS Batch.

Controllo degli accessi basato sugli attributi (ABAC) con AWS Batch

Supporta ABAC (tag nelle policy): sì

Il controllo degli accessi basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base ad attributi chiamati tag. Puoi allegare tag a entità e AWS risorse IAM, quindi progettare politiche ABAC per consentire le operazioni quando il tag del principale corrisponde al tag sulla risorsa.

Per controllare l'accesso basato su tag, fornire informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name` o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Sì. Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per maggiori informazioni su ABAC, consulta Definizione delle autorizzazioni con autorizzazione ABAC nella Guida per l'utente di IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Usa credenziali temporanee con AWS Batch

Supporta le credenziali temporanee: sì

Le credenziali temporanee forniscono l'accesso a breve termine alle AWS risorse e vengono create automaticamente quando si utilizza la federazione o si cambia ruolo. AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza temporanee in IAM e Servizi AWS compatibili con IAM nelle Guida per l'utente IAM.

Autorizzazioni del principale tra servizi per AWS Batch

Supporta l'inoltro delle sessioni di accesso (FAS): sì

Le sessioni di accesso inoltrato (FAS) utilizzano le autorizzazioni del principale che chiama e, in combinazione con la richiesta Servizio AWS, Servizio AWS per effettuare richieste ai servizi downstream. Per i dettagli delle policy relative alle richieste FAS, consulta Forward access sessions.

Ruoli di servizio per AWS Batch

Supporta i ruoli di servizio: sì

Un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire operazioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta Create a role to delegate permissions to an Servizio AWS nella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere AWS Batch la funzionalità. Modifica i ruoli di servizio solo quando AWS Batch fornisce le indicazioni per farlo.

Ruoli collegati ai servizi per AWS Batch

Supporta i ruoli collegati ai servizi: sì

Un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'operazione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati al servizio, ma non modificarle.

Per ulteriori informazioni su come creare e gestire i ruoli collegati ai servizi, consulta Servizi AWS supportati da IAM. Trova un servizio nella tabella che include un `Yes` nella colonna Service-linked role (Ruolo collegato ai servizi). Scegli il collegamento Sì per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Esempi di policy basate sull'identità per AWS Batch

Per impostazione predefinita, gli utenti e i ruoli non dispongono dell'autorizzazione per creare o modificare risorse AWS Batch . Per concedere agli utenti l'autorizzazione a eseguire azioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM (console) nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS Batch, incluso il formato di ARNs per ogni tipo di risorsa, vedere Actions, Resources and Condition Keys AWS Batch nel Service Authorization Reference.

Argomenti

- Best practice per le policy
- Utilizzo della console AWS Batch
- Consentire agli utenti di visualizzare le loro autorizzazioni

Best practice per le policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare AWS Batch risorse nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando si creano o modificano policy basate sull'identità, seguire queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni.AWS Sono disponibili nel tuo. Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per maggiori informazioni, consulta Policy gestite da AWS o Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente di IAM.
- Applicazione delle autorizzazioni con privilegio minimo - Quando si impostano le autorizzazioni con le policy IAM, concedere solo le autorizzazioni richieste per eseguire un'attività. È possibile farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegio minimo. Per maggiori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente di IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso - Per limitare l'accesso ad azioni e risorse è possibile aggiungere una condizione alle policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio CloudFormation. Per maggiori informazioni, consultare la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente di IAM.

- Utilizzo dello strumento di analisi degli accessi IAM per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali - Lo strumento di analisi degli accessi IAM convalida le policy nuove ed esistenti in modo che aderiscano al linguaggio (JSON) della policy IAM e alle best practice di IAM. Lo strumento di analisi degli accessi IAM offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per maggiori informazioni, consultare Convalida delle policy per il Sistema di analisi degli accessi IAM nella Guida per l'utente di IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungere le condizioni MFA alle policy. Per maggiori informazioni, consultare Protezione dell'accesso API con MFA nella Guida per l'utente di IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

Utilizzo della console AWS Batch

Per accedere alla AWS Batch console, è necessario disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle AWS Batch risorse del tuo. Account AWS Se crei una policy basata sull'identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti o ruoli) associate a tale policy.

Non è necessario consentire autorizzazioni minime per la console agli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, è opportuno concedere l'accesso solo alle azioni che corrispondono all'operazione API che stanno cercando di eseguire.

Per garantire che utenti e ruoli possano ancora utilizzare la AWS Batch console, allega anche la policy AWS Batch `ConsoleAccess` o la policy `ReadOnly` AWS gestita alle entità. Per maggiori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente di IAM.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. 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": "*"
        }
    ]
}
```

AWS politiche gestite per AWS Batch

Puoi utilizzare politiche AWS gestite per una gestione più semplice dell'accesso all'identità per il team e le risorse a cui è stato assegnato AWS . AWS le politiche gestite coprono una serie di casi d'uso comuni, sono disponibili per impostazione predefinita nel tuo AWS account e vengono gestite e aggiornate per tuo conto. Non puoi modificare le autorizzazioni nelle politiche AWS gestite. Se hai bisogno di maggiore flessibilità, puoi in alternativa scegliere di creare policy gestite dai clienti IAM. In questo modo, puoi fornire alle risorse assegnate al tuo team solo le autorizzazioni esatte di cui hanno bisogno.

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 per tuo conto. Periodicamente, AWS i servizi aggiungono autorizzazioni aggiuntive a una policy AWS gestita. AWS le politiche gestite vengono molto probabilmente aggiornate quando diventa disponibile il lancio o l'operazione di una nuova funzionalità. Questi aggiornamenti influiscono automaticamente su tutte le identità (utenti, gruppi e ruoli) a cui è allegata la policy. Tuttavia, non rimuovono le autorizzazioni né interrompono le autorizzazioni esistenti.

Inoltre, AWS supporta politiche gestite per le funzioni lavorative che si estendono su più servizi. Ad esempio, la policy `ReadOnlyAccess` AWS gestita fornisce l'accesso in sola lettura a tutti i AWS servizi e le risorse. Quando un servizio lancia una nuova funzionalità, AWS aggiunge autorizzazioni di sola lettura per nuove operazioni e risorse. Per l'elenco e la descrizione delle policy di funzione dei processi, consultare la sezione Policy gestite da AWS per funzioni di processi nella Guida per l'utente di IAM.

AWS politica gestita: BatchServiceRolePolicy

La policy IAM BatchServiceRolePolicygestita viene utilizzata dal ruolo `AWSServiceRoleForBatch`collegato al servizio. Ciò consente di AWS Batch eseguire azioni per tuo conto. Non è possibile attribuire questa policy alle entità IAM. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Batch.

Questa politica consente AWS Batch di completare le seguenti azioni su risorse specifiche:

- `autoscaling`— Consente di AWS Batch creare e gestire risorse Amazon EC2 Auto Scaling. AWS Batch crea e gestisce gruppi Amazon EC2 Auto Scaling per la maggior parte degli ambienti di elaborazione.
- `ec2`— Consente di AWS Batch controllare il ciclo di vita delle EC2 istanze Amazon, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste EC2 Spot Fleet per alcuni ambienti di calcolo EC2 Spot.
- `ecs`- Consente di AWS Batch creare e gestire cluster Amazon ECS, definizioni di attività e attività per l'esecuzione dei lavori.

- `eks`- Consente di AWS Batch descrivere la risorsa del cluster Amazon EKS per le convalide.
- `iam`- Consente di AWS Batch convalidare e trasferire i ruoli forniti dal proprietario ad Amazon EC2, Amazon EC2 Auto Scaling e Amazon ECS.
- `logs`— Consente di AWS Batch creare e gestire gruppi di log e flussi di log per i lavori. AWS Batch

Per visualizzare il codice JSON della policy, consulta la Guida BatchServiceRolePolicydi riferimento alle politiche AWS gestite.

AWS politica gestita: AWSBatchServiceRolePolicyForSageMaker

`AWSServiceRoleForAWSBatchWithSagemaker`consente AWS Batch di eseguire azioni per tuo conto. Non è possibile attribuire questa policy alle entità IAM. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Batch.

Questa politica consente AWS Batch di completare le seguenti azioni su risorse specifiche:

- `sagemaker`— Consente di AWS Batch gestire i lavori di formazione sull' SageMaker intelligenza artificiale e altre risorse di SageMaker intelligenza artificiale.
- `iam:PassRole`— Consente di AWS Batch trasferire ruoli di esecuzione definiti dal cliente all' SageMaker IA per l'esecuzione del lavoro. Il vincolo di risorse consente di trasferire ruoli ai servizi di intelligenza artificiale. SageMaker

Per visualizzare il codice JSON della policy, consulta la Guida di riferimento AWSBatchServiceRolePolicyForSageMakeralle politiche AWS gestite.

AWS politica gestita: AWSBatchServiceRolepolitica

La politica di autorizzazione dei ruoli denominata AWSBatchServiceRoleconsente di AWS Batch completare le seguenti azioni su risorse specifiche:

La policy IAM AWSBatchServiceRolegestita viene spesso utilizzata da un ruolo denominato AWSBatchServiceRolee include le seguenti autorizzazioni. Seguendo i consigli di sicurezza standard relativi alla concessione dei privilegi minimi, la policy gestita AWSBatchServiceRolepuò 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. Questa policy e questo ruolo AWS Batch gestiti possono essere utilizzati con la maggior parte dei tipi di ambienti di calcolo, ma l'utilizzo di ruoli collegati ai servizi è preferibile per

un'esperienza gestita meno soggetta a errori, con un ambito più preciso e una migliore esperienza gestita.

- `autoscaling`— Consente di AWS Batch creare e gestire risorse Amazon EC2 Auto Scaling. AWS Batch crea e gestisce gruppi Amazon EC2 Auto Scaling per la maggior parte degli ambienti di elaborazione.
- `ec2`— Consente di AWS Batch gestire il ciclo di vita delle EC2 istanze Amazon, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste EC2 Spot Fleet per alcuni ambienti di calcolo EC2 Spot.
- `ecs`- Consente di AWS Batch creare e gestire cluster Amazon ECS, definizioni di attività e attività per l'esecuzione dei lavori.
- `iam`- Consente di AWS Batch convalidare e trasferire i ruoli forniti dal proprietario ad Amazon EC2, Amazon EC2 Auto Scaling e Amazon ECS.
- `logs`— Consente di AWS Batch creare e gestire gruppi di log e flussi di log per i lavori. AWS Batch

Per visualizzare il codice JSON della policy, consulta la Guida AWSBatchServiceRoledi riferimento alle politiche AWS gestite.

AWS politica gestita: AWSBatchFullAccess

La AWSBatchFullAccesspolitica garantisce alle AWS Batch azioni il pieno accesso alle AWS Batch risorse. Garantisce inoltre l'accesso alla descrizione e all'elenco delle azioni per i servizi Amazon EC2, Amazon ECS CloudWatch, Amazon EKS e IAM. In questo modo le identità IAM, utenti o ruoli, possono visualizzare le risorse AWS Batch gestite create per loro conto. Infine, questa policy consente anche di trasferire ruoli IAM selezionati a tali servizi.

Puoi collegarti AWSBatchFullAccessalle tue entità IAM. AWS Batch associa inoltre questa policy a un ruolo di servizio che consente di AWS Batch eseguire azioni per tuo conto.

Per visualizzare il codice JSON della policy, consulta la Guida AWSBatchFullAccessdi riferimento alle politiche AWS gestite.

AWS Batch aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti alle politiche AWS gestite AWS Batch da quando questo servizio ha iniziato a tenere traccia di queste modifiche. Per ricevere avvisi automatici sulle modifiche a questa pagina, iscriviti al feed RSS nella pagina della cronologia dei AWS Batch documenti.

Modifica	Descrizione	Data
AWSBatchServiceRolePolicyForSageMakerpolitica aggiunta	È stata aggiunta una nuova politica AWS gestita per il ruolo AWSBatchServiceRolePolicyForSageMakercollegato ai servizi che consente di AWS Batch gestire l' SageMaker IA per tuo conto.	31 luglio 2025
BatchServiceRolePolicypolitica aggiornata	Aggiornato per aggiungere il supporto per la descrizione della cronologia delle richieste e delle Amazon EC2 Auto Scaling attività di Spot Fleet.	5 dicembre 2023
AWSBatchServiceRolepolitica aggiunta	Aggiornato per aggiungere dichiarazioni IDs, concedere AWS Batch autorizzazioni a `ec2:DescribeSpotFleetRequestHistory eautoscaling:DescribeScalingActivities` .	5 dicembre 2023
BatchServiceRolePolicypolitica aggiornata	Aggiornato per aggiungere il supporto per la descrizione dei cluster Amazon EKS.	20 ottobre 2022
AWSBatchFullAccesspolitica aggiornata	Aggiornato per aggiungere il supporto per elencare e descrivere i cluster Amazon EKS.	20 ottobre 2022
BatchServiceRolePolicypolitica aggiornata	Aggiornato per aggiungere il supporto per i gruppi Amazon EC2 Capacity Reservation gestiti da AWS Resource Groups. Per ulteriori informazioni, consulta Work with Capacity Reservation groups nella Amazon EC2 User Guide.	18 maggio 2022
BatchServiceRolePolicye AWSBatchServiceRolepolitiche aggiornate	Aggiornato per aggiungere il supporto per la descrizione dello stato delle istanze AWS Batch gestite in Amazon in EC2 modo da sostituire le istanze non integre.	06 dicembre 2021

Modifica	Descrizione	Data
BatchServiceRolePolicypolitica aggiornata	Aggiornato per aggiungere il supporto per il gruppo di collocamento, la prenotazione della capacità, la GPU elastica e le risorse Elastic Inference in Amazon. EC2	26 marzo 2021
BatchServiceRolePolicypolitica aggiunta	Con la politica BatchServiceRolePolicygestita per il ruolo AWSServiceRoleForBatchcollegato al servizio, è possibile utilizzare un ruolo collegato al servizio gestito da. AWS Batch Con questa policy, non è necessario mantenere il proprio ruolo da utilizzare negli ambienti di elaborazione.	10 marzo 2021
AWSBatchFullAccess- aggiungi l'autorizzazione per aggiungere un ruolo collegato al servizio	Aggiungi le autorizzazioni IAM per consentire l'aggiunta del ruolo AWSServiceRoleForBatchcollegato al servizio all'account.	10 marzo 2021
AWS Batch ha iniziato a tenere traccia delle modifiche	AWS Batch ha iniziato a tenere traccia delle modifiche per le sue politiche AWS gestite.	10 marzo 2021

AWS Batch Politiche, ruoli e autorizzazioni IAM

Per impostazione predefinita, gli utenti non dispongono dell'autorizzazione per creare o modificare AWS Batch risorse o per eseguire attività utilizzando l' AWS Batch API, la AWS Batch console o il AWS CLI. Per consentire agli utenti di eseguire queste azioni, crea policy IAM che concedano agli utenti l'autorizzazione per le risorse e le operazioni API specifiche. Quindi, collega le policy agli utenti o ai gruppi che richiedono tali autorizzazioni.

Quando si associa una politica a un utente o a un gruppo di utenti, la politica consente o nega le autorizzazioni per eseguire attività specifiche su risorse specifiche. Per ulteriori informazioni, consulta Autorizzazioni e politiche nella Guida per l'utente IAM. Per ulteriori informazioni sulla gestione e la creazione di policy IAM personalizzate, consulta la sezione relativa alla gestione delle policy IAM.

AWS Batch effettua chiamate verso altre persone per tuo Servizi AWS conto. Di conseguenza, AWS Batch deve autenticarsi utilizzando le proprie credenziali. Più specificamente, si AWS Batch autentica creando un ruolo e una policy IAM che forniscono queste autorizzazioni. Quindi, associa il ruolo agli ambienti di calcolo al momento della creazione. Per ulteriori informazioni, consulta IAM Roles Ruolo dell'istanza Amazon ECS, Using Service-Linked Roles e Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM.

Argomenti

- Struttura delle politiche IAM
- Risorsa: Politiche di esempio per AWS Batch
- Risorsa: politica AWS Batch gestita

Struttura delle politiche IAM

Nei seguenti argomenti viene illustrata la struttura di una policy IAM.

Argomenti

- Sintassi delle policy
- Azioni API per AWS Batch
- Nomi di risorse Amazon per AWS Batch
- Verifica che gli utenti dispongano delle autorizzazioni richieste

Sintassi delle policy

Una policy IAM è un documento JSON costituito da una o più dichiarazioni. Ogni dichiarazione è strutturata come segue.

```
{
  "Statement":[{
    "Effect":"effect",
    "Action":"action",
    "Resource":"arn",
    "Condition":{
      "condition":{
    "key":"value"
    }
      }
```

```
    }
  ]
}
```

Ci sono quattro elementi principali che costituiscono una dichiarazione:

- Effetto: l'elemento effect può essere `Allow` o `Deny`. Per impostazione predefinita, gli utenti non sono autorizzati a utilizzare risorse e azioni API. Pertanto, tutte le richieste vengono rifiutate. Un permesso esplicito sostituisce l'impostazione predefinita. Un rifiuto esplicito sovrascrive tutti i consensi.
- Azione: l'azione è l'azione API specifica per la quale concedi o neghi l'autorizzazione. Per istruzioni su come specificare l'azione, consulta. Azioni API per AWS Batch
- Resource (Risorsa): la risorsa che viene modificata dall'operazione. Con alcune azioni AWS Batch API, puoi includere nella tua policy risorse specifiche che possono essere create o modificate dall'azione. Per specificare una risorsa nella dichiarazione, si utilizza il suo Amazon Resource Name (ARN). Per ulteriori informazioni, consultare Autorizzazioni supportate a livello di risorsa per le azioni API AWS Batch e Nomi di risorse Amazon per AWS Batch. Se l'operazione AWS Batch API attualmente non supporta le autorizzazioni a livello di risorsa, includi un carattere jolly (*) per specificare che tutte le risorse possono essere influenzate dall'azione.
- Condition: le condizioni sono facoltative. Possono essere utilizzate per controllare quando è in vigore una policy.

Per ulteriori informazioni su esempi di dichiarazioni politiche IAM per, consulta. AWS BatchRisorsa: Politiche di esempio per AWS Batch

Azioni API per AWS Batch

In una dichiarazione di policy IAM, è possibile specificare qualsiasi operazione API per qualsiasi servizio che supporta IAM. Per AWS Batch, utilizza il seguente prefisso con il nome dell'azione API: `batch:` (ad esempio, `batch:SubmitJob` e`batch:CreateComputeEnvironment`).

Per specificare più azioni in una singola istruzione, separa ogni azione con una virgola.

```
"Action": ["batch:action1", "batch:action2"]
```

È inoltre possibile specificare più azioni includendo un carattere jolly (*). Ad esempio, puoi specificare tutte le azioni con un nome che inizia con la parola «Descrivi».

```
"Action": "batch:Describe*"
```

Per specificare tutte le azioni AWS Batch API, includi un carattere jolly (*).

```
"Action": "batch:*"
```

Per un elenco di AWS Batch azioni, consulta Azioni nel riferimento AWS Batch API.

Nomi di risorse Amazon per AWS Batch

Ogni dichiarazione di policy IAM si applica alle risorse specificate utilizzando i rispettivi Amazon Resource Names (ARNs).

Un Amazon Resource Name (ARN) ha la seguente sintassi generale:

```
arn:aws:[service]:[region]:[account]:resourceType/resourcePath
```

service

Il servizio (ad esempio `batch`).

Regione

Il Regione AWS per la risorsa (ad esempio,`us-east-2`).

account

L' Account AWS ID, senza trattini (ad esempio,`123456789012`).

resourceType

Il tipo di risorsa (ad esempio `compute-environment`).

resourcePath

Un percorso che identifica la risorsa. Puoi usare un carattere jolly (*) nei tuoi percorsi.

AWS Batch Le operazioni API attualmente supportano le autorizzazioni a livello di risorsa su diverse operazioni API. Per ulteriori informazioni, consulta Autorizzazioni supportate a livello di risorsa per le azioni API AWS Batch. Per specificare tutte le risorse, o se un'azione API specifica non supporta ARNs, includi un carattere jolly (*) nell'elemento. `Resource`

```
"Resource": "*"
```

Verifica che gli utenti dispongano delle autorizzazioni richieste

Prima di mettere in produzione una policy IAM, assicurati che conceda agli utenti le autorizzazioni per utilizzare le azioni e le risorse API specifiche di cui hanno bisogno.

Per fare ciò, crea innanzitutto un utente a scopo di test e allega la policy IAM all'utente di test. In seguito, effettua una richiesta come utente di test. nella console o con la AWS CLI.

> **Note**
>
> Puoi anche testare le tue policy utilizzando IAM Policy Simulator. Per ulteriori informazioni sul simulatore di policy, consulta Working with the IAM Policy Simulator nella IAM User Guide.

Se la policy non concede all'utente le autorizzazioni previste oppure è eccessivamente permissiva, puoi modificarla in base alle esigenze. Ripeti il test fino a ottenere i risultati desiderati.

> **Important**
>
> La propagazione delle modifiche alla policy e la loro validità potrebbe richiedere alcuni minuti. Pertanto, ti consigliamo di attendere almeno cinque minuti prima di testare gli aggiornamenti delle policy.

Se una verifica dell'autorizzazione ha esito negativo, la richiesta restituisce un messaggio codificato con informazioni di diagnostica. Il messaggio può essere decodificato tramite l'operazione `DecodeAuthorizationMessage`. Per ulteriori informazioni, consulta DecodeAuthorizationMessagel'AWS Security Token Service API Reference e decode-authorization-messageil AWS CLI Command Reference.

Risorsa: Politiche di esempio per AWS Batch

Puoi creare policy IAM specifiche per limitare le chiamate e le risorse a cui hanno accesso gli utenti del tuo account. Quindi, puoi allegare tali politiche agli utenti.

Quando alleghi una politica a un utente o a un gruppo di utenti, la politica consente o nega l'autorizzazione degli utenti per attività specifiche su risorse specifiche. Per ulteriori informazioni, consulta Autorizzazioni e politiche nella Guida per l'utente IAM. Per istruzioni su come gestire e creare policy IAM personalizzate, consulta Managing IAM Policies.

Gli esempi seguenti mostrano le dichiarazioni politiche che puoi utilizzare per controllare le autorizzazioni di cui dispongono gli AWS Batch utenti.

Esempi

- Risorsa: accesso in sola lettura per AWS Batch
- Risorsa: limita all'utente POSIX, all'immagine Docker, al livello di privilegio e al ruolo nell'invio del lavoro
- Risorsa: Limita al prefisso di definizione del lavoro all'invio del lavoro
- Risorsa: Limita a una coda di lavoro
- Nega l'azione quando tutte le condizioni corrispondono alle stringhe
- Risorsa: nega l'azione quando i tasti condizionali corrispondono alle stringhe
- Risorsa: utilizza la chiave di batch:ShareIdentifier condizione
- Gestisci le risorse di SageMaker intelligenza artificiale con AWS Batch

Risorsa: accesso in sola lettura per AWS Batch

La seguente politica concede agli utenti le autorizzazioni per utilizzare tutte le azioni AWS Batch API con un nome che inizia con e. `Describe` `List`

A meno che un'altra dichiarazione non conceda loro l'autorizzazione a farlo, gli utenti non sono autorizzati a eseguire alcuna azione sulle risorse. Per impostazione predefinita, viene loro negata l'autorizzazione a utilizzare le azioni API.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:Describe*",
                "batch:List*",
                "batch:Get*"
            ],
            "Resource": "*"
        }
    ]
```

```
}
```

Risorsa: limita all'utente POSIX, all'immagine Docker, al livello di privilegio e al ruolo nell'invio del lavoro

La seguente politica consente a un utente POSIX di gestire il proprio set di definizioni di lavoro limitate.

Utilizzate la prima e la seconda istruzione per registrare e annullare la registrazione di qualsiasi nome di definizione di lavoro il cui nome è preceduto da. *JobDefA_*

La prima istruzione utilizza inoltre le chiavi di contesto condizionali per impostare restrizioni per utente POSIX, stato con privilegi, valori immagine container nelle `containerProperties` di definizione del processo. Per ulteriori informazioni, consulta RegisterJobDefinition nella documentazione di riferimento dell'API AWS Batch . In questo esempio, le definizioni dei job possono essere registrate solo quando l'utente POSIX è impostato su. `nobody` Il flag privilegiato è impostato su. `false` Infine, l'immagine viene impostata su `myImage` un repository Amazon ECR.

> **Important**
>
> Docker risolve il `user` parametro per quell'utente `uid` dall'interno dell'immagine del contenitore. Nella maggior parte dei casi, questo si trova nel `/etc/passwd` file all'interno dell'immagine del contenitore. Questa risoluzione dei nomi può essere evitata utilizzando `uid` valori diretti sia nella definizione del lavoro che in qualsiasi policy IAM associata. Sia le operazioni AWS Batch API che le chiavi condizionali `batch:User` IAM supportano valori numerici.

Utilizza la terza istruzione per limitare solo un ruolo specifico alla definizione di un lavoro.

JSON

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

```
                "batch:RegisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
            ],
            "Condition": {
                "StringEquals": {
                    "batch:User": [
                        "nobody"
                    ],
                    "batch:Image": [
                        "999999999999.dkr.ecr.us-east-2.amazonaws.com/myImage"
                    ]
                },
                "Bool": {
                    "batch:Privileged": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "batch:DeregisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:us-east-2:999999999999:job-definition/JobDefA_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::999999999999:role/MyBatchJobRole"
            ]
        }
    ]
}
```

Risorsa: Limita al prefisso di definizione del lavoro all'invio del lavoro

Utilizza la seguente politica per inviare lavori a qualsiasi coda di lavoro con qualsiasi nome di definizione di processo che inizi con. *JobDefA*

> **Important**
>
> Quando si definisca l'accesso a livello di risorsa per l'invio del processo, è necessario fornire sia la coda processo sia i tipi di risorse di definizione processo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:SubmitJob"
            ],
            "Resource": [
                "arn:aws:batch:us-east-2:111122223333:job-definition/JobDefA_*",
                "arn:aws:batch:us-east-2:111122223333:job-queue/*"
            ]
        }
    ]
}
```

Risorsa: Limita a una coda di lavoro

Utilizza la seguente politica per inviare i lavori a una coda di lavori specifica denominata queue1 con qualsiasi nome di definizione del processo.

> **Important**
>
> Quando si definisca l'accesso a livello di risorsa per l'invio del processo, è necessario fornire sia la coda processo sia i tipi di risorse di definizione processo.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:SubmitJob"
            ],
            "Resource": [
                "arn:aws:batch:us-east-2:888888888888:job-definition/*",
                "arn:aws:batch:us-east-2:888888888888:job-queue/queue1"
            ]
        }
    ]
}
```

Nega l'azione quando tutte le condizioni corrispondono alle stringhe

La seguente politica nega l'accesso al funzionamento dell'RegisterJobDefinitionAPI quando sia la chiave di condizione `batch:Image` (ID dell'immagine del contenitore) è "*string1*" sia la chiave di condizione `batch:LogDriver` (container log driver) è "». *string2* AWS Batch valuta le chiavi di condizione su ogni contenitore. Quando un lavoro si estende su più contenitori, ad esempio un processo parallelo a più nodi, è possibile che i contenitori abbiano configurazioni diverse. Se più chiavi di condizione vengono valutate in un'unica istruzione, vengono combinate utilizzando la logica. AND Pertanto, se una delle più chiavi di condizione non corrisponde a un contenitore, l'Denyeffetto non viene applicato a quel contenitore. Piuttosto, un contenitore diverso nello stesso lavoro potrebbe essere negato.

Per l'elenco delle chiavi di condizione per AWS Batch, vedi Condition keys for AWS Batch nel Service Authorization Reference. Ad eccezione di`batch:ShareIdentifier`, tutte le chiavi di `batch` condizione possono essere utilizzate in questo modo. La chiave di `batch:ShareIdentifier` condizione è definita per un lavoro, non per una definizione di processo.

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": "batch:RegisterJobDefinition",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "batch:Image": "string1",
          "batch:LogDriver": "string2"
        }
      }
    }
  ]
}
```

Risorsa: nega l'azione quando i tasti condizionali corrispondono alle stringhe

La seguente politica nega l'accesso all'operazione RegisterJobDefinitionAPI quando la chiave di condizione `batch:Image` (ID dell'immagine del contenitore) è "*`string1`*" o la chiave di condizione `batch:LogDriver` (container log driver) è "». *`string2`* Quando un lavoro si estende su più contenitori, ad esempio un processo parallelo a più nodi, è possibile che i contenitori abbiano configurazioni diverse. Se più chiavi di condizione vengono valutate in un'unica istruzione, vengono combinate utilizzando la logica. `AND` Pertanto, se una delle più chiavi di condizione non corrisponde a un contenitore, l'`Deny`effetto non viene applicato a quel contenitore. Piuttosto, un contenitore diverso nello stesso lavoro potrebbe essere negato.

Per l'elenco delle chiavi di condizione per AWS Batch, vedi Condition keys for AWS Batch nel Service Authorization Reference. Ad eccezione di`batch:ShareIdentifier`, tutte le chiavi di `batch` condizione possono essere utilizzate in questo modo. (La chiave di `batch:ShareIdentifier` condizione è definita per un lavoro, non per una definizione di processo.)

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:Image": [
            "string1"
          ]
        }
      }
    },
    {
      "Effect": "Deny",
      "Action": [
        "batch:RegisterJobDefinition"
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
        "StringEquals": {
          "batch:LogDriver": [
            "string2"
          ]
```

```
        }
      }
    }
  ]
}
```

Risorsa: utilizza la chiave di **`batch:ShareIdentifier`** condizione

Utilizza la seguente politica per inviare i lavori che utilizzano la definizione del `jobDefA` processo alla coda dei `jobqueue1` lavori con l'identificatore di `lowCpu` condivisione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:us-east-2:555555555555:job-definition/JobDefA",
        "arn:aws:batch:us-east-2:555555555555:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
      }
    }
  ]
}
```

Gestisci le risorse di SageMaker intelligenza artificiale con AWS Batch

Questa politica consente di AWS Batch gestire le risorse SageMaker AI.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:CreateServiceLinkedRole"
            ],
            "Resource": "arn:aws:iam::*:role/
*AWSServiceRoleForAWSBatchWithSagemaker",
            "Condition": {
                 "StringEquals": {
                     "iam:AWSServiceName": "sagemaker-
queuing.batch.amazonaws.com"
                 }
             }
         },
         {
             "Effect": "Allow",
             "Action": "iam:PassRole",
             "Resource": "*",
             "Condition": {
                 "StringEquals": {
                     "iam:PassedToService": [
                         "sagemaker.amazonaws.com"
                      ]
                  }
              }
          }
    ]
}
```

Risorsa: politica AWS Batch gestita

AWS Batch fornisce una politica gestita che è possibile allegare agli utenti. Questa policy fornisce il permesso di utilizzare AWS Batch risorse e operazioni API. Puoi applicare questa policy direttamente oppure utilizzarla come punto di partenza per la creazione delle tue policy. Per ulteriori informazioni su ciascuna operazione API menzionata in queste politiche, consulta Azioni nell'AWS Batch API Reference.

AWSBatchFullAccess

Questa politica consente l'accesso completo dell'amministratore a AWS Batch.

Per visualizzare il codice JSON della policy, consulta AWSBatchFullAccessla Guida di riferimento alle politiche AWS gestite.

AWS Batch Ruolo di esecuzione IAM

Il ruolo di esecuzione concede al contenitore e AWS Fargate agli agenti Amazon ECS l'autorizzazione a effettuare chiamate AWS API per tuo conto.

> **Note**
>
> Il ruolo di esecuzione è supportato dall'agente container Amazon ECS versione 1.16.0 e successive.

Il ruolo di esecuzione IAM è richiesto in base ai requisiti dell'attività. Puoi avere più ruoli di esecuzione per scopi e servizi diversi associati al tuo account.

> **Note**
>
> Per informazioni sul ruolo dell'istanza Amazon ECS, consultaRuolo dell'istanza Amazon ECS. Per informazioni sui ruoli di servizio, consultaCome AWS Batch funziona con IAM.

Amazon ECS fornisce la policy `AmazonECSTaskExecutionRolePolicy` gestita. Questa policy contiene le autorizzazioni necessarie per i casi d'uso comuni sopra descritti. Potrebbe essere necessario aggiungere politiche in linea al ruolo di esecuzione per i casi d'uso speciali descritti di seguito.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Autorizzazioni supportate a livello di risorsa per le azioni API AWS Batch

Il termine autorizzazioni a livello di risorsa si riferisce alla capacità di specificare le risorse su cui gli utenti possono eseguire azioni. AWS Batch supporta parzialmente le autorizzazioni a livello di risorsa. Per alcune AWS Batch azioni, è possibile controllare quando gli utenti sono autorizzati a utilizzare tali azioni in base alle condizioni che devono essere soddisfatte. Puoi anche controllare in base alle risorse specifiche che gli utenti sono autorizzati a utilizzare. Ad esempio, è possibile concedere agli utenti le autorizzazioni per inviare processi, ma solo per una determinata coda di processo e solo con una definizione di processo specifica.

Per i dettagli sulle azioni e sui tipi di risorse definiti da AWS Batch, incluso il formato di ARNs per ogni tipo di risorsa, vedere Azioni, risorse e chiavi di condizione AWS Batchnel riferimento di autorizzazione del servizio.

Tutorial: Creare il ruolo di esecuzione IAM

Se il tuo account non dispone già di un ruolo di esecuzione IAM, utilizza i seguenti passaggi per creare il ruolo.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Roles (Ruoli).
3. Selezionare Create role (Crea ruolo).
4. Per il tipo di entità affidabile, scegli Servizio AWS.
5. Per Service o use case, scegli Elastic Container Service. Quindi scegli nuovamente Elastic Container Service Task.
6. Scegli Next (Successivo).
7. Per le politiche di autorizzazione, cerca Amazon ECSTask ExecutionRolePolicy.
8. Seleziona la casella di controllo a sinistra della ECSTask ExecutionRolePolicy politica di Amazon, quindi scegli Avanti.
9. Per Nome ruolo, inserisci `ecsTaskExecutionRole` e quindi scegli Crea ruolo.

Tutorial: verifica il ruolo di esecuzione IAM

Utilizza la seguente procedura per verificare che il tuo account disponga già del ruolo di esecuzione IAM e allega la policy IAM gestita, se necessario.

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Cerca l'elenco dei ruoli per `ecsTaskExecutionRole`. Se non riesci a trovare il ruolo, consultaTutorial: Creare il ruolo di esecuzione IAM. Se hai trovato il ruolo, scegli il ruolo per visualizzare le politiche allegate.
4. Nella scheda Autorizzazioni, verifica che la policy ECSTask ExecutionRolePolicy gestita da Amazon sia associata al ruolo. Se la policy è allegata, il tuo ruolo di esecuzione è configurato correttamente. In caso contrario, la procedura riportata di seguito consente di collegare la policy.
   a. Scegli Aggiungi autorizzazioni, quindi scegli Allega politiche.
   b. Cerca Amazon ECSTask ExecutionRolePolicy.
   c. Seleziona la casella a sinistra della ECSTask ExecutionRolePolicy politica di Amazon e scegli Allega politiche.
5. Scegli Trust relationships (Relazioni di trust).
6. Verifica che la relazione di trust includa la policy seguente. Se la relazione di fiducia corrisponde alla politica seguente, il ruolo è configurato correttamente. Se la relazione di fiducia non corrisponde, scegli Modifica politica di fiducia, inserisci quanto segue e scegli Aggiorna politica.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Utilizzo di ruoli collegati ai servizi per AWS Batch

AWS Batch utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS Batch I ruoli collegati ai servizi sono predefiniti AWS Batch e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

AWS Batch utilizza due diversi ruoli collegati ai servizi:

- AWSServiceRoleForBatch- Per AWS Batch operazioni che includono ambienti di elaborazione.
- AWSServiceRoleForAWSBatchWithSagemaker- Per la gestione e l'accodamento dei carichi di lavoro basati sull' SageMaker intelligenza artificiale.

Argomenti

- Utilizzo dei ruoli per AWS Batch
- Utilizzo dei ruoli per AWS Batch l'intelligenza artificiale SageMaker

Utilizzo dei ruoli per AWS Batch

AWS Batch utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS Batch I ruoli collegati ai servizi sono predefiniti AWS Batch e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione AWS Batch perché non è necessario aggiungere manualmente le autorizzazioni necessarie. AWS Batch definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS Batch Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni. Una policy delle autorizzazioni specifica non può essere collegata a un'altra entità IAM.

> **Note**
>
> Effettua una delle seguenti operazioni per specificare un ruolo di servizio per un AWS Batch ambiente di calcolo.
>
> - Usa una stringa vuota per il ruolo di servizio. Ciò consente di AWS Batch creare il ruolo di servizio.
> - Specificare il ruolo di servizio nel seguente formato:`arn:aws:iam::`*`account_number`*`:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.
>
> Per ulteriori informazioni, consulta Nome del ruolo o ARN errati la Guida AWS Batch per l'utente.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. In questo modo proteggi AWS Batch le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta AWS i servizi che funzionano con IAM e cerca i servizi con Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per AWS Batch

AWS Batch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForBatch: consente di creare e gestire AWS Batch AWS risorse per conto dell'utente.

Il ruolo AWSService RoleForBatch collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `batch.amazonaws.com`

La politica di autorizzazione dei ruoli denominata BatchServiceRolePolicyconsente di AWS Batch completare le seguenti azioni sulle risorse specificate:

- `autoscaling`— Consente di AWS Batch creare e gestire risorse Amazon EC2 Auto Scaling. AWS Batch crea e gestisce gruppi Amazon EC2 Auto Scaling per la maggior parte degli ambienti di elaborazione.
- `ec2`— Consente di AWS Batch controllare il ciclo di vita delle EC2 istanze Amazon, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste EC2 Spot Fleet per alcuni ambienti di calcolo EC2 Spot.
- `ecs`- Consente di AWS Batch creare e gestire cluster Amazon ECS, definizioni di attività e attività per l'esecuzione dei lavori.
- `eks`- Consente di AWS Batch descrivere la risorsa del cluster Amazon EKS per le convalide.
- `iam`- Consente di AWS Batch convalidare e trasferire i ruoli forniti dal proprietario ad Amazon EC2, Amazon EC2 Auto Scaling e Amazon ECS.
- `logs`— Consente di AWS Batch creare e gestire gruppi di log e flussi di log per i lavori. AWS Batch

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente IAM.

Creazione di un ruolo collegato ai servizi per AWS Batch

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente di elaborazione nella Console di gestione AWS, la o l' AWS API AWS CLI, AWS Batch crea automaticamente il ruolo collegato al servizio.

> **Important**
>
> Questo ruolo collegato al servizio può apparire nell'account, se è stata completata un'operazione in un altro servizio che utilizza le caratteristiche supportate da questo ruolo. Se utilizzavi il AWS Batch servizio prima del 10 marzo 2021, quando ha iniziato a supportare ruoli collegati al servizio, hai AWS Batch creato il AWSService RoleForBatch ruolo nel tuo account. Per ulteriori informazioni, vedi A new role appeared in my. Account AWS

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un ambiente di elaborazione, AWS Batch crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato ai servizi per AWS Batch

AWS Batch non consente di modificare il ruolo collegato al AWSService RoleForBatch servizio. Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'utente di IAM.

Per consentire a un'entità IAM di modificare la descrizione del ruolo collegato al servizio AWSService RoleForBatch

Aggiungi la seguente dichiarazione alla politica delle autorizzazioni. Ciò consente all'entità IAM di modificare la descrizione di un ruolo collegato al servizio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:UpdateRoleDescription"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Eliminazione di un ruolo collegato ai servizi per AWS Batch

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare quel ruolo. In questo modo non sarà più presente un'entità non

utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

Per consentire a un'entità IAM di eliminare il ruolo collegato al servizio AWSService RoleForBatch

Aggiungi la seguente dichiarazione alla politica delle autorizzazioni. Ciò consente all'entità IAM di eliminare un ruolo collegato al servizio.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:DeleteServiceLinkedRole",
        "iam:GetServiceLinkedRoleDeletionStatus"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/batch.amazonaws.com/
AWSServiceRoleForBatch",
    "Condition": {"StringLike": {"iam:AWSServiceName": "batch.amazonaws.com"}}
}
```

Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato al servizio, devi prima confermare che il ruolo non abbia sessioni attive ed eliminare tutti gli ambienti di AWS Batch calcolo che utilizzano il ruolo in tutte le AWS regioni in un'unica partizione.

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 quindi il AWSService RoleForBatch 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 sai se AWS Batch sta usando il AWSService RoleForBatch ruolo, puoi provare a eliminarlo. Se il servizio utilizza il ruolo, il ruolo non verrà eliminato. È possibile visualizzare le regioni in cui viene utilizzato il ruolo. 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 ai servizi.

Per rimuovere AWS Batch le risorse utilizzate dal ruolo collegato al AWSService RoleForBatch servizio

È necessario eliminare tutti gli ambienti di AWS Batch elaborazione che utilizzano il AWSService RoleForBatch ruolo in tutte le AWS regioni prima di poter eliminare il ruolo. AWSService RoleForBatch

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel riquadro di navigazione, seleziona Compute environments (Ambienti di calcolo).
4. Seleziona l'ambiente di calcolo.
5. Scegliere Disabilita. Attendi che lo Stato passi a DISABILITATO.
6. Seleziona l'ambiente di calcolo.
7. Scegli Elimina. Conferma di voler eliminare l'ambiente di calcolo scegliendo Elimina ambiente di calcolo.
8. Ripeti i passaggi da 1 a 7 per tutti gli ambienti di calcolo che utilizzano il ruolo collegato ai servizi in tutte le regioni.

Eliminazione di un ruolo collegato al servizio in IAM (Console)

Puoi utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione della console IAM seleziona Ruoli. Quindi seleziona la casella di controllo accanto a AWSServiceRoleForBatch, non il nome o la riga stessa.
3. Scegli Delete role (Elimina ruolo).
4. Nella finestra di dialogo di conferma controlla i dati relativi all'ultimo accesso ai servizi, che indicano quando ognuno dei ruoli selezionati ha effettuato l'accesso a un Servizio AWS. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, seleziona Yes, Delete (Sì, elimina) per richiedere l'eliminazione del ruolo collegato ai servizi.
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta

richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno.

- Se il task viene eseguito correttamente, il ruolo viene rimosso dall'elenco e nella parte superiore della pagina viene visualizzata una notifica di completamento.
- Se il task non viene eseguito correttamente, puoi scegliere View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

  > **Note**
  >
  > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna.

- Se il task non viene eseguito e la notifica non include un elenco di risorse, il servizio potrebbe non restituire questa informazione. Per scoprire come eliminare le risorse per quel servizio, consulta Servizi AWS che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione di un ruolo collegato al servizio in IAM ()AWS CLI

È possibile utilizzare i comandi IAM di AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (CLI)

1. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id`dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, digita il seguente comando:

```
$ aws iam delete-service-linked-role --role-name AWSServiceRoleForBatch
```

2. Digita il seguente comando per verificare lo stato del processo di eliminazione:

```
$ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
```

Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

> **Note**
>
> In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio potrebbe restituire tutte le risorse, alcune. Oppure, potrebbe non riportare alcuna risorsa. Per informazioni su come ripulire le risorse per un servizio che non riporta alcuna risorsa, consulta AWS Servizi che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione di un ruolo collegato al servizio in IAM (API)AWS

È possibile utilizzare l'API di IAM; per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (API)

1. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, chiamare DeleteServiceLinkedRole. Nella richiesta, specifica il nome del AWSService RoleForBatch ruolo.

Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId`dalla risposta per controllare lo stato del task di eliminazione.

2. Chiamare GetServiceLinkedRoleDeletionStatus per controllare lo stato dell'eliminazione. Nella richiesta, specificare il `DeletionTaskId`.

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

   > **Note**
   >
   > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna. Per scoprire come eliminare le risorse per un servizio che non restituisce nessuna risorsa, consulta Servizi AWS che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Regioni supportate per i ruoli collegati ai servizi AWS Batch

AWS Batch supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta la pagina relativa agli endpoint AWS Batch.

Utilizzo dei ruoli per AWS Batch l'intelligenza artificiale SageMaker

AWS Batch utilizza ruoli collegati ai servizi AWS Identity and Access Management (IAM). Un ruolo collegato ai servizi è un tipo unico di ruolo IAM a cui è collegato direttamente. AWS Batch I ruoli

collegati ai servizi sono predefiniti AWS Batch e includono tutte le autorizzazioni richieste dal servizio per chiamare altri servizi per tuo conto. AWS

Un ruolo collegato al servizio semplifica la configurazione AWS Batch perché non è necessario aggiungere manualmente le autorizzazioni necessarie. AWS Batch definisce le autorizzazioni dei ruoli collegati ai servizi e, se non diversamente definito, solo può assumerne i ruoli. AWS Batch Le autorizzazioni definite includono la policy di attendibilità e la policy delle autorizzazioni che non può essere allegata a nessun'altra entità IAM.

È possibile eliminare un ruolo collegato al servizio solo dopo avere eliminato le risorse correlate. In questo modo proteggi AWS Batch le tue risorse perché non puoi rimuovere inavvertitamente l'autorizzazione ad accedere alle risorse.

Per informazioni su altri servizi che supportano i ruoli collegati ai servizi, consulta AWS i servizi che funzionano con IAM e cerca i servizi con Sì nella colonna Ruoli collegati ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato al servizio per tale servizio.

Autorizzazioni di ruolo collegate ai servizi per AWS Batch

AWS Batch utilizza il ruolo collegato al servizio denominato AWSServiceRoleForAWSBatchWithSagemaker: consente di AWS Batch mettere in coda e gestire SageMaker i lavori di formazione per tuo conto.

Il ruolo AWSService RoleFor AWSBatch WithSagemaker collegato al servizio prevede che i seguenti servizi assumano il ruolo:

- `sagemaker-queuing.batch.amazonaws.com`

La politica di autorizzazione dei ruoli consente di AWS Batch completare le seguenti azioni sulle risorse specificate:

- `sagemaker`— Consente di AWS Batch gestire i lavori di SageMaker formazione, trasformare i lavori e altre risorse di SageMaker intelligenza artificiale.
- `iam:PassRole`— Consente di AWS Batch trasferire ruoli di esecuzione definiti dal cliente all' SageMaker IA per l'esecuzione del lavoro. Il vincolo di risorse consente di trasferire ruoli ai servizi di intelligenza artificiale. SageMaker

Per consentire a utenti, gruppi o ruoli di creare, modificare o eliminare un ruolo orientato ai servizi, devi configurare le autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente IAM.

Creazione di un ruolo collegato ai servizi per AWS Batch

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando crei un ambiente di servizio utilizzando `CreateServiceEnvironment` in Console di gestione AWS, the o l' AWS API AWS CLI, AWS Batch crea automaticamente il ruolo collegato al servizio.

Se elimini questo ruolo collegato al servizio, è possibile ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando crei un ambiente di servizio utilizzando`CreateServiceEnvironment`, AWS Batch crea nuovamente il ruolo collegato al servizio per te.

Per visualizzare il codice JSON della policy, consulta AWSBatchServiceRolePolicyForSageMakerla Guida di riferimento per le policy AWS gestite.

Modifica di un ruolo collegato ai servizi per AWS Batch

AWS Batch non consente di modificare il ruolo collegato al AWSService RoleFor AWSBatch WithSagemaker servizio. Dopo avere creato un ruolo collegato al servizio, non sarà possibile modificarne il nome perché varie entità potrebbero farvi riferimento. È possibile tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori informazioni, consulta Modifica di un ruolo collegato al servizio nella Guida per l'utente di IAM.

Eliminazione di un ruolo collegato ai servizi per AWS Batch

Se non è più necessario utilizzare una caratteristica o un servizio che richiede un ruolo collegato ai servizi, ti consigliamo di eliminare quel ruolo. In questo modo non sarà più presente un'entità non utilizzata che non viene monitorata e gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato ai servizi prima di poterlo eliminare manualmente.

Pulizia di un ruolo collegato ai servizi

Prima di poter utilizzare IAM per eliminare un ruolo collegato al servizio, è necessario innanzitutto confermare che il ruolo non abbia sessioni attive ed eliminare tutti gli ambienti di servizio che utilizzano il ruolo in tutte le AWS regioni in un'unica partizione.

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 quindi il AWSService RoleFor AWSBatch WithSagemaker 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 sai se AWS Batch sta usando il AWSService RoleFor AWSBatch WithSagemaker ruolo, puoi provare a eliminarlo. Se il servizio utilizza il ruolo, il ruolo non verrà eliminato. È possibile visualizzare le regioni in cui viene utilizzato il ruolo. 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 ai servizi.

Per rimuovere AWS Batch le risorse utilizzate dal ruolo collegato al AWSService RoleFor AWSBatch WithSagemaker servizio

È necessario dissociare tutte le code di lavoro da tutti gli ambienti di servizio, quindi è necessario eliminare tutti gli ambienti di servizio che utilizzano il AWSService RoleFor AWSBatch WithSagemaker ruolo in tutte le AWS regioni prima di poter eliminare il ruolo. AWSService RoleFor AWSBatch WithSagemaker

1. Apri la console all' AWS Batch indirizzo. https://console.aws.amazon.com/batch/
2. Seleziona la Regione da utilizzare nella barra di navigazione.
3. Nel riquadro di navigazione, scegli Ambienti, quindi Ambienti di servizio.
4. Seleziona tutti gli ambienti di servizio.
5. Scegliere Disabilita. Attendi che lo Stato passi a DISABILITATO.
6. Seleziona l'ambiente di servizio.
7. Scegli Elimina. Conferma di voler eliminare l'ambiente di servizio selezionando Elimina ambiente di servizio.
8. Ripeti i passaggi da 1 a 7 per tutti gli ambienti di servizio che utilizzano il ruolo collegato ai servizi in tutte le regioni.

Eliminazione di un ruolo collegato al servizio in IAM (Console)

Puoi utilizzare la console IAM per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (console)

1. Accedi Console di gestione AWS e apri la console IAM all'indirizzo. https://console.aws.amazon.com/iam/
2. Nel pannello di navigazione della console IAM seleziona Ruoli. Quindi seleziona la casella di controllo accanto a AWSService RoleFor AWSBatchWithSagemaker, non il nome o la riga stessa.
3. Scegli Delete role (Elimina ruolo).
4. Nella finestra di dialogo di conferma controlla i dati relativi all'ultimo accesso ai servizi, che indicano quando ognuno dei ruoli selezionati ha effettuato l'accesso a un Servizio AWS. In questo modo potrai verificare se il ruolo è attualmente attivo. Se desideri procedere, seleziona Yes, Delete (Sì, elimina) per richiedere l'eliminazione del ruolo collegato ai servizi.
5. Controlla le notifiche della console IAM per monitorare lo stato dell'eliminazione del ruolo collegato ai servizi. Poiché l'eliminazione del ruolo collegato ai servizi IAM è asincrona, una volta richiesta l'eliminazione del ruolo, il task di eliminazione può essere eseguito correttamente o meno.

   - Se il task viene eseguito correttamente, il ruolo viene rimosso dall'elenco e nella parte superiore della pagina viene visualizzata una notifica di completamento.
   - Se il task non viene eseguito correttamente, puoi scegliere View details (Visualizza dettagli) o View Resources (Visualizza risorse) dalle notifiche per capire perché l'eliminazione non è stata effettuata. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

     > **Note**
     >
     > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio può restituire tutte le risorse, solo alcune o nessuna.

   - Se il task non viene eseguito e la notifica non include un elenco di risorse, il servizio potrebbe non restituire questa informazione. Per scoprire come eliminare le risorse per quel servizio, consulta Servizi AWS che funzionano con IAM. Trova il servizio nella tabella e seleziona il

link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione di un ruolo collegato al servizio in IAM ()AWS CLI

È possibile utilizzare i comandi IAM di AWS Command Line Interface per eliminare un ruolo collegato al servizio.

Per eliminare un ruolo collegato ai servizi (CLI)

1. Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `deletion-task-id`dalla risposta per controllare lo stato del task di eliminazione. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, digita il seguente comando:

   ```
   $ aws iam delete-service-linked-role --role-name
    AWSServiceRoleForAWSBatchWithSagemaker
   ```

2. Digita il seguente comando per verificare lo stato del processo di eliminazione:

   ```
   $ aws iam get-service-linked-role-deletion-status --deletion-task-id deletion-task-id
   ```

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

   > **Note**
   >
   > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio potrebbe

restituire tutte le risorse, alcune. Oppure, potrebbe non riportare alcuna risorsa. Per informazioni su come ripulire le risorse per un servizio che non riporta alcuna risorsa, consulta AWS Servizi che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Eliminazione di un ruolo collegato al servizio in IAM (API)AWS

È possibile utilizzare l'API di IAM; per eliminare un ruolo collegato ai servizi.

Per eliminare un ruolo collegato ai servizi (API)

1. Per inviare una richiesta di eliminazione per un ruolo collegato ai servizi, chiamare DeleteServiceLinkedRole. Nella richiesta, specifica il nome del AWSService RoleFor AWSBatch WithSagemaker ruolo.

   Poiché un ruolo collegato ai servizi non può essere eliminato se è in uso o se a esso sono associate delle risorse, occorre inviare una richiesta di eliminazione. Se queste condizioni non sono soddisfatte, la richiesta può essere rifiutata. Acquisisci il valore di `DeletionTaskId`dalla risposta per controllare lo stato del task di eliminazione.
2. Chiamare GetServiceLinkedRoleDeletionStatus per controllare lo stato dell'eliminazione. Nella richiesta, specificare il `DeletionTaskId`.

   Lo stato di un task di eliminazione può essere `NOT_STARTED`, `IN_PROGRESS`, `SUCCEEDED`o `FAILED`. Se l'eliminazione non viene eseguita correttamente, la chiamata restituisce il motivo dell'errore per consentire all'utente di risolvere il problema. Se l'eliminazione non viene eseguita perché il ruolo sta utilizzando le risorse del servizio, la notifica include un elenco di risorse, se il servizio restituisce questa informazione. A questo punto puoi eliminare le risorse e richiedere nuovamente l'eliminazione.

   > **Note**
   >
   > In base alle informazioni restituite dal servizio, è possibile che sia necessario ripetere questo processo diverse volte. Ad esempio, il ruolo collegato ai servizi potrebbe utilizzare sei risorse e il servizio potrebbe restituire informazioni su cinque di esse. Se elimini le cinque risorse e richiedi nuovamente l'eliminazione del ruolo, l'eliminazione non viene eseguita correttamente e il servizio segnala la risorsa rimanente. Un servizio

può restituire tutte le risorse, solo alcune o nessuna. Per scoprire come eliminare le risorse per un servizio che non restituisce nessuna risorsa, consulta Servizi AWS che funzionano con IAM. Trova il servizio nella tabella e seleziona il link Yes (Sì) per visualizzare la documentazione relativa al ruolo collegato ai servizi per quel servizio.

Regioni supportate per i ruoli collegati ai servizi AWS Batch

AWS Batch supporta l'utilizzo di ruoli collegati al servizio in tutte le regioni in cui il servizio è disponibile. Per ulteriori informazioni, consulta la pagina relativa agli endpoint AWS Batch.

Ruolo dell'istanza Amazon ECS

AWS Batch gli ambienti di calcolo sono popolati con istanze di container Amazon ECS. Eseguono l'agente container Amazon ECS localmente. L'agente container Amazon ECS effettua chiamate a varie operazioni AWS API per tuo conto. Pertanto, le istanze di container che eseguono l'agente richiedono una politica e un ruolo IAM affinché questi servizi riconoscano che l'agente appartiene a te. È necessario creare un ruolo IAM e un profilo di istanza per le istanze del contenitore da utilizzare al momento del lancio. Altrimenti, non puoi creare un ambiente di calcolo e avviare istanze di container al suo interno. Questo requisito si applica alle istanze di container lanciate con o senza l'AMI ottimizzata Amazon ECS fornita da Amazon. Per ulteriori informazioni, consulta Ruolo dell'istanza Amazon ECS nella Guida per lo sviluppatore di Amazon Elastic Container Service.

Argomenti

- Verifica il ruolo dell'istanza Amazon ECS del tuo account

Verifica il ruolo dell'istanza Amazon ECS del tuo account

Il ruolo e il profilo dell'istanza di Amazon ECS vengono creati automaticamente durante la prima esecuzione della console. Tuttavia, puoi seguire questi passaggi per verificare se il tuo account ha già il ruolo e il profilo dell'istanza Amazon ECS. I passaggi seguenti spiegano anche come allegare la policy IAM gestita.

Tutorial: verifica la presenza di **`ecsInstanceRole`** nella console IAM

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.

3. Cerca l'elenco dei ruoli per `ecsInstanceRole`. Se il ruolo non esiste, usa i seguenti passaggi per crearlo.

   a. Selezionare Crea ruolo.

   b. Per Tipo di entità attendibile, seleziona Servizio AWS.

   c. Per i casi d'uso comuni, scegli EC2.

   d. Scegli Next (Successivo).

   e. Per le politiche di autorizzazione, cerca Amazon EC2 ContainerServicefor EC2 Role.

   f. Seleziona la casella di controllo accanto ad Amazon EC2 ContainerServicefor EC2 Role, quindi scegli Avanti.

   g. Per Role Name (Nome ruolo), digita `ecsInstanceRole`, quindi scegli Create Role (Crea ruolo).

   > **Note**
   >
   > Se utilizzi il per Console di gestione AWS creare un ruolo per Amazon EC2, la console crea un profilo di istanza con lo stesso nome del ruolo.

In alternativa, puoi utilizzare il AWS CLI per creare il ruolo `ecsInstanceRole` IAM. L'esempio seguente crea un ruolo IAM con una policy di fiducia e una policy AWS gestita.

Tutorial: Creare un ruolo IAM e un profilo di istanza (AWS CLI)

1. Crea la seguente politica di fiducia e salvala in un file di testo denominato`ecsInstanceRole-role-trust-policy.json`.

   JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": { "Service": "ec2.amazonaws.com"},
      "Action": "sts:AssumeRole"
    }
  ]
```

```
}
```

2. Utilizzate il comando create-role per creare il `ecsInstanceRole` ruolo. Specificate la posizione del file delle politiche di fiducia nel `assume-role-policy-document` parametro.

```
$ aws iam create-role \
    --role-name ecsInstanceRole \
    --assume-role-policy-document file://ecsInstanceRole-role-trust-policy.json
```

3. Utilizzate il create-instance-profilecomando per creare un profilo di istanza denominato`ecsInstanceRole`.

> **Note**
>
> È necessario creare ruoli e profili di istanza come azioni separate nell' AWS API AWS CLI and.

```
$ aws iam create-instance-profile --instance-profile-name ecsInstanceRole
```

Di seguito è riportata una risposta di esempio.

```
{
    "InstanceProfile": {
        "Path": "/",
        "InstanceProfileName": "ecsInstanceRole",
        "InstanceProfileId": "AIPAT46P5RDITREXAMPLE",
        "Arn": "arn:aws:iam::123456789012:instance-profile/ecsInstanceRole",
        "CreateDate": "2022-06-30T23:53:34.093Z",
        "Roles": [],    }
}
```

4. Utilizzate il comando add-role-to-instance-profile per aggiungere il `ecsInstanceRole` ruolo al profilo dell'`ecsInstanceRole`istanza.

```
aws iam add-role-to-instance-profile \
    --role-name ecsInstanceRole --instance-profile-name ecsInstanceRole
```

5. Utilizzate il attach-role-policycomando per allegare la policy `AmazonEC2ContainerServiceforEC2Role` AWS gestita al `ecsInstanceRole` ruolo.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/
AmazonEC2ContainerServiceforEC2Role \
    --role-name ecsInstanceRole
```

Ruolo della flotta di Amazon EC2 spot

Se crei un ambiente di elaborazione gestito che utilizza Amazon EC2 Spot Fleet Instances, devi creare la `AmazonEC2SpotFleetTaggingRole` policy. Questa politica concede a Spot Fleet l'autorizzazione ad avviare, etichettare e chiudere le istanze per tuo conto. Specificare il ruolo nella richiesta di parco istanze Spot. È inoltre necessario disporre dei ruoli AWSServiceRoleForEC2Spot e AWSServiceRoleForEC2SpotFleetcollegati ai servizi per Amazon EC2 Spot e Spot Fleet. Usa le seguenti istruzioni per creare tutti questi ruoli. Per ulteriori informazioni, consulta Using Service-Linked Roles e Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella IAM User Guide.

Argomenti

- Crea ruoli per la flotta Amazon EC2 Spot nel Console di gestione AWS
- Crea ruoli per la flotta Amazon EC2 Spot con AWS CLI

Crea ruoli per la flotta Amazon EC2 Spot nel Console di gestione AWS

Per creare il ruolo collegato ai servizi **`AmazonEC2SpotFleetTaggingRole`** IAM per Amazon EC2 Spot Fleet

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Per la gestione degli accessi, scegli Ruoli,
3. Per Ruoli, scegli Crea ruolo.
4. Da Seleziona entità attendibile per Tipo di entità affidabile, scegli Servizio AWS.
5. Per altri casi d'uso Servizi AWS, scegli EC2e poi scegli EC2 - Spot Fleet Tagging.
6. Scegli Next (Successivo).
7. Da Politiche di autorizzazione per il nome della politica, verifica. `AmazonEC2SpotFleetTaggingRole`
8. Scegli Next (Successivo).

9. Per Denominare, rivedere e creare:

   a. Per Nome ruolo, inserisci un nome per identificare il ruolo.

   b. Per Descrizione, inserisci una breve spiegazione della politica.

   c. (Facoltativo) Per il passaggio 1: seleziona le entità attendibili, scegli Modifica per modificare il codice.

   d. (Facoltativo) Per la Fase 2: Aggiungere le autorizzazioni, scegliete Modifica per modificare il codice.

   e. (Facoltativo) Per Aggiungi tag, scegli Aggiungi tag per aggiungere tag alla risorsa.

   f. Scegli Crea ruolo.

> **Note**
>
> In passato, esistevano due politiche gestite per il ruolo di Amazon EC2 Spot Fleet.
>
> - Amazon EC2 SpotFleetRole: questa è la politica gestita originale per il ruolo Spot Fleet. Tuttavia, non ti consigliamo più di utilizzarlo con AWS Batch. Questa policy non supporta il tagging delle istanze Spot negli ambienti di elaborazione, necessario per utilizzare il ruolo collegato al `AWSServiceRoleForBatch` servizio. Se in precedenza hai creato un ruolo di Spot Fleet con questa politica, applica la nuova politica consigliata a quel ruolo. Per ulteriori informazioni, consulta Istanze Spot non taggate al momento della creazione.
> - Amazon EC2 SpotFleetTaggingRole: questo ruolo fornisce tutte le autorizzazioni necessarie per etichettare le istanze Amazon EC2 Spot. Usa questo ruolo per consentire l'applicazione di tag alle istanze Spot nei tuoi AWS Batch ambienti di elaborazione.

Crea ruoli per la flotta Amazon EC2 Spot con AWS CLI

Per creare il ruolo Amazon EC2 SpotFleetTaggingRole IAM per gli ambienti di elaborazione della tua flotta Spot

1. Esegui il seguente comando con. AWS CLI

```
$ aws iam create-role --role-name AmazonEC2SpotFleetTaggingRole \
    --assume-role-policy-document '{
  "Version":"2012-10-17",
  "Statement": [
```

```
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "spotfleet.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}'
```

2. Per collegare la policy Amazon EC2 SpotFleetTaggingRole managed IAM al tuo EC2 SpotFleetTaggingRole ruolo Amazon, esegui il seguente comando con AWS CLI.

```
$ aws iam attach-role-policy \
  --policy-arn \
    arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole \
  --role-name \
    AmazonEC2SpotFleetTaggingRole
```

Per creare il ruolo collegato ai servizi **`AWSServiceRoleForEC2Spot`** IAM per Amazon Spot EC2

> **Note**
>
> Se il ruolo collegato al servizio `AWSServiceRoleForEC2Spot` IAM esiste già, viene visualizzato un messaggio di errore simile al seguente.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2Spot has been taken in this account,
>  please try a different suffix.
> ```

- Esegui il comando seguente con. AWS CLI

```
$ aws iam create-service-linked-role --aws-service-name spot.amazonaws.com
```

Per creare il ruolo collegato ai servizi **`AWSServiceRoleForEC2SpotFleet`** IAM per Amazon EC2 Spot Fleet

> **Note**
>
> Se il ruolo collegato al servizio `AWSServiceRoleForEC2SpotFleet` IAM esiste già, viene visualizzato un messaggio di errore simile al seguente.
>
> ```
> An error occurred (InvalidInput) when calling the CreateServiceLinkedRole
>  operation:
> Service role name AWSServiceRoleForEC2SpotFleet has been taken in this account,
>  please try a different suffix.
> ```

- Esegui il comando seguente con. AWS CLI

```
$ aws iam create-service-linked-role --aws-service-name spotfleet.amazonaws.com
```

EventBridge Ruolo IAM

Amazon EventBridge offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle AWS risorse. AWS Batch i posti di lavoro sono disponibili come obiettivi EventBridge . Grazie a semplici regole rapidamente configurabili, puoi abbinare eventi e inviargli processi di AWS Batch in risposta. Prima di poter inviare AWS Batch lavori con EventBridge regole e obiettivi, EventBridge devi disporre delle autorizzazioni per eseguire AWS Batch lavori per tuo conto.

> **Note**
>
> Quando crei una regola nella EventBridge console che specifica una AWS Batch coda come destinazione, puoi creare questo ruolo. Per un esempio di procedura guidata, consulta AWS Batch i posti di lavoro come EventBridge obiettivi. Puoi creare il EventBridge ruolo manualmente utilizzando la console IAM. Per istruzioni, consulta Creazione di un ruolo utilizzando policy di fiducia personalizzate (console) nella Guida per l'utente IAM.

La relazione di fiducia per il tuo ruolo EventBridge IAM deve fornire al responsabile del `events.amazonaws.com` servizio la capacità di assumere il ruolo.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "events.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Assicurati che la policy allegata al tuo ruolo EventBridge IAM consenta `batch:SubmitJob` le autorizzazioni sulle tue risorse. Nell'esempio seguente, AWS Batch fornisce la politica `AWSBatchServiceEventTargetRole` gestita per fornire queste autorizzazioni.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
       ],
      "Resource": "*"
    }
  ]
}
```

Crea un cloud privato virtuale

Le risorse di elaborazione nei tuoi ambienti di elaborazione richiedono l'accesso alla rete esterna per comunicare con gli endpoint del AWS Batch servizio Amazon ECS. Tuttavia, potresti avere lavori che desideri eseguire in sottoreti private. Per avere la flessibilità necessaria per eseguire lavori in una sottorete pubblica o privata, crea un VPC con sottoreti sia pubbliche che private.

Puoi usare Amazon Virtual Private Cloud (Amazon VPC) per lanciare AWS risorse in una rete virtuale definita da te. Questo argomento fornisce un collegamento alla procedura guidata di Amazon VPC e un elenco delle opzioni da selezionare.

Crea un VPC

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

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

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

Fasi successive

Dopo aver creato il tuo VPC, considera i seguenti passaggi successivi:

- Crea gruppi di sicurezza per le tue risorse pubbliche e private se richiedono l'accesso di rete in entrata. Per ulteriori informazioni, consulta Utilizzo dei gruppi di sicurezza nella Guida per l'utente di Amazon VPC.
- Crea un ambiente di elaborazione AWS Batch gestito che lanci risorse di elaborazione nel tuo nuovo VPC. Per ulteriori informazioni, consulta Crea un ambiente di elaborazione. Se utilizzi la procedura guidata per la creazione dell'ambiente di calcolo nella AWS Batch console, puoi specificare il VPC appena creato e le sottoreti pubbliche o private in cui desideri avviare le tue istanze.
- Crea una coda di AWS Batch lavoro mappata al tuo nuovo ambiente di calcolo. Per ulteriori informazioni, consulta Creare una coda di lavoro.
- Crea una definizione per l'esecuzione dei processi. Per ulteriori informazioni, consulta Creare una definizione di processo a nodo singolo .
- Invia un processo con la definizione del processo alla nuova coda dei processi. Questo lavoro arriva nell'ambiente di elaborazione che hai creato con il tuo nuovo VPC e le tue sottoreti. Per ulteriori informazioni, consulta Tutorial: invia un lavoro.

Usa un endpoint di interfaccia per Access AWS Batch

Puoi usarlo AWS PrivateLink per creare una connessione privata tra il tuo VPC e. AWS Batch Puoi accedere AWS Batch come se fosse nel tuo VPC, senza l'uso di un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per accedervi. AWS Batch

Stabilisci questa connessione privata creando un endpoint di interfaccia attivato da AWS PrivateLink. In ciascuna sottorete viene creata un'interfaccia di rete endpoint da abilitare per l'endpoint di interfaccia. Queste sono interfacce di rete gestite dal richiedente che fungono da punto di ingresso per il traffico destinato a AWS Batch.

Per ulteriori informazioni, consulta Interface VPC Endpoints nella Guida.AWS PrivateLink

Considerazioni per AWS Batch

Prima di configurare un endpoint di interfaccia per AWS Batch, consulta le proprietà e le limitazioni dell'endpoint dell'interfaccia nella Guida.AWS PrivateLink

AWS Batch supporta l'esecuzione di chiamate a tutte le sue azioni API tramite l'endpoint dell'interfaccia.

Prima di configurare gli endpoint VPC dell'interfaccia per AWS Batch, tieni presente le seguenti considerazioni:

- I lavori che utilizzano il tipo di avvio delle risorse Fargate non richiedono l'interfaccia VPC endpoint per Amazon ECS, ma potrebbero essere necessari endpoint VPC di interfaccia per Amazon ECR, Secrets Manager o AWS Batch Amazon Logs descritti nei punti seguenti. CloudWatch
  - Per eseguire i job, devi creare gli endpoint VPC dell'interfaccia per Amazon ECS. Per ulteriori informazioni, consulta Interface VPC Endpoints (AWS PrivateLink) nella Amazon Elastic Container Service Developer Guide.
  - Per consentire ai tuoi lavori di estrarre immagini private da Amazon ECR, devi 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.
  - Per consentire ai lavori di estrarre dati sensibili da Secrets Manager, è necessario creare gli endpoint VPC di interfaccia per Secrets Manager. Per ulteriori informazioni, consulta Utilizzo di Secrets Manager con endpoint VPC nella Guida per l'utente di Gestione dei segreti AWS .
  - Se il tuo VPC non dispone di un gateway Internet e i tuoi job utilizzano il driver di `awslogs` registro per inviare le informazioni di registro ai CloudWatch registri, devi creare un endpoint VPC di interfaccia per Logs. CloudWatch Per ulteriori informazioni, consulta Using CloudWatch Logs with Interface VPC Endpoints nella CloudWatch Amazon Logs User Guide.
- I lavori che utilizzano le EC2 risorse richiedono che le istanze del contenitore su cui vengono lanciate eseguano una versione `1.25.1` o successiva dell'agente container Amazon ECS. Per ulteriori informazioni, consulta le versioni degli agenti container di Amazon ECS Linux nella Amazon Elastic Container Service Developer 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 a AWS Batch.
- 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 443 dalla sottorete privata del VPC.
- AWS Batch non supporta gli endpoint dell'interfaccia VPC nei seguenti casi: Regioni AWS
  - Asia Pacifico (Osaka-Locale) (`ap-northeast-3`)
  - Asia Pacifico (Giacarta) (`ap-southeast-3`)

Crea un endpoint di interfaccia per AWS Batch

Puoi creare un endpoint di interfaccia per AWS Batch utilizzare la console Amazon VPC o AWS Command Line Interface ().AWS CLI Per ulteriori informazioni, consulta la sezione Creazione di un endpoint di interfaccia nella Guida per l'utente di AWS PrivateLink .

Crea un endpoint di interfaccia per AWS Batch utilizzare i seguenti nomi di servizio:

- com.amazonaws. *region*.batch
- com.amazonaws.it. *region*.batch-fips (per gli endpoint conformi a FIPS , consulta endpoint e quote)AWS Batch

Esempio:

```
com.amazonaws.us-east-2.batch
```

```
com.amazonaws.us-east-2.batch-fips
```

Nella partizione, `aws-cn` il formato è diverso:

```
cn.com.amazonaws.region.batch
```

Esempio:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Nomi DNS privati per AWS Batch gli endpoint di interfaccia

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi utilizzare nomi DNS specifici a cui connetterti. Forniamo queste opzioni AWS Batch:

- lotto. *region*.amazonaws.com
- lotto. *region*.api.aws

Per gli endpoint conformi allo standard FIPS:

- batch-fips. *region*.api.aws
- fips.batch. *region*.amazonaws.com non è supportato

Per maggiori informazioni, consulta Accedere a un servizio tramite un endpoint di interfaccia nella Guida.AWS PrivateLink

Creazione di una policy dell' endpoint per l'endpoint dell'interfaccia

Una policy dell'endpoint è una risorsa IAM che è possibile allegare all'endpoint dell'interfaccia. La policy predefinita per gli endpoint consente l'accesso completo AWS Batch tramite l'endpoint dell'interfaccia. Per controllare l'accesso consentito ad AWS Batch dal VPC, collega una policy di endpoint personalizzata all'endpoint di interfaccia.

Una policy di endpoint specifica le informazioni riportate di seguito:

- I principali che possono eseguire azioni (utenti e Account AWS ruoli IAM).
- Le azioni che possono essere eseguite.
- Le risorse in cui è possibile eseguire le operazioni.

Per ulteriori informazioni, consulta la sezione Controllo dell'accesso ai servizi con policy di endpoint nella Guida di AWS PrivateLink .

Esempio: policy degli endpoint VPC per le azioni AWS Batch

Di seguito è riportato l'esempio di una policy dell'endpoint personalizzata. Quando alleghi questa policy all'endpoint dell'interfaccia, concede l'accesso alle AWS Batch azioni elencate per tutti i principali su tutte le risorse.

```
{
   "Statement": [
      {
         "Principal": "*",
         "Effect": "Allow",
```

```
         "Action": [
            "batch:SubmitJob",
            "batch:ListJobs",
            "batch:DescribeJobs"
         ],
         "Resource":"*"
      }
   ]
}
```

Convalida della conformità per AWS Batch

Per sapere se un Servizio AWS programma rientra nell'ambito di specifici programmi di conformità, consulta Servizi AWS la sezione Scope by Compliance Program Servizi AWS e scegli il programma di conformità che ti interessa. Per informazioni generali, consulta Programmi di AWS conformità Programmi di di .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. Per ulteriori informazioni sulla responsabilità di conformità durante l'utilizzo Servizi AWS, consulta la Documentazione AWS sulla sicurezza.

Sicurezza dell'infrastruttura in AWS Batch

In quanto servizio gestito, AWS Batch 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.

Utilizzate chiamate API AWS pubblicate per accedere AWS Batch attraverso la rete. I client devono supportare quanto segue:

- Transport Layer Security (TLS). È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Suite di cifratura con Perfect Forward Secrecy (PFS), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni, come Java 7 e versioni successive, supporta tali modalità.

Puoi richiamare queste operazioni API da qualsiasi posizione di rete, AWS Batch ma supportano politiche di accesso basate sulle risorse, che possono includere restrizioni basate sull'indirizzo IP di origine. Puoi anche utilizzare AWS Batch le policy per controllare l'accesso da endpoint Amazon Virtual Private Cloud (Amazon VPC) specifici o specifici. VPCs In effetti, questo isola l'accesso alla rete a una determinata AWS Batch risorsa solo dal VPC specifico all'interno AWS della rete.

Prevenzione del problema "confused deputy" tra servizi

Il confused deputy è un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire un'azione può costringere un'entità dotata di privilegi maggiori a eseguire l'azione. Nel AWS, l'impersonificazione intersettoriale può portare al confuso problema del vice. La rappresentazione tra servizi può verificarsi quando un servizio (il servizio chiamante) effettua una chiamata a un altro servizio (il servizio chiamato). Il servizio chiamante può essere manipolato per utilizzare le proprie autorizzazioni e agire sulle risorse di un altro cliente, a cui normalmente non avrebbe accesso. Per evitare ciò, AWS fornisce strumenti per poterti a proteggere i tuoi dati per tutti i servizi con entità di servizio a cui è stato concesso l'accesso alle risorse dell'account.

Si consiglia di utilizzare `aws:SourceArn`le chiavi di contesto della condizione `aws:SourceAccount`globale nelle politiche delle risorse per limitare le autorizzazioni che AWS Batch forniscono un altro servizio alla risorsa. Se il valore `aws:SourceArn` non contiene l'ID account, ad esempio un ARN di un bucket Amazon S3, è necessario utilizzare entrambe le chiavi di contesto delle condizioni globali per limitare le autorizzazioni. Se si utilizzano entrambe le chiavi di contesto delle condizioni globali e il valore `aws:SourceArn` contiene l'ID account, il valore `aws:SourceAccount` e l'account nel valore `aws:SourceArn` deve utilizzare lo stesso ID account nella stessa dichiarazione di policy. Utilizzare `aws:SourceArn` se si desidera consentire l'associazione di una sola risorsa all'accesso tra servizi. Utilizzare `aws:SourceAccount` se si desidera consentire l'associazione di qualsiasi risorsa in tale account all'uso tra servizi.

Il valore di `aws:SourceArn` deve essere la risorsa che AWS Batch memorizza.

Il modo più efficace per proteggersi dal problema "confused deputy" è quello di utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con l'ARN completo della risorsa. Se non si conosce l'ARN completo della risorsa o si scelgono più risorse, utilizzare la chiave di contesto della condizione globale `aws:SourceArn` con caratteri jolly (*) per le parti sconosciute dell'ARN. Ad esempio, `arn:aws:`*`servicename`*`:*:`*`123456789012`*`:*`.

Gli esempi seguenti mostrano come utilizzare le chiavi di contesto `aws:SourceArn` e `aws:SourceAccount` global condition AWS Batch per prevenire il confuso problema del vice.

Esempio: ruolo per l'accesso a un solo ambiente di elaborazione

Il ruolo seguente può essere utilizzato solo per accedere a un ambiente di calcolo. Il nome del processo deve essere specificato in * quanto la coda dei lavori può essere associata a più ambienti di elaborazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/testCE",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Esempio: ruolo per l'accesso a più ambienti di elaborazione

Il seguente ruolo può essere utilizzato per accedere a più ambienti di elaborazione. Il nome del processo deve essere specificato in * quanto la coda dei lavori può essere associata a più ambienti di elaborazione.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "batch.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": [
            "arn:aws:batch:us-east-1:123456789012:compute-environment/*",
            "arn:aws:batch:us-east-1:123456789012:job/*"
          ]
        }
      }
    }
  ]
}
```

Registrazione delle chiamate API con AWS BatchAWS CloudTrail

AWS Batch è integrato con AWS CloudTrail, un servizio che fornisce una registrazione delle azioni intraprese da un utente, un ruolo o un AWS servizio in AWS Batch. CloudTrail acquisisce tutte le chiamate API AWS Batch come eventi. Le chiamate acquisite includono chiamate dalla AWS Batch console e chiamate di codice alle operazioni AWS Batch API. Se crei un trail, puoi abilitare la distribuzione continua di CloudTrail eventi a un bucket Amazon S3, inclusi gli eventi per. AWS Batch 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, è possibile determinare a quale richiesta è stata inviata AWS Batch, l'indirizzo IP da cui è stata effettuata la richiesta, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

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

Argomenti

AWS Batch informazioni in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività in AWS Batch, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. Puoi visualizzare, cercare e scaricare gli 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 di AWS Batch, crea un percorso. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando crei un percorso nella console, il percorso si applica a tutte le AWS 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 gli argomenti seguenti:

- 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 AWS Batch azioni vengono registrate CloudTrail e documentate nella versione più recente/ /. https://docs.aws.amazon.com/batch/ APIReference Ad esempio, le chiamate alle sezioni `SubmitJob`, `ListJobs` e `DescribeJobs` generano voci nei file di log 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 dell'utente IAM o root.
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.

- Se la richiesta è stata effettuata da un altro servizio. AWS

Per ulteriori informazioni, consulta Elemento CloudTrail userIdentity.

Riferimento: informazioni sulle voci dei file di AWS Batch registro

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 da qualsiasi sorgente e include informazioni sull'azione richiesta, la data e l'ora dell'operazione, i parametri della richiesta e così via. I file di log di CloudTrail non sono una traccia di stack ordinata delle chiamate API pubbliche, pertanto non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'`CreateComputeEnvironment`azione.


```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:admin",
    "arn": "arn:aws:sts::012345678910:assumed-role/Admin/admin",
    "accountId": "012345678910",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2017-12-20T00:48:46Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::012345678910:role/Admin",
        "accountId": "012345678910",
        "userName": "Admin"
      }
    }
  },
  "eventTime": "2017-12-20T00:48:46Z",
  "eventSource": "batch.amazonaws.com",
  "eventName": "CreateComputeEnvironment",
```

```
  "awsRegion": "us-east-1",
  "sourceIPAddress": "203.0.113.1",
  "userAgent": "aws-cli/1.11.167 Python/2.7.10 Darwin/16.7.0 botocore/1.7.25",
  "requestParameters": {
    "computeResources": {
      "subnets": [
        "subnet-5eda8e04"
      ],
      "tags": {
        "testBatchTags": "CLI testing CE"
      },
      "desiredvCpus": 0,
      "minvCpus": 0,
      "instanceTypes": [
        "optimal"
      ],
      "securityGroupIds": [
        "sg-aba9e8db"
      ],
      "instanceRole": "ecsInstanceRole",
      "maxvCpus": 128,
      "type": "EC2"
    },
    "state": "ENABLED",
    "type": "MANAGED",
    "computeEnvironmentName": "Test"
  },
  "responseElements": {
    "computeEnvironmentName": "Test",
    "computeEnvironmentArn": "arn:aws:batch:us-east-1:012345678910:compute-environment/
Test"
  },
  "requestID": "890b8639-e51f-11e7-b038-EXAMPLE",
  "eventID": "874f89fa-70fc-4798-bc00-EXAMPLE",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "012345678910"
}
```


Risolvi i problemi AWS Batch di IAM

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con AWS Batch IAM.

Argomenti

Non sono autorizzato a eseguire un'operazione in AWS Batch

Se ti Console di gestione AWS dice che non sei autorizzato a eseguire un'azione, devi contattare l'amministratore per ricevere assistenza. L'amministratore è la persona da cui si sono ricevuti il nome utente e la password.

Il seguente esempio di errore si verifica quando l'utente `mateojackson` prova a utilizzare la console per visualizzare i dettagli relativi a una risorsa *`my-example-widget`* fittizia, ma non dispone di autorizzazioni `batch:`*`GetWidget`* fittizie.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 batch:GetWidget on resource: my-example-widget
```

In questo caso, Mateo richiede al suo amministratore di aggiornare le policy per poter accedere alla risorsa *`my-example-widget`* utilizzando l'operazione `batch:`*`GetWidget`*. Per ulteriori informazioni sulla concessione delle autorizzazioni per il trasferimento di un ruolo, vedi Concessione a un utente delle autorizzazioni per trasferire un ruolo a un servizio. AWS

Non sono autorizzato a eseguire iam: PassRole

Se ricevi un errore che indica che non sei autorizzato a eseguire l'operazione `iam:PassRole`, le tue policy devono essere aggiornate per poter passare un ruolo a AWS Batch.

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 AWS Batch. Tuttavia, l'operazione richiede che il servizio disponga delle autorizzazioni concesse da un ruolo di servizio. Mary non dispone delle autorizzazioni per trasmettere il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS Batch risorse

È possibile creare un ruolo con il quale utenti in altri account o persone esterne all'organizzazione possono accedere alle tue risorse. È possibile specificare chi è attendibile per l'assunzione del ruolo. Per i servizi che supportano politiche basate sulle risorse o liste di controllo degli accessi (ACLs), puoi utilizzare tali politiche per concedere alle persone l'accesso alle tue risorse.

Per maggiori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Batch supporta queste funzionalità, consulta. Come AWS Batch funziona con IAM
- Per scoprire come fornire l'accesso alle tue risorse attraverso Account AWS le risorse di tua proprietà, consulta Fornire l'accesso a un utente IAM in un altro Account AWS di tua proprietà nella IAM User Guide.
- Per scoprire come fornire l'accesso alle tue risorse a terze parti Account AWS, consulta Fornire l'accesso a soggetti Account AWS di proprietà di terze parti nella Guida per l'utente IAM.
- Per informazioni su come fornire l'accesso tramite la federazione delle identità, consulta Fornire l'accesso a utenti autenticati esternamente (federazione delle identità) nella Guida per l'utente IAM.
- Per informazioni sulle differenze di utilizzo tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Accesso a risorse multi-account in IAM nella Guida per l'utente IAM.

AWS Step Functions

È possibile utilizzare la AWS Batch console per visualizzare i dettagli sulle macchine a stati Step Functions e sulle funzioni che utilizzano.

Sections

Tutorial: Visualizza i dettagli delle macchine a stati

La AWS Batch console visualizza un elenco delle macchine a stati correnti Regione AWS che contengono almeno una fase del flusso di lavoro che invia un AWS Batch lavoro.

Scegliere una macchina a stati per visualizzare una rappresentazione grafica del flusso di lavoro. I passaggi evidenziati in blu rappresentano i AWS Batch lavori. Utilizza i controlli del grafico per ingrandire, ridurre e centrare il grafico.

> **Note**
>
> Quando si fa riferimento dinamicamente a un AWS Batch lavoro JsonPath nella definizione della macchina a stati, i dettagli della funzione non possono essere visualizzati nella AWS Batch console. Il nome del processo viene invece elencato come riferimento dinamico e i passaggi corrispondenti nel grafico sono visualizzati in grigio.

Per visualizzare i dettagli della macchina a stati

1. Apri la pagina Workflow orchestration della AWS Batch console con tecnologia Step Functions.
2. Scegliere una macchina a stati.
   <result>

   La AWS Batch console apre la pagina Dettagli.
   </result>

Per ulteriori informazioni, consulta Step Functions nella la Guida per sviluppatori di AWS Step Functions .

Tutorial: modificare una macchina a stati

Quando si desidera modificare una macchina a stati, AWS Batch apre la pagina Modifica definizione della console Step Functions.

Per modificare una macchina a stati

1. Apri la pagina Workflow orchestration della AWS Batch console con tecnologia Step Functions.
2. Scegliere una macchina a stati.
3. Scegli Modifica.

   La console Step Functions apre la pagina Modifica definizione.
4. Modificare la macchina a stati e scegliere Salva.

Per ulteriori informazioni sulla modifica delle macchine a stati, vedere la lingua della macchina a stati Step Functions nella Guida per gli sviluppatori di AWS Step Functions .

Tutorial: Esegui una macchina a stati

Quando si desidera eseguire una macchina a stati, AWS Batch apre la pagina Nuova esecuzione della console Step Functions.

Per eseguire una macchina a stati

1. Apri la pagina Workflow orchestration della AWS Batch console con tecnologia Step Functions.
2. Scegliere una macchina a stati.
3. Scegli Execute (Esegui).

   La console Step Functions apre la pagina Nuova esecuzione.
4. (Facoltativo) Modificare la macchina a stati e scegliere Avvia esecuzione.

Per ulteriori informazioni sull'esecuzione di macchine a stati, vedere Concetti di esecuzione macchina a stati Step Functions nella Guida per gli sviluppatori di AWS Step Functions .

AWS Batch flusso di eventi per Amazon EventBridge

Puoi utilizzare lo stream di AWS Batch eventi per Amazon per EventBridge ricevere notifiche quasi in tempo reale sullo stato attuale dei lavori nelle tue code di lavoro.

Puoi utilizzarlo EventBridge per ottenere ulteriori informazioni sul tuo AWS Batch servizio. Più specificamente, puoi utilizzarlo per controllare lo stato di avanzamento dei lavori, creare flussi di lavoro AWS Batch personalizzati, generare report o metriche sull'utilizzo o creare dashboard personalizzate. Con AWS Batch e EventBridge, non è necessario pianificare e monitorare un codice che effettui continuamente sondaggi AWS Batch per rilevare eventuali modifiche allo stato delle mansioni. Puoi invece gestire le modifiche dello stato del AWS Batch lavoro in modo asincrono utilizzando una varietà di obiettivi Amazon. EventBridge Questi includono AWS Lambda Amazon Simple Queue Service, Amazon Simple Notification Service o Amazon Kinesis Data Streams.

È garantito che AWS Batch gli eventi del flusso di eventi vengano consegnati almeno una volta. Nel caso di invio di eventi duplicati, l'evento fornisce informazioni sufficienti a identificare i duplicati. In questo modo, puoi confrontare la data e l'ora dell'evento e lo stato del lavoro.

AWS Batch i posti di lavoro sono disponibili come EventBridge obiettivi. Utilizzando regole semplici, puoi abbinare gli eventi e inviare AWS Batch lavori in risposta ad essi. Per ulteriori informazioni, consulta Cos'è EventBridge? nella Amazon EventBridge User Guide. Puoi anche usarla EventBridge per pianificare azioni automatiche che si attivano automaticamente in determinati momenti utilizzando cron o valuta le espressioni. Per ulteriori informazioni, consulta la sezione Creazione di una EventBridge regola Amazon che viene eseguita secondo una pianificazione nella Amazon EventBridge User Guide. Per un esempio di procedura guidata, consulta AWS Batch i posti di lavoro come EventBridge obiettivi. Per informazioni sull'uso dello EventBridge Scheduler, consulta Configurazione di Amazon EventBridge Scheduler nella Amazon EventBridge User Guide.

Argomenti

- AWS Batch eventi
- Tutorial: utilizzare le notifiche AWS utente con AWS Batch
- AWS Batch i posti di lavoro come EventBridge obiettivi
- Tutorial: ascolta gli eventi AWS Batch di lavoro utilizzando EventBridge
- Tutorial: invio di avvisi di Amazon Simple Notification Service per eventi di lavoro non riusciti

AWS Batch eventi

AWS Batch invia gli eventi di modifica dello stato del lavoro a EventBridge. AWS Batch tiene traccia dello stato dei tuoi lavori. Se lo stato di un lavoro inviato in precedenza cambia, viene richiamato un evento. Ad esempio, se un lavoro nello `RUNNING` stato passa allo `FAILED` stato. Questi eventi vengono classificati come eventi di modifica dello stato del processo.

> **Note**
>
> AWS Batch potrebbe aggiungere altri tipi di eventi, fonti e dettagli in futuro. Se stai deserializzando a livello di codice i dati JSON degli eventi, assicurati che l'applicazione sia pronta a gestire proprietà sconosciute. Questo serve per evitare problemi se e quando vengono aggiunte queste proprietà aggiuntive.

Eventi di modifica dello stato del lavoro

Ogni volta che un job esistente (inviato in precedenza) cambia stato, viene creato un evento. Per ulteriori informazioni sugli stati AWS Batch lavorativi, vedereJob stati.

> **Note**
>
> Gli eventi non vengono creati per l'invio iniziale del lavoro.

Example Evento di modifica dello stato del processo

Gli eventi di modifica dello stato del lavoro vengono forniti nel seguente formato. La `detail` sezione è simile all'JobDetailoggetto restituito da un'operazione DescribeJobsAPI nell'AWS Batch API Reference. Per ulteriori informazioni sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

```
{
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job State Change",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
```

```
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "attempts": [],
        "createdAt": 1641944200058,
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "propagateTags": false,
```

```
        "platformCapabilities": []
    }
}
```

Eventi bloccati in Job queue

Ogni volta che AWS Batch rileva un lavoro nello `RUNNABLE` stato e quindi blocca una coda, viene creato un evento in Amazon Events. CloudWatch Per ulteriori informazioni sulle cause supportate per le code bloccate, consulta. Lavori bloccati in uno RUNNABLE status Lo stesso motivo è disponibile anche nel `statusReason` campo dell'azione `DescribeJobs`API.

Example Evento bloccato Job queue

Gli eventi bloccati di Job queue vengono consegnati nel seguente formato. La *`detail`* sezione è simile all'JobDetailoggetto restituito da un'operazione DescribeJobsAPI nell'API Reference AWS Batch . Per ulteriori informazioni sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

```
{
    "version": "0",
    "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
    "detail-type": "Batch Job Queue Blocked",
    "source": "aws.batch",
    "account": "123456789012",
    "time": "2022-01-11T23:36:40Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "arn:aws:batch:us-east-1:123456789012:job-queue/PexjEHappyPathCanary2JobQueue"
    ],
    "detail": {
        "jobArn": "arn:aws:batch:us-east-1:123456789012:job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
        "jobName": "event-test",
        "jobId": "4c7599ae-0a82-49aa-ba5a-4727fcce14a8",
        "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/
PexjEHappyPathCanary2JobQueue",
        "status": "RUNNABLE",
        "statusReason": "blocked-reason",
        "attempts": [],
        "createdAt": 1641944200058,
```

```
        "retryStrategy": {
            "attempts": 2,
            "evaluateOnExit": []
        },
        "dependsOn": [],
        "jobDefinition": "arn:aws:batch:us-east-1:123456789012:job-definition/first-
run-job-definition:1",
        "parameters": {},
        "container": {
            "image": "137112412989.dkr.ecr.us-east-1.amazonaws.com/amazonlinux:latest",
            "command": [
                "sleep",
                "600"
            ],
            "volumes": [],
            "environment": [],
            "mountPoints": [],
            "ulimits": [],
            "networkInterfaces": [],
            "resourceRequirements": [
                {
                    "value": "2",
                    "type": "VCPU"
                }, {
                    "value": "256",
                    "type": "MEMORY"
                }
            ],
            "secrets": []
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
}
```

Eventi di modifica dello stato del lavoro del servizio

Ogni volta che un job di servizio esistente cambia stato, viene creato un evento. Per ulteriori informazioni sugli stati dei processi di assistenza, vedereMappatura dello stato AWS Batch del lavoro del servizio allo stato dell' SageMaker IA.

> **Note**
>
> Gli eventi non vengono creati per l'invio iniziale del lavoro.

Example Evento Service Job State Change

Gli eventi di modifica dello stato del lavoro di servizio vengono forniti nel seguente formato. La `detail` sezione è simile alla risposta restituita da un'operazione DescribeServiceJobAPI nell'AWS Batch API Reference. Per ulteriori informazioni sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

> **Note**
>
> I `serviceRequestPayload` campi `tags` e non sono inclusi nell'evento`detail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job State Change",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [
      {
        "serviceResourceId": {
          "name": "TrainingJobArn",
          "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-
training-job88b610a69aa8380ca5b0a7aba3f81cb8"
        },
        "startedAt": 1641944300058,
        "stoppedAt": 1641944400058,
        "statusReason": "Received status from SageMaker: Training job completed"
      }
```

```
    ],
    "createdAt": 1641944200058,
    "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
    "jobId": "0bb17543-ece6-4480-b1a7-a556d344746b",
    "jobName": "event-test",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
    "latestAttempt": {
      "serviceResourceId": {
        "name": "TrainingJobArn",
        "value": "arn:aws:sagemaker:us-east-1:123456789012:training-job/AWSBatchmy-
training-job88b610a69aa8380ca5b0a7aba3f81cb8"
      }
    },
    "serviceJobType": "SAGEMAKER_TRAINING",
    "startedAt": 1641944300058,
    "status": "SUCCEEDED",
    "statusReason": "Received status from SageMaker: Training job completed",
    "stoppedAt": 1641944400058,
    "timeoutConfig": {
      "attemptDurationSeconds": 60
    }
  }
}
```

Eventi bloccati nella coda dei lavori di servizio

Ogni volta che AWS Batch rileva una coda bloccata, viene creato un evento in Amazon Events. CloudWatch Il motivo della coda bloccata è disponibile nel `statusReason` campo dell'azione API. `DescribeServiceJob`

Example Evento bloccato Service Job Queue

Gli eventi bloccati di Service Job Queue vengono forniti nel seguente formato. La *`detail`* sezione è simile alla risposta restituita dall'operazione DescribeServiceJobAPI nell'API Reference AWS Batch . Per ulteriori informazioni sui EventBridge parametri, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

> **Note**
>
> I `serviceRequestPayload` campi `tags` e non sono inclusi nell'evento`detail`.

```
{
  "version": "0",
  "id": "c8f9c4b5-76e5-d76a-f980-7011e206042b",
  "detail-type": "Batch Service Job Queue Blocked",
  "source": "aws.batch",
  "account": "123456789012",
  "time": "2022-01-11T23:36:40Z",
  "region": "us-east-1",
  "resources": [
    "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8"
  ],
  "detail": {
    "attempts": [],
    "createdAt": 1641944200058,
    "jobArn": "arn:aws:batch:us-east-1:123456789012:service-job/4c7599ae-0a82-49aa-
ba5a-4727fcce14a8",
    "jobId": "6271dfdf-d8a7-41b1-a4d2-55a2224f5375",
    "jobName": "event-test",
    "jobQueue": "arn:aws:batch:us-east-1:123456789012:job-queue/HappyPathJobQueue",
    "serviceJobType": "SAGEMAKER_TRAINING",
    "status": "RUNNABLE",
    "statusReason": "blocked-reason",
    "timeoutConfig": {
      "attemptDurationSeconds": 60
    }
  }
}
```

Tutorial: utilizzare le notifiche AWS utente con AWS Batch

Puoi utilizzare le notifiche AWS utente per configurare i canali di consegna per ricevere notifiche sugli AWS Batch eventi. L'utente riceverà una notifica quando un evento corrisponde a una regola specificata. Puoi ricevere notifiche relative agli eventi attraverso più canali, tra cui e-mail, Amazon Q Developer in applicazioni di chat, notifiche chat o notifiche AWS Console Mobile Applicationpush. È anche possibile visualizzare le notifiche nel Centro notifiche della console. La funzionalità Notifiche all'utente supporta l'aggregazione, che può ridurre il numero di notifiche ricevute durante eventi specifici.

Per configurare le notifiche utente in AWS Batch:

1. Apri la AWS Batch console.
2. Seleziona Dashboard (Pannello di controllo).
3. Scegli Configura notifiche.
4. In Notifiche AWS utente, scegli Crea configurazione di notifica.

Per ulteriori informazioni su come configurare e visualizzare le notifiche utente, consulta Introduzione alle notifiche AWS utente.

AWS Batch i posti di lavoro come EventBridge obiettivi

Amazon EventBridge offre un flusso quasi in tempo reale di eventi di sistema che descrivono i cambiamenti nelle risorse di Amazon Web Services. In genere, AWS Batch su Amazon Elastic Container Service, Amazon Elastic Kubernetes Service AWS e Fargate sono disponibili lavori come obiettivi. EventBridge Utilizzando semplici regole, puoi abbinare gli eventi e inviare AWS Batch lavori in risposta ad essi. Per ulteriori informazioni, vedi Cos'è EventBridge? nella Amazon EventBridge User Guide.

Puoi anche usarlo EventBridge per pianificare azioni automatiche che vengono richiamate in determinati momenti utilizzando cron o valuta le espressioni. Per ulteriori informazioni, consulta Creazione di una EventBridge regola Amazon che viene eseguita secondo una pianificazione nella Amazon EventBridge User Guide.

Per informazioni su come creare una regola che viene eseguita quando un evento corrisponde a un modello di evento, consulta Creazione di EventBridge regole Amazon che reagiscono agli eventi nella Amazon EventBridge User Guide.

I casi d'uso più comuni AWS Batch per Job as a EventBridge Target includono i seguenti casi d'uso:

- Un processo pianificato viene eseguito a intervalli di tempo regolari. Ad esempio, un cron processo si verifica solo durante le ore di utilizzo ridotto, quando le istanze Amazon EC2 Spot sono meno costose.
- Un AWS Batch processo viene eseguito in risposta a un'operazione API che ha effettuato l'accesso. CloudTrail Ad esempio, un lavoro viene inviato ogni volta che un oggetto viene caricato in un bucket Amazon S3 specificato. Ogni volta che ciò accade, il trasformatore EventBridge di input passa il nome del bucket e della chiave dell'oggetto ai parametri. AWS Batch

> **Note**
>
> In questo scenario, tutte le AWS risorse correlate devono trovarsi nella stessa regione. Ciò include risorse come il bucket, le EventBridge regole e i log di Amazon S3. CloudTrail

Prima di poter inviare AWS Batch lavori con EventBridge regole e obiettivi, il EventBridge servizio richiede diverse autorizzazioni per eseguire i lavori. AWS Batch Quando crei una regola nella EventBridge console che specifica un AWS Batch lavoro come destinazione, puoi creare anche questo ruolo. Per ulteriori informazioni sul principale del servizio richiesto e le autorizzazioni IAM per questo ruolo, consulta EventBridge Ruolo IAM.

Argomenti

- Tutorial: Creare un AWS Batch lavoro pianificato
- Tutorial: crea una regola con uno schema di eventi
- Tutorial: passa le informazioni sugli eventi a un AWS Batch target in base a una pianificazione utilizzando il trasformatore EventBridge di ingresso

Tutorial: Creare un AWS Batch lavoro pianificato

La procedura seguente illustra come creare un AWS Batch lavoro pianificato e il ruolo EventBridge IAM richiesto.

Per creare un AWS Batch lavoro pianificato con EventBridge

> **Note**
>
> Questa procedura è valida per tutti i AWS Batch job di Amazon ECS, Amazon EKS e AWS Fargate.

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel pannello di navigazione, scegli Regole.
4. Scegli Crea regola.

5. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 64 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   > **Note**
   >
   > Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.

6. (Facoltativo) In Descrizione, inserisci una descrizione per la regola.
7. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un Servizio AWS utente del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.
8. (Facoltativo) Disattiva la regola sul bus selezionato se non desideri eseguirla immediatamente.
9. Per Rule type (Tipo di regola), scegli Schedule (Pianifica).
10. Scegli Continua per creare la regola o Avanti.
11. Per Schedule pattern (Modello di pianificazione), esegui una delle seguenti operazioni:

    - Scegli Una pianificazione dettagliata che viene eseguita a un'ora specifica, ad esempio alle 8:00. PST il primo lunedì di ogni mese, quindi inserisci un'espressione cron. Per ulteriori informazioni, consulta Cron Expressions nella Amazon EventBridge User Guide.
    - Scegli una pianificazione che venga eseguita a una frequenza regolare, ad esempio ogni 10 minuti. e quindi inserisci un'espressione di frequenza.

12. Scegli Next (Successivo).
13. Per Target types (Tipi di target), scegli Servizio AWS.
14. Per Seleziona una destinazione, scegli Batch job queue. Quindi, configura quanto segue:

    - Job queue (Coda di processo): inserisci il nome della risorsa Amazon (ARN) della coda di processo in cui pianificare il processo.
    - Job definition: (Definizione processo:) Inserisci il nome e la revisione o l'ARN completo della definizione del processo da utilizzare per il processo.
    - Job name: (Nome processo:) Inserisci un nome per il processo.
    - Array size: (Dimensione array:) (Facoltativo) Inserisci una dimensione di array per il processo per eseguire più di una copia. Per ulteriori informazioni, consulta Lavori di matrice.

- Job attempts: (Tentativi dei processi:) (Facoltativo) Inserisci il numero di tentativi del processo in caso di esito negativo. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

15. Per i tipi di destinazione della coda di processi Batch, è EventBridge necessaria l'autorizzazione per inviare eventi alla destinazione. EventBridge può creare il ruolo IAM necessario per l'esecuzione della regola. Esegui una di queste operazioni:

    - Per creare un ruolo IAM automaticamente, seleziona Crea un nuovo ruolo per questa risorsa specifica.
    - Per utilizzare un ruolo IAM che hai già creato, scegli Usa il ruolo esistente.

16. (Facoltativo) Espandere Additional settings (Impostazioni aggiuntive).

    a. Per Configure target input, scegli come elaborare il testo di un evento prima che venga passato alla destinazione.

    b. Per Età massima dell'evento, specificate l'intervallo di tempo per cui vengono conservati gli eventi non elaborati.

    c. Per Riprovare, inserite il numero di volte in cui un evento viene ripetuto.

    d. Per la coda Dead-letter, scegliete un'opzione per la gestione degli eventi non elaborati. Se necessario, specifica la coda Amazon SQS da utilizzare come coda di lettere non scritte.

17. (Facoltativo) Scegli Aggiungi destinazione per aggiungere un'altra destinazione per questa regola.
18. Scegli Next (Successivo).
19. (Facoltativo) Per i tag, scegli Aggiungi nuovo tag per aggiungere un'etichetta di risorsa per la regola. Per ulteriori informazioni, consulta Amazon EventBridge tags.
20. Scegli Next (Successivo).
21. Per Revisione e creazione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Al termine, scegliere Create rule (Crea regola).

Per ulteriori informazioni sulla creazione di regole, consulta Creazione di una EventBridge regola Amazon che viene eseguita secondo una pianificazione nella Amazon EventBridge User Guide.

Tutorial: crea una regola con uno schema di eventi

La procedura seguente illustra come creare una regola con uno schema di eventi.

Per creare una regola che invii l'evento a un obiettivo quando l'evento corrisponde a uno schema definito

> **Note**
>
> Questa procedura è valida per tutti i AWS Batch job di Amazon ECS, Amazon EKS e AWS Fargate.

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel pannello di navigazione, scegli Regole.
4. Scegli Crea regola.
5. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 64 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   > **Note**
   >
   > Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.

6. (Facoltativo) In Descrizione, inserisci una descrizione per la regola.
7. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un Servizio AWS utente del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.
8. (Facoltativo) Disattiva la regola sul bus selezionato se non desideri eseguirla immediatamente.
9. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
10. Scegli Next (Successivo).
11. Per Event Source, scegli AWS l'evento o gli eventi EventBridge partner.
12. (Facoltativo) Per un evento di esempio:

    a. Per Tipo di evento di esempio, scegli AWS eventi.

b. Per gli eventi Sample, scegliete Batch Job State Change.

13. In Metodo di creazione scegli Utilizza modulo del modello.
14. Per Event Pattern:

    a. In Event source (Origine eventi), selezionare Servizi AWS.

    b. Per Servizio AWS, scegli Batch.

    c. Per Tipo di evento, scegliete Batch Job State Change.

15. Scegli Next (Successivo).
16. Per Target types (Tipi di target), scegli Servizio AWS.
17. Per Seleziona un obiettivo, scegli un tipo di obiettivo. Ad esempio, scegli Batch job queue. Quindi specificate quanto segue:

    - Job queue (Coda di processo): inserisci il nome della risorsa Amazon (ARN) della coda di processo in cui pianificare il processo.
    - Job definition: (Definizione processo:) Inserisci il nome e la revisione o l'ARN completo della definizione del processo da utilizzare per il processo.
    - Job name: (Nome processo:) Inserisci un nome per il processo.
    - Array size: (Dimensione array:) (Facoltativo) Inserisci una dimensione di array per il processo per eseguire più di una copia. Per ulteriori informazioni, consulta Lavori di matrice.
    - Job attempts: (Tentativi dei processi:) (Facoltativo) Inserisci il numero di tentativi del processo in caso di esito negativo. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

18. Per i tipi di destinazione della coda di processi Batch, è EventBridge necessaria l'autorizzazione per inviare eventi alla destinazione. EventBridge può creare il ruolo IAM necessario per l'esecuzione della regola. Esegui una di queste operazioni:

    - Per creare un ruolo IAM automaticamente, seleziona Create a new role for this specific resource (Crea un nuovo ruolo per questa risorsa specifica).
    - Per utilizzare un ruolo IAM creato in precedenza, seleziona Use existing role (Utilizza un ruolo esistente).

19. (Facoltativo) Espandere Additional settings (Impostazioni aggiuntive).

    a. Per Configure target input, scegli come viene elaborato il testo di un evento.

    b. Per Età massima dell'evento, specificate l'intervallo di tempo per cui vengono conservati gli eventi non elaborati.

c. Per Riprovare, inserite il numero di volte in cui un evento viene ripetuto.

d. Per la coda Dead-letter, scegliete un'opzione per la gestione degli eventi non elaborati. Se necessario, specifica la coda Amazon SQS da utilizzare come coda di lettere non scritte.

20. (Facoltativo) Scegli Aggiungi un altro obiettivo per aggiungere un altro obiettivo.
21. Scegli Next (Successivo).
22. (Facoltativo) Per Tag, scegli Aggiungi nuovo tag per aggiungere un'etichetta di risorsa. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
23. Scegli Next (Successivo).
24. Per Revisione e creazione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Dopo aver finito, scegli Crea regola.

    Per ulteriori informazioni sulla creazione di regole, consulta la sezione Creazione di EventBridge regole Amazon che reagiscono agli eventi nella Amazon EventBridge User Guide.

Tutorial: passa le informazioni sugli eventi a un AWS Batch target in base a una pianificazione utilizzando il trasformatore EventBridge di ingresso

È possibile utilizzare il trasformatore EventBridge di input per trasmettere informazioni sugli eventi durante l'invio AWS Batch di un lavoro. Ciò può essere particolarmente utile se si richiamano lavori a seguito di altre AWS informazioni sull'evento. Un esempio è il caricamento di un oggetto su un bucket Amazon S3. Puoi anche utilizzare una definizione di processo con valori di sostituzione dei parametri nel comando del contenitore. Il trasformatore EventBridge di input può fornire i valori dei parametri in base ai dati dell'evento.

Quindi, in seguito, si crea un target di AWS Batch evento che analizza le informazioni dall'evento che lo avvia e le trasforma in un oggetto. `parameters` Quando il processo viene eseguito, i parametri dell'evento trigger vengono passati al comando del contenitore del lavoro.

> **Note**
>
> In questo scenario, tutte le AWS risorse (ad esempio bucket, EventBridge regole e CloudTrail log di Amazon S3) devono trovarsi nella stessa regione.

Per creare un AWS Batch target che utilizzi il trasformatore di input

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel pannello di navigazione, scegli Regole.
4. Scegli Crea regola.
5. Per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 64 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   > **Note**
   >
   > Una regola non può avere lo stesso nome di un'altra regola nello stesso Regione AWS bus di eventi sullo stesso bus di eventi.

6. (Facoltativo) In Descrizione, inserisci una descrizione per la regola.
7. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se vuoi che questa regola corrisponda agli eventi provenienti dal tuo account, seleziona Predefinito. Quando un Servizio AWS utente del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.
8. (Facoltativo) Disattiva la regola sul bus selezionato se non desideri eseguirla immediatamente.
9. Per Rule type (Tipo di regola), scegli Schedule (Pianifica).
10. Scegli Continua per creare la regola o Avanti.
11. Per Schedule pattern (Modello di pianificazione), esegui una delle seguenti operazioni:

    - Scegli Una pianificazione dettagliata che viene eseguita a un'ora specifica, ad esempio alle 8:00. PST il primo lunedì di ogni mese, quindi inserisci un'espressione cron. Per ulteriori informazioni, consulta Cron Expressions nella Amazon EventBridge User Guide.
    - Scegli una pianificazione che venga eseguita a una frequenza regolare, ad esempio ogni 10 minuti. e quindi inserisci un'espressione di frequenza.

12. Scegli Next (Successivo).
13. Per Target types (Tipi di target), scegli Servizio AWS.
14. Per Seleziona una destinazione, scegli Batch job queue. Quindi, configura quanto segue:

- Job queue (Coda di processo): inserisci il nome della risorsa Amazon (ARN) della coda di processo in cui pianificare il processo.
- Job definition: (Definizione processo:) Inserisci il nome e la revisione o l'ARN completo della definizione del processo da utilizzare per il processo.
- Job name: (Nome processo:) Inserisci un nome per il processo.
- Array size: (Dimensione array:) (Facoltativo) Inserisci una dimensione di array per il processo per eseguire più di una copia. Per ulteriori informazioni, consulta Lavori di matrice.
- Job attempts: (Tentativi dei processi:) (Facoltativo) Inserisci il numero di tentativi del processo in caso di esito negativo. Per ulteriori informazioni, consulta Ritentativi di lavoro automatizzati.

15. Per i tipi di destinazione della coda di processi Batch, è EventBridge necessaria l'autorizzazione per inviare eventi alla destinazione. EventBridge può creare il ruolo IAM necessario per l'esecuzione della regola. Esegui una di queste operazioni:

    - Per creare un ruolo IAM automaticamente, seleziona Crea un nuovo ruolo per questa risorsa specifica.
    - Per utilizzare un ruolo IAM che hai già creato, scegli Usa il ruolo esistente.

16. (Facoltativo) Espandere Additional settings (Impostazioni aggiuntive).
17. Nella sezione Additional settings (Impostazioni aggiuntive), per Configure target input (Configura input di destinazione, scegli Input Transformer (Trasformatore di input).
18. Seleziona Configure input transformer (Configura trasformatore di input).
19. (Facoltativo) Per l'evento Sample:

    a. Per Tipo di evento di esempio, scegli AWS eventi.

    b. Per gli eventi Sample, scegliete Batch Job State Change.

20. Nella sezione Target input transformer (Trasformatore di input di destinazione), per Input path (Percorso di input), specifica i valori da analizzare dell'evento di attivazione. Ad esempio, per analizzare l'evento Batch Job State Change, utilizzare il seguente formato JSON.

```
{
    "instance": "$.detail.jobId",
    "state": "$.detail.status"
}
```

21. Per Template, immettete quanto segue.

```
{
    "instance": <jobId> ,
    "status": <status>
}
```

22. Scegli Conferma.
23. Per Età massima dell'evento, specificate l'intervallo di tempo per cui vengono conservati gli eventi non elaborati.
24. Per Riprovare, inserite il numero di volte in cui un evento viene ripetuto.
25. Per la coda Dead-letter, scegliete un'opzione per la gestione degli eventi non elaborati. Se necessario, specifica la coda Amazon SQS da utilizzare come coda di lettere non scritte.
26. (Facoltativo) Scegli Aggiungi un altro obiettivo per aggiungere un altro obiettivo.
27. Scegli Next (Successivo).
28. (Facoltativo) Per Tag, scegli Aggiungi nuovo tag per aggiungere un'etichetta di risorsa. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
29. Scegli Next (Successivo).
30. Per Revisione e creazione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Al termine, scegli Crea regola.

Tutorial: ascolta gli eventi AWS Batch di lavoro utilizzando EventBridge

In questo tutorial, configuri una semplice AWS Lambda funzione che ascolta gli eventi di AWS Batch lavoro e li scrive in un flusso di log di CloudWatch Logs.

Prerequisiti

Questo tutorial presuppone che tu disponga di un ambiente di calcolo funzionante e di una coda di processo pronti per accettare processi. Se non disponi di un ambiente di elaborazione in esecuzione e di una coda di lavoro da cui acquisire gli eventi, segui i passaggi indicati per crearne uno. Guida introduttiva ai AWS Batch tutorial Alla fine di questo tutorial, puoi facoltativamente inviare un lavoro a questa coda di lavori per verificare di aver configurato correttamente la tua funzione Lambda.

Argomenti

- Tutorial: Creare la funzione Lambda
- Tutorial: registra la regola dell'evento
- Tutorial: verifica la tua configurazione

Tutorial: Creare la funzione Lambda

In questa procedura, crei una semplice funzione Lambda che funga da destinazione per i messaggi del flusso di AWS Batch eventi.

Per creare una funzione Lambda di destinazione

1. Apri la AWS Lambda console all'indirizzo https://console.aws.amazon.com/lambda/.
2. Scegli Create function (Crea funzione) e Author from scratch (Crea da zero).
3. Nel campo Function name (Nome funzione), immettere batch-event-stream-handler.
4. In Runtime, scegliere Python 3.8.
5. Scegli Crea funzione.
6. Nella sezione Codice sorgente, modifica il codice di esempio in modo che corrisponda al seguente esempio:

```
import json


def lambda_handler(event, _context):
    # _context is not used
    del _context
    if event["source"] != "aws.batch":
        raise ValueError("Function only supports input from events with a source
 type of: aws.batch")

    print(json.dumps(event))
```

   Questa è una semplice funzione Python 3.8 che stampa gli eventi inviati da. AWS Batch Se tutto è configurato correttamente, alla fine di questo tutorial, i dettagli dell'evento vengono visualizzati nel flusso di log CloudWatch Logs associato a questa funzione Lambda.
7. Seleziona Deploy (Implementa).

Tutorial: registra la regola dell'evento

In questa sezione, crei una regola di EventBridge evento che registra gli eventi di lavoro provenienti dalle tue AWS Batch risorse. Questa regola registra tutti gli eventi provenienti AWS Batch dall'account in cui è definita. I messaggi di lavoro stessi contengono informazioni sull'origine dell'evento, inclusa la coda dei lavori in cui è stato inviato. È possibile utilizzare queste informazioni per filtrare e ordinare gli eventi a livello di programmazione.

> **Note**
>
> Se utilizzi il Console di gestione AWS per creare una regola di evento, la console aggiunge automaticamente le autorizzazioni IAM per EventBridge chiamare la tua funzione Lambda. Tuttavia, se stai creando una regola di evento utilizzando il AWS CLI, devi concedere le autorizzazioni in modo esplicito. Per ulteriori informazioni, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

Per creare la tua EventBridge regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se la regola deve cercare eventi corrispondenti provenienti dal tuo account, seleziona Bus di eventi predefiniti di AWS . Quando un AWS servizio del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Next (Successivo).
8. In Event source (Origine eventi), scegli Other (Altro).
9. Per Event pattern, seleziona Modelli personalizzati (editor JSON).

10. Incolla il modello di eventi seguente nell'area di testo.

```
{
  "source": [
    "aws.batch"
  ]
}
```

    Questa regola si applica a tutti i AWS Batch gruppi e a tutti gli AWS Batch eventi. In alternativa, puoi creare una regola più specifica per filtrare alcuni risultati.

11. Scegli Next (Successivo).
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona un obiettivo, scegli la funzione Lambda e seleziona la tua funzione Lambda.
14. (Facoltativo) Per Additional settings (Impostazioni aggiuntive), procedi come segue:

    a. Per Maximum age of event (Età massima dell'evento), immetti un valore compreso tra un minuto (00:01) e 24 ore (24:00).

    b. Per Tentativi, specifica un numero compreso tra 0 e 185.

    c. Per la coda di lettere non scritte, scegli se utilizzare una coda Amazon SQS standard come coda di lettere non scritte. EventBridge invia gli eventi che corrispondono a questa regola alla coda di lettere non scritte se non vengono consegnati correttamente alla destinazione. Esegui una di queste operazioni:

    - Scegli Nessuna per non utilizzare una coda DLQ.
    - Scegli Seleziona una coda Amazon SQS nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dal menu a discesa.
    - Scegli Seleziona una coda Amazon SQS in un altro AWS account come coda di lettere non scritte, quindi inserisci l'ARN della coda da utilizzare. È necessario allegare una policy basata sulle risorse alla coda che conceda l'autorizzazione a inviarle messaggi. EventBridge Per ulteriori informazioni, consulta Concessione delle autorizzazioni alla coda delle lettere non scritte nella Amazon User Guide. EventBridge

15. Scegli Next (Successivo).
16. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
17. Scegli Next (Successivo).
18. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Tutorial: verifica la tua configurazione

Ora puoi testare la tua EventBridge configurazione inviando un lavoro alla tua coda di lavoro. Se tutto è configurato correttamente, la funzione Lambda viene attivata e scrive i dati dell'evento in un flusso di log di CloudWatch Logs per la funzione.

Per testare la configurazione

1. Apri la console all' AWS Batch indirizzo. https://console.aws.amazon.com/batch/
2. Invia un nuovo AWS Batch lavoro. Per ulteriori informazioni, consulta Tutorial: invia un lavoro.
3. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
4. Nel pannello di navigazione, scegli Logs e seleziona il gruppo di log per la tua funzione Lambda (ad esempio, *my-function* /aws/lambda/).
5. Seleziona un flusso di log per visualizzare i dati di evento.

Tutorial: invio di avvisi di Amazon Simple Notification Service per eventi di lavoro non riusciti

In questo tutorial, configuri una regola di EventBridge evento Amazon che acquisisce solo gli eventi di lavoro in cui il lavoro è passato a uno `FAILED` stato. Alla fine di questo tutorial, puoi facoltativamente anche inviare un lavoro a questa coda di lavori. Questo serve per verificare che gli avvisi di Amazon SNS siano stati configurati correttamente.

Prerequisiti

Questo tutorial presuppone che tu disponga di un ambiente di calcolo funzionante e di una coda di processo pronti per accettare processi. Se non disponi di un ambiente di elaborazione in esecuzione e di una coda di lavoro da cui acquisire gli eventi, segui i passaggi indicati per crearne uno. Guida introduttiva ai AWS Batch tutorial

Argomenti

- Tutorial: creare e sottoscrivere un argomento di Amazon SNS
- Tutorial: Registra una regola di evento
- Tutorial: verifica la tua regola
- Regola alternativa: coda di processi Batch bloccata

Tutorial: creare e sottoscrivere un argomento di Amazon SNS

In questo tutorial, configuri un argomento Amazon SNS che funga da destinazione evento per la nuova regola di evento.

Come creare un argomento Amazon SNS

1. Apri la console Amazon SNS nella versione v3/home. https://console.aws.amazon.com/sns/
2. Scegliere Topics (Argomenti), Create topic (Crea argomento).
3. Per Tipo, scegliere Standard.
4. Per Nome, inserisci **`JobFailedAlert`** e scegli Crea argomento.
5. Sullo JobFailedAlertschermo, scegli Crea abbonamento.
6. Per Protocollo, scegli E-mail.
7. In Endpoint, inserire un indirizzo e-mail a cui si ha accesso correntemente, quindi scegliere Crea sottoscrizione.
8. Controlla l'account e-mail e attendi di ricevere una e-mail di conferma della sottoscrizione. Una volta ricevuta, seleziona Confirm subscription (Conferma sottoscrizione).

Tutorial: Registra una regola di evento

Quindi, registra una regola di evento che acquisisca solo gli eventi di processi non riusciti.

Per registrare la tua EventBridge regola

1. Apri la EventBridge console Amazon all'indirizzo https://console.aws.amazon.com/events/.
2. Nel pannello di navigazione, scegli Regole.
3. Scegli Create rule (Crea regola).
4. Inserire un nome e una descrizione per la regola.

   Una regola non può avere lo stesso nome di un'altra regola nella stessa regione e sullo stesso router di eventi.
5. Per Select event bus (Seleziona bus di eventi), scegli il bus di eventi che desideri associare a questa regola. Se la regola deve cercare eventi corrispondenti provenienti dal tuo account, seleziona Bus di eventi predefiniti di AWS . Quando un AWS servizio del tuo account emette un evento, questo passa sempre al bus eventi predefinito del tuo account.

6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Scegli Next (Successivo).
8. In Event source (Origine eventi), scegli Other (Altro).
9. Per Event pattern, seleziona Modelli personalizzati (editor JSON).
10. Incolla il modello di eventi seguente nell'area di testo.

```
{
  "detail-type": [
    "Batch Job State Change"
  ],
  "source": [
    "aws.batch"
  ],
  "detail": {
    "status": [
      "FAILED"
    ]
  }
}
```

Questo codice definisce una EventBridge regola che corrisponde a qualsiasi evento in cui si trova lo stato del lavoro. `FAILED` Per ulteriori informazioni sui pattern di eventi, consulta Events and Event Patterns nella Amazon EventBridge User Guide.

11. Scegli Next (Successivo).
12. Per Target types (Tipi di destinazione), scegli AWS service (Servizio).
13. Per Seleziona un target, scegli l'argomento SNS e per Argomento, scegli JobFailedAlert.
14. (Facoltativo) Per Additional settings (Impostazioni aggiuntive), procedi come segue:

    a. Per Maximum age of event (Età massima dell'evento), immetti un valore compreso tra un minuto (00:01) e 24 ore (24:00).
    b. Per Tentativi, specifica un numero compreso tra 0 e 185.
    c. Per la coda di lettere non scritte, scegli se utilizzare una coda Amazon SQS standard come coda di lettere non scritte. EventBridge invia gli eventi che corrispondono a questa regola alla coda di lettere non scritte se non vengono consegnati correttamente alla destinazione. Esegui una di queste operazioni:
        - Scegli Nessuna per non utilizzare una coda DLQ.

- Scegli Seleziona una coda Amazon SQS nell' AWS account corrente da utilizzare come coda di lettere non scritte, quindi seleziona la coda da utilizzare dal menu a discesa.
- Scegli Seleziona una coda Amazon SQS in un altro AWS account come coda di lettere non scritte, quindi inserisci l'ARN della coda da utilizzare. È necessario allegare una policy basata sulle risorse alla coda che conceda l'autorizzazione a inviarle messaggi. EventBridge Per ulteriori informazioni, consulta Concessione delle autorizzazioni alla coda delle lettere non scritte nella Amazon User Guide. EventBridge

15. Scegli Next (Successivo).
16. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
17. Scegli Next (Successivo).
18. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Tutorial: verifica la tua regola

Per testare la regola, invia un processo che termini subito dopo l'avvio con un codice di uscita diverso da zero. Se la regola dell'evento è configurata correttamente, dovresti ricevere un messaggio e-mail entro pochi minuti con il testo dell'evento.

Verifica di una regola

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Invia un nuovo AWS Batch lavoro. Per ulteriori informazioni, consulta Tutorial: invia un lavoro. Per il comando del processo, sostituisci questo comando per uscire dal container con un codice di uscita 1.

   ```
   /bin/sh, -c, 'exit 1'
   ```

3. Controlla la tua e-mail per confermare di aver ricevuto un avviso e-mail per la notifica di lavoro non riuscito.

Regola alternativa: coda di processi Batch bloccata

Per creare una regola di evento che controlli il blocco della coda dei processi in batch, ripeti questi tutorial con le seguenti modifiche:

1. In Tutorial: creare e sottoscrivere un argomento di Amazon SNS, usa *BlockedJobQueue* come nome dell'argomento.
2. In Tutorial: Registra una regola di evento, usa il seguente schema nell'editor JSON:

```
{
   "detail-type": [
     "Batch Job Queue Blocked"
   ],
   "source": [
     "aws.batch"
   ]
}
```

Elastic Fabric Adapter

Un Elastic Fabric Adapter (EFA) è un dispositivo di rete per accelerare le applicazioni HPC (High Performance Computing). AWS Batch supporta le applicazioni che utilizzano EFA se vengono soddisfatte le seguenti condizioni.

- Per un elenco dei tipi di istanze che supportano EFAs, consulta Tipi di istanze supportati nella Amazon EC2 User Guide.

  > **Tip**
  >
  > Per visualizzare un elenco di tipi di istanze che supportano EFAs in an Regione AWS, esegui il comando seguente. Quindi, incrocia l'elenco restituito con l'elenco dei tipi di istanze disponibili nella AWS Batch console.
  >
  > ```
  > $ aws ec2 describe-instance-types  --region us-east-1  --filters Name=network-info.efa-supported,Values=true  --query "InstanceTypes[*].[InstanceType]"  --output text | sort
  > ```

- Per un elenco dei sistemi operativi che supportano EFA, consulta Sistemi operativi supportati.
- L'AMI ha il driver EFA caricato.
- Il gruppo di sicurezza per l'EFA deve consentire tutto il traffico in entrata e in uscita da e verso il gruppo di sicurezza stesso.
- Tutte le istanze che utilizzano un EFA devono appartenere allo stesso gruppo di collocamento del cluster.
- La definizione di processo deve includere un membro `devices` con `hostPath` impostato su `/dev/infiniband/uverbs0` per consentire al dispositivo EFA essere passato al container. Se `containerPath` specificato, deve essere impostato anche su. `/dev/infiniband/uverbs0` Se `permissions` è specificato, deve essere impostato su `READ` | `WRITE` | `MKNOD`.

  La posizione dei LinuxParametersmembri è diversa per i job paralleli multinodo e per i job container a nodo singolo. Gli esempi seguenti mostrano le differenze, ma mancano i valori obbligatori.

  Example Esempio per il processo parallelo a più nodi

  ```
  {
    "jobDefinitionName": "EFA-MNP-JobDef",
  ```

```
  "type": "multinode",
  "nodeProperties": {
    ...
    "nodeRangeProperties": [
      {
        ...
        "container": {
          ...
          "linuxParameters": {
            "devices": [
              {
                "hostPath": "/dev/infiniband/uverbs0",
                "containerPath": "/dev/infiniband/uverbs0",
                "permissions": [
                    "READ", "WRITE", "MKNOD"
                ]
              },
            ],
          },
        },
      },
    ],
  },
}
```

Example Esempio per il processo container a nodo singolo

```
{
  "jobDefinitionName": "EFA-Container-JobDef",
  "type": "container",
  ...
  "containerProperties": {
    ...
    "linuxParameters": {
      "devices": [
        {
          "hostPath": "/dev/infiniband/uverbs0",
        },
      ],
    },
  },
}
```

Per ulteriori informazioni su EFA, consulta Elastic Fabric Adapter nella Guida per EC2 l'utente di Amazon.

Monitor AWS Batch

Il monitoraggio è un elemento importante per mantenere l'affidabilità, la disponibilità e le prestazioni della AWS Batch AWS soluzione.

Consigliamo vivamente di raccogliere dati di monitoraggio da tutte le parti della AWS soluzione per semplificare il debug di un errore multipunto, se si verifica. Iniziare creando un piano di monitoraggio che risponda alle seguenti domande. Se non sei sicuro di come rispondere a queste domande, puoi comunque utilizzare Amazon CloudWatch Logs per stabilire le tue linee di base delle prestazioni.

- Quali sono gli obiettivi del monitoraggio?
- Quali risorse verranno monitorate?
- Con quale frequenza eseguirai il monitoraggio di queste risorse?
- Quali strumenti di monitoraggio verranno usati?
- Chi eseguirà i processi di monitoraggio?
- Chi deve ricevere una notifica quando si verifica un problema?

Il passo successivo è stabilire una linea di base delle AWS Batch prestazioni normali nel tuo ambiente misurando le prestazioni in diversi momenti e in diverse condizioni di carico. Durante il monitoraggio AWS Batch, conservate i dati di monitoraggio storici in modo da poterli confrontare con i dati sulle prestazioni correnti. Questo permette di identificare i normali modelli di prestazioni e le anomalie di prestazioni e definire metodi per risolvere i problemi.

Gli argomenti di questa sezione consentono di avviare la registrazione e il monitoraggio di AWS Batch.

Argomenti

Utilizzo dei CloudWatch registri con AWS Batch

Puoi configurare i tuoi AWS Batch lavori sulle EC2 risorse per inviare informazioni e metriche di registro dettagliate a CloudWatch Logs. In questo modo, puoi visualizzare diversi registri dei tuoi

lavori in un'unica comoda posizione. Per ulteriori informazioni sui CloudWatch log, consulta What is Amazon CloudWatch Logs? nella Amazon CloudWatch User Guide.

> **Note**
>
> Per impostazione predefinita, CloudWatch i registri sono attivati per i contenitori AWS Fargate.

Per attivare e personalizzare la registrazione dei CloudWatch log, esaminate le seguenti attività di configurazione una tantum:

- Per gli ambienti di AWS Batch calcolo basati sulle EC2 risorse, aggiungi una policy IAM al ruolo. `ecsInstanceRole` Per ulteriori informazioni, consulta the section called "Tutorial: Aggiungere una politica CloudWatch Logs IAM".
- Crea un modello di EC2 lancio Amazon che includa un CloudWatch monitoraggio dettagliato, quindi specifica il modello quando crei il tuo ambiente di AWS Batch elaborazione. Puoi anche installare l' CloudWatchagente su un'immagine esistente e quindi specificare l'immagine nella procedura guidata di AWS Batch prima esecuzione.
- (Facoltativo) Configura il driver awslogs. È possibile aggiungere parametri che modificano il comportamento predefinito sia sulle risorse di Fargate che su quelle EC2 di Fargate. Per ulteriori informazioni, consulta the section called "Usa il driver di registro awslogs".

Argomenti

- Tutorial: Aggiungere una politica CloudWatch Logs IAM
- Installa e configura l'agente CloudWatch
- Tutorial: Visualizza CloudWatch i registri

Tutorial: Aggiungere una politica CloudWatch Logs IAM

Prima che i job possano inviare dati di log e metriche dettagliate a CloudWatch Logs, devi creare una policy IAM che utilizzi i Logs. CloudWatch APIs Dopo aver creato la policy IAM, collegala al ruolo. `ecsInstanceRole`

> **Note**
>
> Se la `ECS-CloudWatchLogs` policy non è associata al `ecsInstanceRole` ruolo, le metriche di base possono comunque essere inviate a CloudWatch Logs. Tuttavia, le metriche di base non includono dati di registro o metriche dettagliate come lo spazio libero su disco.

AWS Batch gli ambienti di calcolo utilizzano EC2 risorse Amazon. Quando crei un ambiente di calcolo utilizzando la procedura guidata AWS Batch di prima esecuzione, AWS Batch crea il `ecsInstanceRole` ruolo e configura l'ambiente con esso.

Se non utilizzi la procedura guidata per la prima esecuzione, puoi specificare il `ecsInstanceRole` ruolo quando crei un ambiente di calcolo nell'API or. AWS Command Line Interface AWS Batch Per ulteriori informazioni, consulta AWS CLI Command Reference o AWS Batch API Reference.

Co,e creare la policy IAM **`ECS-CloudWatchLogs`**

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel riquadro di navigazione, scegli Policy.
3. Scegli Create Policy (Crea policy).
4. Scegli JSON, quindi inserisci la seguente politica:

   JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogStreams"
            ],
            "Resource": [
                "arn:aws:logs:*:*:*"
            ]
        }
```

```
    ]
}
```

5. Scegli Successivo: Tag.
6. (Facoltativo) Per Aggiungi tag, scegli Aggiungi tag per aggiungere un tag alla politica.
7. Scegli Prossimo: Rivedi.
8. Nella pagina di revisione della politica, per Nome, inserisci**`ECS-CloudWatchLogs`**, quindi inserisci una Descrizione facoltativa.
9. Scegli Crea policy.

Per collegare la policy **`ECS-CloudWatchLogs`** a **`ecsInstanceRole`**

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Nel pannello di navigazione, seleziona Ruoli.
3. Scegli `ecsInstanceRole`. Se il ruolo non esiste, segui le procedure indicate Ruolo dell'istanza Amazon ECS per crearlo.
4. Scegli Aggiungi autorizzazioni, quindi scegli Allega politiche.
5. Scegli la policy ECS-, quindi scegli Allega CloudWatchLogs policy.

Installa e configura l'agente CloudWatch

Puoi creare un modello di EC2 lancio di Amazon che includa CloudWatch il monitoraggio. Per ulteriori informazioni, consulta Launch an instance from a launch template e Advanced details nella Amazon EC2 User Guide.

Puoi anche installare l' CloudWatch agente su un EC2 AMI Amazon esistente e quindi specificare l'immagine nella procedura guidata AWS Batch di prima esecuzione. Per ulteriori informazioni, consulta Installazione dell' CloudWatch agente e. Guida introduttiva ai AWS Batch tutorial

> **Note**
>
> I modelli di avvio non sono supportati nelle AWS Fargate risorse.

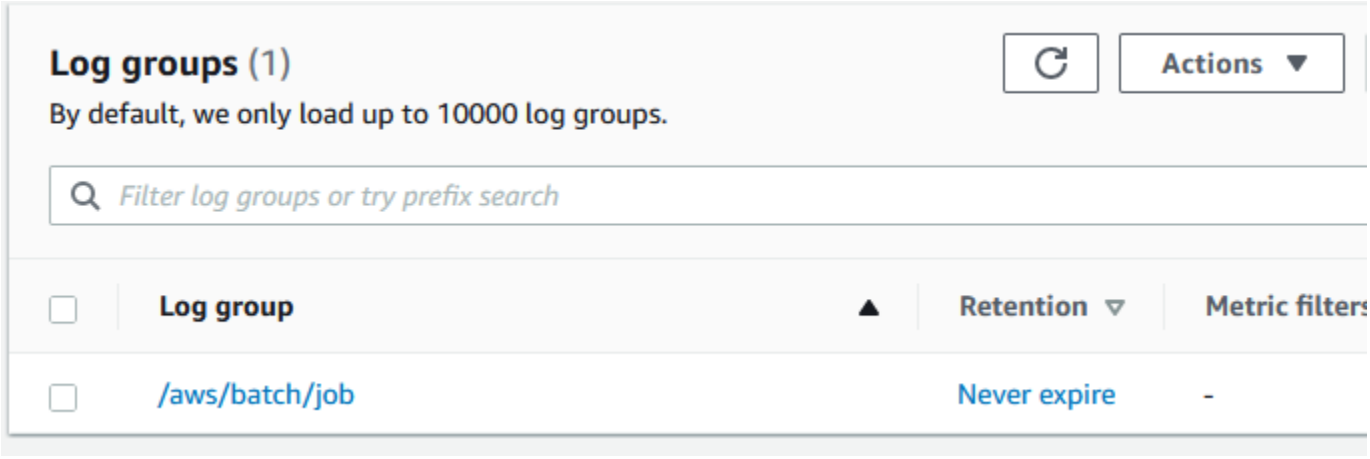
Tutorial: Visualizza CloudWatch i registri

È possibile visualizzare e cercare CloudWatch i registri dei registri in. Console di gestione AWS

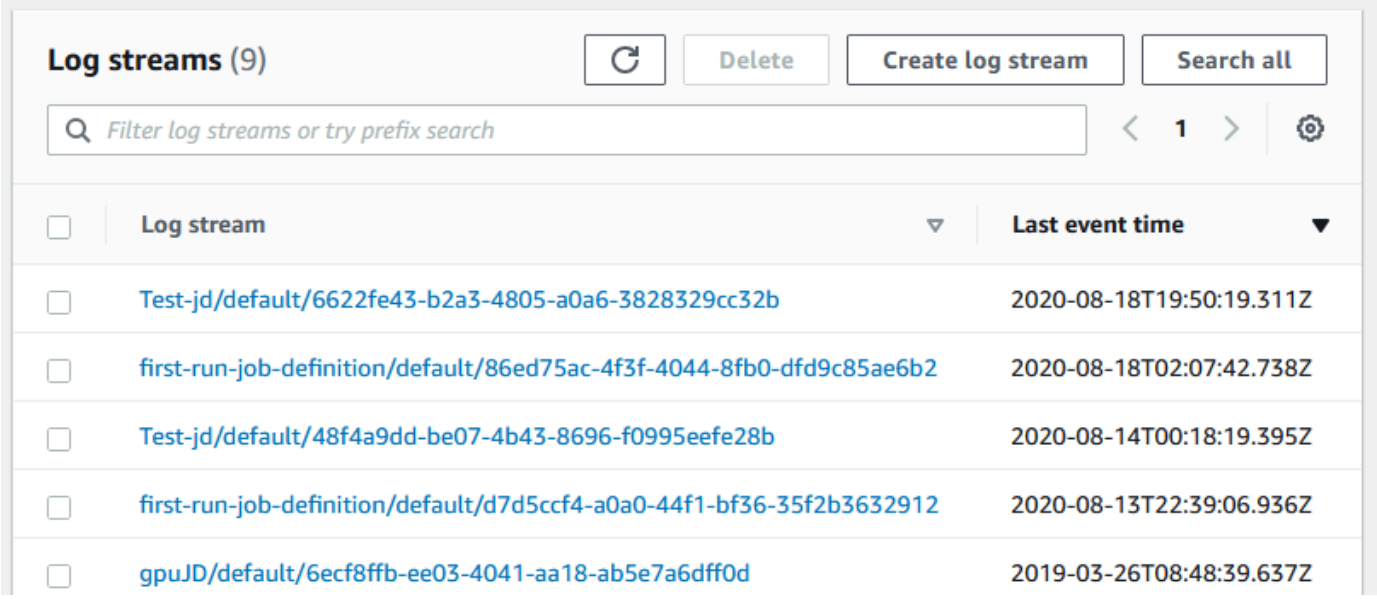
> **Note**
>
> La visualizzazione dei dati nei registri potrebbe richiedere alcuni minuti. CloudWatch

Per visualizzare i dati dei CloudWatch log

1. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
2. Nel riquadro di navigazione a sinistra, scegli Registri, quindi scegli Gruppi di log.

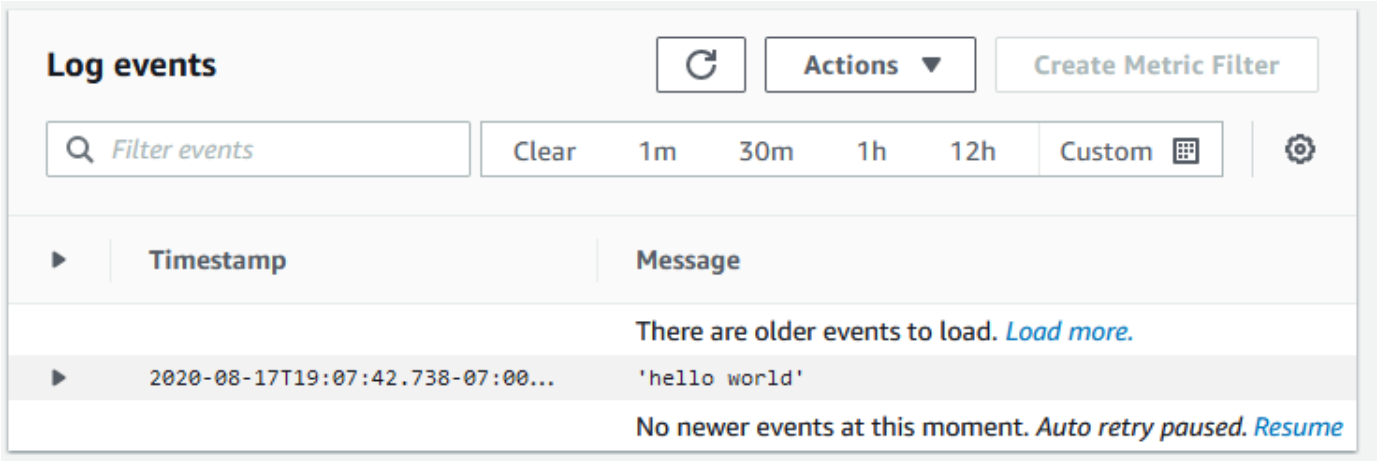


3. Scegli un gruppo di log da visualizzare.



4. Scegli un flusso di log da visualizzare. Per impostazione predefinita, gli stream sono identificati dai primi 200 caratteri del nome del lavoro e dall'ID attività Amazon ECS.

> **Tip**
>
> Per scaricare i dati del flusso di log, scegli Azioni.



AWS Batch CloudWatch Informazioni approfondite sui container

CloudWatch Container Insights raccoglie, aggrega e riepiloga le metriche e i log degli ambienti e dei lavori di elaborazione. AWS Batch Le metriche includono l'utilizzo di CPU, memoria, disco e rete. Puoi aggiungere queste metriche ai dashboard. CloudWatch

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 ambiente di calcolo e a livello di lavoro come metriche. CloudWatch Per ulteriori informazioni, consulta Container Insights Structured Logs for Amazon ECS nella Amazon CloudWatch User Guide.

> **Important**
>
> CloudWatch I costi di Container Insights vengono addebitati come parametri personalizzati da. CloudWatch Per ulteriori informazioni, consulta i prezzi di Amazon CloudWatch Events

Argomenti

- Attiva Container Insights

Attiva Container Insights

Completa i seguenti passaggi per attivare Container Insights per gli ambienti di AWS Batch elaborazione.

1. Apri la AWS Batch console.
2. Scegliere Environments (Ambienti).
3. Scegli l'ambiente di calcolo che desideri.
4. Nella scheda Container insights, attiva Container insights per l'ambiente di calcolo.

> **Tip**
>
> Puoi selezionare un intervallo predefinito per aggregare le metriche o creare un intervallo personalizzato.

Per impostazione predefinita, vengono visualizzate le seguenti metriche. Per un elenco completo dei parametri di Amazon ECS Container Insights, consulta Amazon ECS Container Insights Metrics nella Amazon User Guide. CloudWatch

- **`JobCount`**— Il numero di processi eseguiti nell'ambiente di elaborazione.
- **`ContainerInstanceCount`**— Il numero di istanze Amazon Elastic Compute Cloud che eseguono l'agente Amazon ECS e sono registrate nell'ambiente di calcolo.
- **`MemoryReserved`**— La memoria riservata dai lavori in ambiente di calcolo. Questa metrica viene raccolta solo per i lavori che hanno una prenotazione di memoria definita nella definizione del processo.
- **`MemoryUtilized`**— La memoria utilizzata dai processi in ambiente di calcolo. Questa metrica viene raccolta solo per i lavori che hanno una prenotazione di memoria definita nella definizione del processo.
- **`CpuReserved`**— Le unità CPU riservate dai job dell'ambiente di calcolo. Questa metrica viene raccolta solo per i lavori che hanno una prenotazione CPU definita nella definizione del processo.
- **`CpuUtilized`**— Le unità CPU utilizzate dai processi nell'ambiente di elaborazione. Questa metrica viene raccolta solo per i lavori che hanno una prenotazione CPU definita nella definizione del processo.
- **`NetworkRxBytes`**- Il numero di byte ricevuti. Questa metrica è disponibile solo per i contenitori nei lavori che utilizzano le modalità di rete `awsvpc` o bridge.

- **`NetworkTxBytes`**— Il numero di byte che vengono trasmessi. Questa metrica è disponibile solo per i contenitori nei lavori che utilizzano le modalità di rete `awsvpc` o bridge.
- **`StorageReadBytes`**— Il numero di byte letti dallo storage.
- **`StorageWriteBytes`**— Il numero di byte che vengono scritti nello storage.

Usa CloudWatch Logs per monitorare i lavori AWS Batch su Amazon EKS

Puoi utilizzare Amazon CloudWatch Logs per monitorare, archiviare e visualizzare tutti i tuoi file di registro in un'unica posizione. Utilizzando CloudWatch Logs, puoi cercare, filtrare e analizzare i dati di log da più fonti.

È possibile scaricare un modulo AWS Fluent Bit immagine che include un plug-in per il monitoraggio dei AWS Batch lavori Amazon EKS in CloudWatch Logs. Fluent Bit è un elaboratore di log e forwarder open source che è sia Docker che Kubernetes compatibile. Ti consigliamo di utilizzare Fluent Bit come router di registro perché richiede meno risorse rispetto a Fluentd. Per ulteriori informazioni, consulta Installare l' CloudWatch agente con il componente aggiuntivo Amazon CloudWatch Observability EKS o il grafico Helm.

Prerequisiti

- Allega la `CloudWatchAgentServerPolicy` policy alla AWS Identity and Access Management policy dei tuoi nodi di lavoro. Per ulteriori informazioni, consulta Verificare i prerequisiti.

Installa il componente aggiuntivo

Per istruzioni su come installare AWS per Fluent Bit e crea i CloudWatch gruppi, consulta Installare l' CloudWatch agente con il componente aggiuntivo Amazon CloudWatch Observability EKS o il grafico Helm.

È necessario fornire i seguenti dati di configurazione aggiuntivi durante l'installazione del componente aggiuntivo:

- Se si installa il componente aggiuntivo con il, è Console di gestione AWS necessario fornire le seguenti tolleranze nei valori di configurazione:

```
{
```

```
  "tolerations": [
    {
      "key": "batch.amazonaws.com/batch-node",
      "operator": "Exists"
    }
  ]
}
```

- Se installi il componente aggiuntivo con, AWS CLI aggiungi i seguenti argomenti:

```
--configuration-values '{"tolerations":[{"key":"batch.amazonaws.com/batch-
node","operator":"Exists"}]}'
```

> **Tip**
>
> Ricordate che Fluent Bit utilizza 0,5 CPU e 100 MB di memoria sui AWS Batch nodi. Ciò riduce la capacità totale disponibile per i AWS Batch lavori. Consideratelo quando valutate le vostre opportunità di lavoro.

Tagga le tue AWS Batch risorse

Per aiutarti a gestire AWS Batch le tue risorse, puoi assegnare i tuoi metadati a ciascuna risorsa sotto forma di tag. Questo argomento descrive i tag e mostra come crearli.

Argomenti

Nozioni di base sui tag

Un tag è un'etichetta che si assegna a una AWS risorsa. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili.

I tag consentono di classificare le AWS risorse in base, ad esempio, allo scopo, al proprietario o all'ambiente. Se disponi di un numero elevato di risorse, puoi individuare rapidamente una risorsa specifica in base ai tag assegnati. Ad esempio, puoi definire un set di tag per AWS Batch i tuoi servizi per aiutarti a tenere traccia del proprietario e del livello di stack di ciascun servizio. Consigliamo di definire un set coerente di chiavi di tag per ciascun tipo di risorsa.

I tag non vengono assegnati in automatico alle risorse. Dopo aver aggiunto un tag, puoi modificarne le chiavi e i valori oppure rimuovere i tag da una risorsa in qualsiasi momento. Se elimini una risorsa, verranno eliminati anche tutti i tag a essa associati.

I tag non hanno alcun significato semantico AWS Batch e vengono interpretati rigorosamente come una stringa di caratteri. Puoi impostare il valore di un tag su una stringa vuota, ma non su null. Se aggiungi un tag con la stessa chiave di un tag esistente a una risorsa specifica, il nuovo valore sovrascrive quello precedente.

Puoi lavorare con i tag utilizzando l' Console di gestione AWS AWS CLI, la e l' AWS Batch API.

Se utilizzi AWS Identity and Access Management (IAM), puoi controllare quali utenti del tuo AWS account sono autorizzati a creare, modificare o eliminare i tag.

Assegnazione di tag alle risorse

È possibile contrassegnare ambienti di AWS Batch elaborazione, processi, definizioni di lavoro, code di lavoro e politiche di pianificazione nuovi o esistenti.

Se utilizzi la AWS Batch console, puoi applicare i tag alle nuove risorse al momento della creazione o alle risorse esistenti in qualsiasi momento utilizzando la scheda Tag nella pagina delle risorse pertinente.

Se utilizzi l' AWS Batch API, l' AWS CLI o un AWS SDK, puoi applicare i tag a nuove risorse utilizzando il `tags` parametro sull'azione API pertinente o alle risorse esistenti utilizzando l'azione `TagResource` API. Per ulteriori informazioni, consulta TagResource.

Alcune operazioni per la creazione di risorse ti consentono di specificare tag per una risorsa durante la sua creazione. Se i tag non possono essere applicati durante la creazione della risorsa, il processo di creazione della risorsa avrà esito negativo. In questo modo, le risorse a cui desideri applicare tag al momento della creazione vengono create con tag specifici o non vengono create affatto. Se aggiungi tag alle risorse al momento della creazione, non devi eseguire script di tagging personalizzati dopo la creazione delle risorse.

La tabella seguente descrive le AWS Batch risorse che possono essere taggate e le risorse che possono essere taggate al momento della creazione.

Supporto dei tag per AWS Batch le risorse

Risorsa	Supporta tag	Supporta la propagazione di tag	Supporta l'etichettatura alla creazione (AWS Batch API AWS CLI, AWS SDK)
AWS Batch ambienti di calcolo	Sì	No. I tag dell'ambiente di calcolo non si propagano a nessun'altra risorsa. I tag per le risorse sono specificati nei tag membri dell'oggetto ComputeResources passato	Sì

Risorsa	Supporta tag	Supporta la propagazione di tag	Supporta l'etichettatura alla creazione (AWS Batch API AWS CLI, AWS SDK)
		nell'operazione API. CreateComputeEnvironment	
AWS Batch lavori	Sì	Sì	Sì
AWS Batch definizioni di lavoro	Sì	No	Sì
AWS Batch code di lavoro	Sì	No	Sì
AWS Batch politiche di pianificazione	Sì	No	Sì

Limitazioni applicate ai tag

Si applicano le seguenti limitazioni di base ai tag:

- Numero massimo di tag per risorsa: 50
- Per ciascuna risorsa, ogni chiave del tag deve essere univoca e ogni chiave del tag può avere un solo valore.
- Lunghezza massima della chiave: 128 caratteri Unicode in formato UTF-8
- Lunghezza massima del valore: 256 caratteri Unicode in formato UTF-8
- Se il tuo schema di tagging viene utilizzato su più AWS servizi e risorse, ricorda che altri servizi potrebbero avere restrizioni sui caratteri consentiti. I caratteri generalmente consentiti sono: lettere, numeri, spazi rappresentabili in formato UTF-8 e i seguenti caratteri speciali: + - = . _ : / @.
- I valori e le chiavi dei tag rispettano la distinzione tra maiuscole e minuscole.
- Non utilizzare alcuna combinazione maiuscola o minuscola `aws:AWS:`, ad esempio un prefisso per chiavi o valori, poiché è riservata all'uso. AWS Non è possibile modificare né eliminare le chiavi o

i valori di tag con tale prefisso. I tag con questo prefisso non vengono conteggiati ai fini del limite.
tags-per-resource

Tutorial: gestire i tag utilizzando la console

Utilizzando la AWS Batch console, puoi gestire i tag associati ad ambienti di calcolo, processi, definizioni di lavoro e code di lavoro nuovi o esistenti.

Aggiungi tag a una singola risorsa al momento della creazione

Puoi aggiungere tag agli ambienti di AWS Batch calcolo, ai lavori, alle definizioni dei processi, alle code di lavoro e alle politiche di pianificazione al momento della creazione.

Aggiunta ed eliminazione di tag in una singola risorsa

AWS Batch consente di aggiungere o eliminare tag associati ai cluster direttamente dalla pagina della risorsa.

Per aggiungere o eliminare un tag su una singola risorsa

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, scegli la regione da usare.
3. Nel riquadro di navigazione, scegli un tipo di risorsa (ad esempio Job Queues).
4. Scegli una risorsa specifica, quindi scegli Modifica tag.
5. Aggiungi o elimina i tag secondo necessità.
   - Per aggiungere un tag, specifica la chiave e il valore nelle caselle di testo vuote alla fine dell'elenco.
   - Per eliminare un tag, scegli il

     Delete icon
     pulsante accanto al tag.
6. Ripeti questa procedura per ogni tag che desideri aggiungere o eliminare, quindi scegli Modifica tag per terminare.

Gestisci i tag utilizzando la CLI o l'API

Utilizza i seguenti AWS CLI comandi o operazioni AWS Batch API per aggiungere, aggiornare, elencare ed eliminare i tag delle tue risorse.

Supporto dei tag per AWS Batch le risorse

Attività	Azione API	AWS CLI	AWS Tools for Windows PowerShell
Aggiungere sovrascrivere uno o più tag.	TagResource	tag-resource	Aggiungi BATResource tag
Eliminare uno o più tag.	UntagResource	untag-resource	Rimuovi- BATResource Tag
Elencazione dei tag associati a una risorsa	ListTagsForResource	list-tags-for-resource	Ottieni- BATResource Tag

I seguenti esempi mostrano come aggiungere o rimuovere tag alle o dalle risorse utilizzando la AWS CLI.

Esempio 1: tag a una risorsa esistente

Il comando seguente applica un tag a una risorsa esistente.

```
aws batch tag-resource --resource-arn resource_ARN --tags team=devs
```

Esempio 2: rimozione di un tag da una risorsa esistente

Il comando seguente elimina un tag da una risorsa esistente.

```
aws batch untag-resource --resource-arn resource_ARN --tag-keys tag_key
```

Esempio 3: elencazione dei tag di una risorsa

Il comando seguente elenca i tag associati a una risorsa esistente.

```
aws batch list-tags-for-resource --resource-arn resource_ARN
```

Alcune operazioni per la creazione di risorse ti consentono di specificare tag quando crei le risorse. Le seguenti operazioni supportano il tagging in fase di creazione.

Attività	Azione API	AWS CLI	AWS Tools for Windows PowerShell
Crea un ambiente di calcolo	CreateComputeEnvironment	create-compute-environment	Nuovo ambiente BATCompute
Crea una coda di lavoro	CreateJobQueue	create-job-queue	Nuova coda BATJob
Crea una politica di pianificazione	CreateSchedulingPolicy	create-scheduling-policy	Nuova politica BATScheduling
Registra una definizione di lavoro	RegisterJobDefinition	register-job-definition	Registro - BATJob Definizione
Invio di un processo	SubmitJob	invia lavoro	Invia- BATJob
Crea una risorsa consumabile	CreateConsumableResource	create-consumable-resource	Crea - Risorsa BATConsumable

Le migliori pratiche per AWS Batch

Puoi utilizzarlo AWS Batch per eseguire una varietà di carichi di lavoro computazionali impegnativi su larga scala senza gestire un'architettura complessa. AWS Batch i job possono essere utilizzati in un'ampia gamma di casi d'uso in aree come l'epidemiologia, i giochi e l'apprendimento automatico.

Questo argomento descrive le migliori pratiche da considerare durante l'utilizzo AWS Batch e le indicazioni su come eseguire e ottimizzare i carichi di lavoro durante l'utilizzo. AWS Batch

Argomenti

Quando usare AWS Batch

AWS Batch esegue lavori su larga scala e a basso costo e fornisce servizi di coda e scalabilità ottimizzata in termini di costi. Tuttavia, non tutti i carichi di lavoro sono adatti all'esecuzione. AWS Batch

- Processi brevi: se un processo viene eseguito solo per pochi secondi, il sovraccarico necessario per pianificare il processo batch potrebbe richiedere più tempo dell'esecuzione del lavoro stesso. Come soluzione alternativa, binpack raggruppate le vostre attività prima di inviarle. AWS Batch Quindi, configura i tuoi AWS Batch lavori in modo che ripetano le attività. Ad esempio, inserisci gli argomenti delle singole attività in una tabella Amazon DynamoDB o come file in un bucket Amazon S3. Prendi in considerazione la possibilità di raggruppare le attività in modo che le attività vengano eseguite da 3 a 5 minuti ciascuna. Dopo di te binpack i lavori, passa in rassegna i gruppi di attività all'interno del tuo AWS Batch lavoro.
- I lavori che devono essere eseguiti immediatamente: AWS Batch possono elaborare i lavori rapidamente. Tuttavia, AWS Batch è uno strumento di pianificazione e ottimizza i costi, le

prestazioni, la priorità dei lavori e la produttività. AWS Batch l'elaborazione delle richieste potrebbe richiedere del tempo. Se hai bisogno di una risposta in meno di pochi secondi, è più adatto un approccio basato sui servizi che utilizza Amazon ECS o Amazon EKS.

Lista di controllo da eseguire su larga scala

Prima di eseguire un carico di lavoro di grandi dimensioni su 50.000 o più vCPUs, considera la seguente lista di controllo.

> **Note**
>
> Se prevedi di eseguire un carico di lavoro elevato su un milione o più di v CPUs o hai bisogno di assistenza su larga scala, contatta il tuo team. AWS

- Controlla le tue EC2 quote Amazon: controlla le tue EC2 quote Amazon (note anche come limiti) nel pannello Service Quotas di. Console di gestione AWS Se necessario, richiedi un aumento della quota per il numero massimo di EC2 istanze Amazon. Ricorda che le istanze Amazon EC2 Spot e Amazon On-Demand hanno quote separate. Per ulteriori informazioni, vedere Guida introduttiva a Service Quotas.
- Verifica la tua quota di Amazon Elastic Block Store per ogni regione: ogni istanza utilizza un GP3 volume GP2 or per il sistema operativo. Per impostazione predefinita, la quota per ciascuno Regione AWS è di 300 TiB. Tuttavia, ogni istanza utilizza i conteggi come parte di questa quota. Quindi, assicurati di tenerne conto quando verifichi la quota di Amazon Elastic Block Store per ogni regione. Se la tua quota viene raggiunta, non puoi creare altre istanze. Per ulteriori informazioni, consulta Endpoint e quote di Amazon Elastic Block Store
- Usa Amazon S3 per lo storage: Amazon S3 offre un throughput elevato e aiuta a eliminare le congetture sulla quantità di storage da fornire in base al numero di processi e istanze in ciascuna zona di disponibilità. Per ulteriori informazioni, consulta Modelli di progettazione basati sulle best practice: ottimizzazione delle prestazioni di Amazon S3.
- Scalabilità graduale per identificare tempestivamente le strozzature: per un lavoro che viene eseguito con un milione o più di VCPUs, inizia con un valore inferiore e aumenta gradualmente, in modo da poter identificare tempestivamente gli ostacoli. Ad esempio, inizia eseguendo 50 mila v. CPUs Quindi, aumenta il conteggio a 200 mila vCPUs, quindi a 500 mila v CPUs e così via. In altre parole, continuate ad aumentare gradualmente il numero di vCPU fino a raggiungere il numero desiderato di v. CPUs

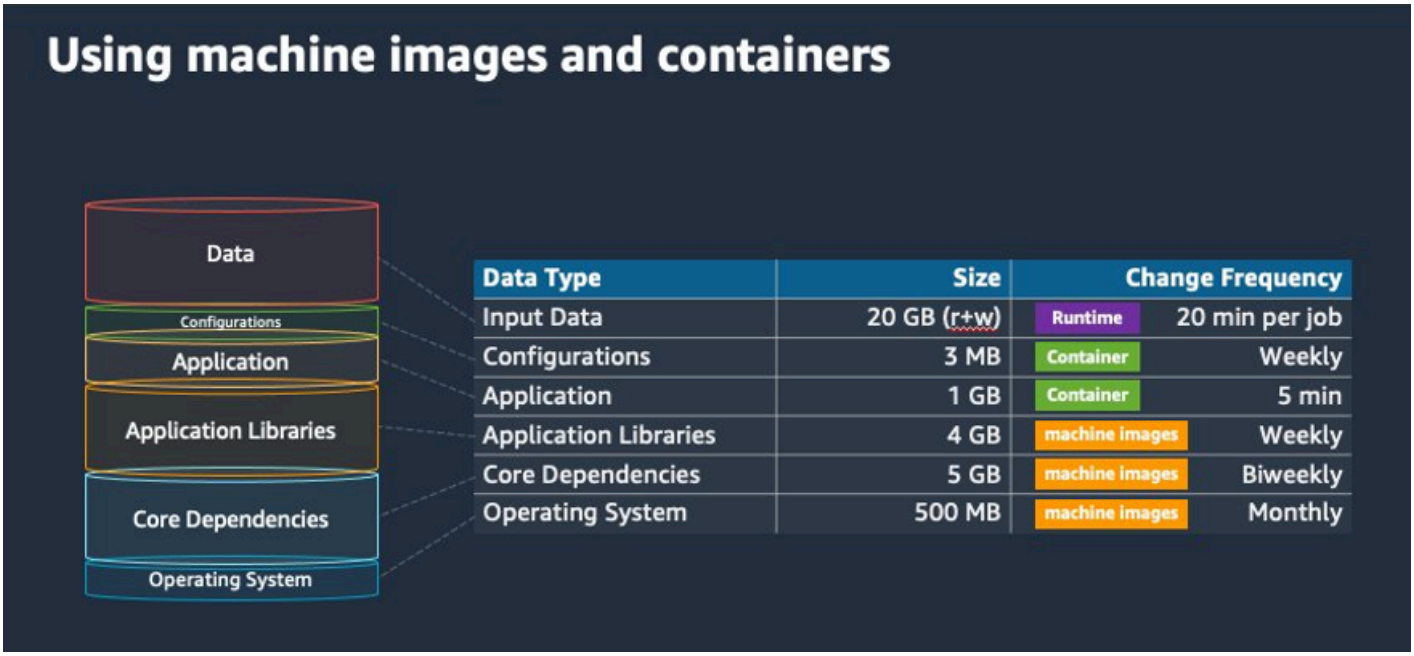
- Monitora per identificare tempestivamente potenziali problemi: per evitare potenziali interruzioni e problemi durante l'esecuzione su larga scala, assicurati di monitorare sia l'applicazione che l'architettura. Potrebbero verificarsi interruzioni anche quando si passa da 1.000 a 5.000 v. CPUs Puoi utilizzare Amazon CloudWatch Logs per esaminare i dati di log o utilizzare CloudWatch Embedded Metrics utilizzando una libreria client. Per ulteriori informazioni, consulta CloudWatch Logs agent reference e `aws-embedded-metrics`

Ottimizza i contenitori e AMIs

Le dimensioni e la struttura dei contenitori sono importanti per la prima serie di lavori eseguiti. Ciò è particolarmente vero se il contenitore è più grande di 4 GB. Le immagini del contenitore sono costruite a strati. I livelli vengono recuperati in parallelo da Docker utilizzando tre thread simultanei. È possibile aumentare il numero di thread simultanei utilizzando il parametro. `max-concurrent-downloads` Per ulteriori informazioni, consulta la documentazione di Dockerd.

Sebbene sia possibile utilizzare contenitori più grandi, si consiglia di ottimizzare la struttura e le dimensioni dei contenitori per tempi di avvio più rapidi.

- I contenitori più piccoli vengono recuperati più rapidamente: i contenitori più piccoli possono portare a tempi di avvio delle applicazioni più rapidi. Per ridurre le dimensioni del contenitore, scarica le librerie o i file che vengono aggiornati di rado all'Amazon Machine Image (AMI). Puoi anche usare bind mount per consentire l'accesso ai tuoi contenitori. Per ulteriori informazioni, consulta Bind mounts.
- Crea livelli di dimensioni uniformi e suddividi strati di grandi dimensioni: ogni livello viene recuperato da un thread. Pertanto, un livello di grandi dimensioni potrebbe influire in modo significativo sui tempi di avvio del lavoro. Consigliamo una dimensione massima del layer di 2 GB come buon compromesso tra dimensioni del contenitore più grandi e tempi di avvio più rapidi. Puoi eseguire il `docker history your_image_id` comando per controllare la struttura dell'immagine del contenitore e le dimensioni del livello. Per ulteriori informazioni, consulta la documentazione di Docker.
- Usa Amazon Elastic Container Registry come repository di container: quando esegui migliaia di lavori in parallelo, un repository autogestito può fallire o limitare il throughput. Amazon ECR funziona su larga scala e può gestire carichi di lavoro fino a oltre un milione di v. CPUs



Scegli la risorsa giusta per l'ambiente di elaborazione

AWS Fargate richiede meno impostazioni e configurazioni iniziali rispetto ad Amazon EC2 ed è probabilmente più facile da usare, soprattutto se è la prima volta. 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.

Se hai i seguenti requisiti, ti consigliamo di utilizzare le istanze Fargate:

- I lavori devono iniziare rapidamente, in particolare in meno di 30 secondi.
- I requisiti per i tuoi processi sono 16 v CPUs o meno GPUs, niente e 120 GiB di memoria o meno.

Per ulteriori informazioni, consulta Quando usare Fargate.

Se hai i seguenti requisiti, ti consigliamo di utilizzare EC2 le istanze Amazon:

- È necessario un maggiore controllo sulla selezione delle istanze o è necessario utilizzare tipi di istanze specifici.
- I tuoi lavori richiedono risorse che non AWS Fargate possono fornire, ad GPUs esempio più memoria, un'AMI personalizzata o Amazon Elastic Fabric Adapter.
- È necessario un elevato livello di velocità effettiva o di concorrenza.

- Devi personalizzare la tua AMI, Amazon EC2 Launch Template o accedere a parametri Linux speciali.

Con Amazon EC2, puoi ottimizzare in modo più preciso il tuo carico di lavoro in base ai tuoi requisiti specifici ed eseguirlo su larga scala, se necessario.

Amazon EC2 On-Demand o Amazon EC2 Spot

La maggior parte dei AWS Batch clienti utilizza le istanze Amazon EC2 Spot grazie ai risparmi rispetto alle istanze On-Demand. Tuttavia, se il tuo carico di lavoro dura più ore e non può essere interrotto, le istanze On-Demand potrebbero essere più adatte a te. Puoi sempre provare prima le istanze Spot e passare a On-Demand, se necessario.

Se hai i seguenti requisiti e aspettative, utilizza le istanze Amazon EC2 On-Demand:

- La durata dei tuoi processi è superiore a un'ora e non puoi tollerare interruzioni del carico di lavoro.
- Hai un SLO (obiettivo a livello di servizio) rigoroso per il tuo carico di lavoro complessivo e non puoi aumentare il tempo di calcolo.
- È più probabile che le istanze di cui hai bisogno subiscano interruzioni.

Se hai i seguenti requisiti e aspettative, utilizza le istanze Amazon EC2 Spot:

- La durata dei processi è in genere di 30 minuti o meno.
- Puoi tollerare potenziali interruzioni e riprogrammazioni dei lavori come parte del tuo carico di lavoro. Per ulteriori informazioni, consulta Spot Instance advisor.
- I lavori di lunga durata possono essere riavviati da un checkpoint se interrotti.

Puoi combinare entrambi i modelli di acquisto inviandoli prima su un'istanza Spot e poi utilizzando l'istanza On-Demand come opzione di riserva. Ad esempio, invia i tuoi lavori su una coda connessa ad ambienti di elaborazione in esecuzione su istanze Amazon EC2 Spot. Se un lavoro viene interrotto, cattura l'evento da Amazon EventBridge e correlalo a un recupero di un'istanza Spot. Quindi, invia nuovamente il lavoro a una coda On-Demand utilizzando una funzione o. AWS Lambda AWS Step Functions Per ulteriori informazioniTutorial: invio di avvisi di Amazon Simple Notification Service per eventi di lavoro non riusciti, consulta le best practice per la gestione delle interruzioni delle istanze Amazon EC2 Spot e Manage AWS Batch with Step Functions.

> **Important**
>
> Utilizza diversi tipi, dimensioni e zone di disponibilità per il tuo ambiente di calcolo On-Demand per mantenere la disponibilità del pool di istanze Amazon EC2 Spot e ridurre il tasso di interruzione.

Utilizza le best practice di Amazon EC2 Spot per AWS Batch

Quando scegli le istanze Spot di Amazon Elastic Compute Cloud (EC2), probabilmente puoi ottimizzare il flusso di lavoro per risparmiare sui costi, a volte in modo significativo. Per ulteriori informazioni, consulta Best practice per Amazon EC2 Spot.

Per ottimizzare il flusso di lavoro e ridurre i costi, prendi in considerazione le seguenti best practice di Amazon EC2 Spot per AWS Batch:

- Scegli la strategia di **`SPOT_CAPACITY_OPTIMIZED`** allocazione: AWS Batch seleziona EC2 le istanze Amazon dai pool di capacità Amazon Spot più profondi. EC2 Se sei preoccupato per le interruzioni, questa è la scelta giusta. Per ulteriori informazioni, consulta Strategie di allocazione del tipo di istanza per AWS Batch.
- Diversificate i tipi di istanze: per diversificare i tipi di istanze, prendete in considerazione dimensioni e famiglie compatibili, quindi lasciate che AWS Batch scegliate in base al prezzo o alla disponibilità. Ad esempio, `c5.24xlarge` considera un'alternativa a `c5.12xlarge` or`c5a`,, `c5n`c5d, `m5` e famiglie. `m5d` Per ulteriori informazioni, consulta Essere flessibili sui tipi di istanze e sulle zone di disponibilità.
- Riduci la durata del processo o il checkpoint: sconsigliamo di eseguire lavori che richiedono un'ora o più quando si utilizzano istanze Amazon EC2 Spot per evitare interruzioni. Se dividi o controlli i tuoi lavori in parti più piccole che durano 30 minuti o meno, puoi ridurre significativamente la possibilità di interruzioni.
- Utilizza nuovi tentativi automatici: per evitare interruzioni dei AWS Batch lavori, imposta nuovi tentativi automatici per i lavori. I processi Batch possono essere interrotti per uno dei seguenti motivi: viene restituito un codice di uscita diverso da zero, si verifica un errore di servizio o si verifica il recupero di un'istanza. È possibile impostare fino a 10 tentativi automatici. Per cominciare, ti consigliamo di impostare almeno 1-3 tentativi automatici. Per informazioni sul monitoraggio delle interruzioni di Amazon EC2 Spot, consulta Spot Interruption Dashboard.

Infatti AWS Batch, se imposti il parametro retry, il lavoro viene posizionato in primo piano nella coda dei lavori. Cioè, al lavoro viene data priorità. Quando si crea la definizione del processo o si invia il lavoro in AWS CLI, è possibile configurare una strategia di nuovo tentativo. Per ulteriori informazioni, consulta submit-job.

```
$ aws batch submit-job --job-name MyJob \
    --job-queue MyJQ \
    --job-definition MyJD \
    --retry-strategy attempts=2
```

- Utilizza nuovi tentativi: puoi configurare una strategia di ripetizione del processo in base a un codice di uscita specifico dell'applicazione o al recupero dell'istanza. Nell'esempio seguente, se l'host causa l'errore, il processo può essere riprovato fino a cinque volte. Tuttavia, se il processo fallisce per un motivo diverso, viene chiuso e lo stato viene impostato su. `FAILED`

```
"retryStrategy": {
    "attempts": 5,
    "evaluateOnExit":
    [{
        "onStatusReason" :"Host EC2*",
        "action": "RETRY"
    },{
      "onReason" : "*",
        "action": "EXIT"
    }]
}
```

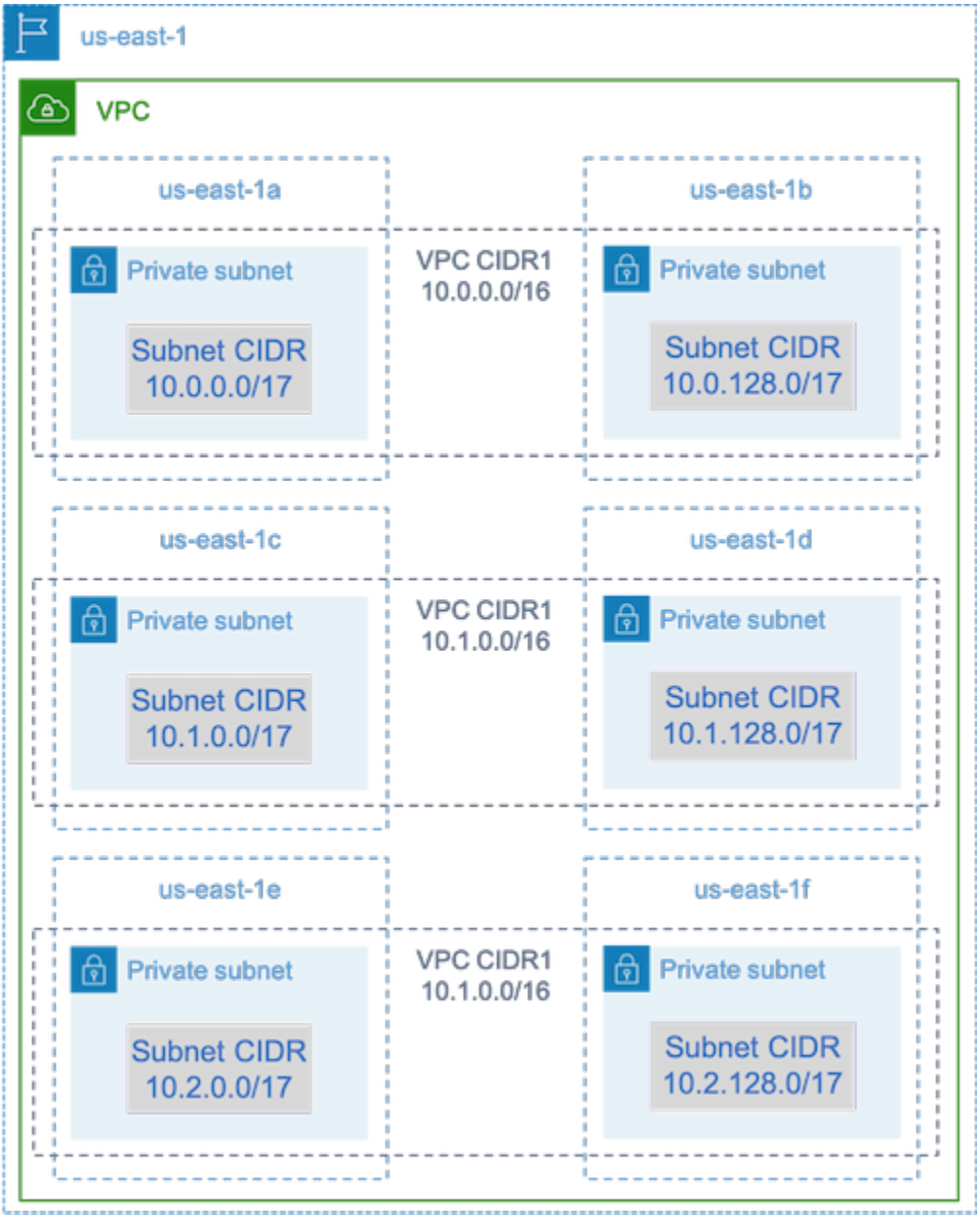
- Usa la dashboard di Spot Interruption: puoi utilizzare la dashboard di Spot Interruption per tenere traccia delle interruzioni di Spot. L'applicazione fornisce parametri sulle istanze Amazon EC2 Spot che vengono recuperate e sulle zone di disponibilità in cui si trovano le istanze Spot. Per ulteriori informazioni, consulta Spot Interruption Dashboard

Errori comuni e risoluzione dei problemi

Gli errori si verificano AWS Batch spesso a livello di applicazione o sono causati da configurazioni di istanza che non soddisfano i requisiti lavorativi specifici. Altri problemi includono lavori che rimangono bloccati nello `RUNNABLE` stato o gli ambienti di calcolo che rimangono bloccati in uno `INVALID` stato. Per ulteriori informazioni sulla risoluzione dei problemi relativi al blocco `RUNNABLE` dello stato dei

lavori, consultaLavori bloccati in uno RUNNABLE status. Per informazioni sulla risoluzione dei problemi degli ambienti di calcolo in uno INVALID stato, consultaINVALIDambiente di calcolo.

- Controlla le quote di vCPU di Amazon EC2 Spot: verifica che le quote di servizio attuali soddisfino i requisiti del lavoro. Ad esempio, supponiamo che la tua attuale quota di servizio sia 256 v CPUs e che il lavoro richieda 10.000 v. CPUs Quindi, la quota di servizio non soddisfa i requisiti del lavoro. Per ulteriori informazioni e istruzioni per la risoluzione dei problemi, consulta le quote di EC2 servizio Amazon e Come posso aumentare la quota di servizio del mio Amazon EC2resources? .
- I lavori falliscono prima dell'esecuzione dell'applicazione: alcuni processi potrebbero fallire a causa di un DockerTimeoutError errore o di un CannotPullContainerError errore. Per informazioni sulla risoluzione dei problemi, vedi Come si risolve l'errore DockerTimeoutError "" in AWS Batch? .
- Indirizzi IP insufficienti: il numero di indirizzi IP nel VPC e nelle sottoreti può limitare il numero di istanze che è possibile creare. Utilizza Classless Inter-Domain Routings (CIDRs) per fornire più indirizzi IP di quelli necessari per eseguire i carichi di lavoro. Se necessario, puoi anche creare un VPC dedicato con un ampio spazio di indirizzi. Ad esempio, puoi creare un VPC con più ingressi 10.x.0.0/16 e una sottorete CIDRs in ogni zona di disponibilità con un CIDR di. 10.x.y.0/17 In questo esempio, x è compreso tra 1-4 e y è 0 o 128. Questa configurazione fornisce 36.000 indirizzi IP in ogni sottorete.



- Verifica che le istanze siano registrate su Amazon EC2: se vedi le tue istanze nella EC2 console Amazon, ma nessuna istanza del contenitore Amazon Elastic Container Service nel tuo cluster Amazon ECS, l'agente Amazon ECS potrebbe non essere installato su un'Amazon Machine Image (AMI). Inoltre, l'agente Amazon ECS, gli Amazon EC2 Data nell'AMI o il modello di avvio potrebbero non essere configurati correttamente. Per isolare la causa principale, crea un' EC2 istanza Amazon separata o connettiti a un'istanza esistente tramite SSH. Per ulteriori informazioni, consulta la configurazione dell'agente container di Amazon ECS, le posizioni dei file di log di Amazon ECS e. Risorsa di calcolo AMIs

- Esamina la AWS dashboard: esamina la AWS dashboard per verificare che il processo previsto sia in stato e che l'ambiente di calcolo sia scalabile come previsto. Puoi anche controllare i registri dei lavori. CloudWatch
- Verifica che l'istanza sia stata creata: se viene creata un'istanza, significa che l'ambiente di calcolo è stato scalato come previsto. Se le tue istanze non sono state create, trova le sottoreti associate nel tuo ambiente di calcolo da modificare. Per ulteriori informazioni, consulta Verificare un'attività di ridimensionamento per un gruppo Auto Scaling.

  Ti consigliamo inoltre di verificare che le tue istanze siano in grado di soddisfare i relativi requisiti lavorativi. Ad esempio, un processo potrebbe richiedere 1 TiB di memoria, ma l'ambiente di calcolo utilizza un tipo di istanza C5 limitato a 192 GB di memoria.
- Verifica che le tue istanze siano state richieste da AWS Batch: controlla la cronologia del gruppo Auto Scaling per verificare che le tue istanze siano state richieste da. AWS Batch Questa è un'indicazione di come Amazon EC2 tenta di acquisire istanze. Se ricevi un errore che indica che Amazon EC2 Spot non può acquisire un'istanza in una zona di disponibilità specifica, ciò potrebbe essere dovuto al fatto che la zona di disponibilità non offre una famiglia di istanze specifica.
- Verifica che le istanze si registrino con Amazon ECS: se vedi istanze nella EC2 console Amazon, ma nessuna istanza di container Amazon ECS nel tuo cluster Amazon ECS, l'agente Amazon ECS potrebbe non essere installato su Amazon Machine Image (AMI). Inoltre, l'agente Amazon ECS, gli Amazon EC2 Data nell'AMI o il modello di avvio potrebbero non essere configurati correttamente. Per isolare la causa principale, crea un' EC2 istanza Amazon separata o connettiti a un'istanza esistente tramite SSH. Per ulteriori informazioni, consulta File di configurazione CloudWatch dell'agente: sezione Logs, Amazon ECS Log File Locations e. Risorsa di calcolo AMIs
- Apri un ticket di assistenza: se continui a riscontrare problemi dopo la risoluzione dei problemi e disponi di un piano di supporto, apri un ticket di supporto. Nel ticket di assistenza, assicurati di includere informazioni sul problema, le specifiche del carico di lavoro, la configurazione e i risultati dei test. Per ulteriori informazioni, consulta Confronta Supporto i piani.
- Consulta i forum AWS Batch e HPC: per ulteriori informazioni, consulta i forum AWS Batche HPC.
- Consulta la dashboard AWS Batch di monitoraggio del runtime: questa dashboard utilizza un'architettura serverless per acquisire eventi da Amazon ECS e Amazon EC2 per fornire informazioni dettagliate su processi e istanze. AWS Batch Per ulteriori informazioni, consulta AWS Batch Runtime Monitoring Dashboards Solution.

Risoluzione dei problemi AWS Batch

Potrebbe essere necessario risolvere problemi relativi agli ambienti di calcolo, alle code di lavoro, alle definizioni dei lavori o ai lavori. Questo capitolo descrive come risolvere tali problemi nell'ambiente in uso. AWS Batch

AWS Batch utilizza policy, ruoli e autorizzazioni IAM e funziona su Amazon EC2, Amazon ECS e Amazon Elastic AWS Fargate Kubernetes Service. Per risolvere i problemi relativi a questi servizi, consulta quanto segue:

- Risoluzione dei problemi di IAM nella IAM User Guide
- Risoluzione dei problemi di Amazon ECS nella Amazon Elastic Container Service Developer Guide
- Risoluzione dei problemi di Amazon EKS nella Guida per l'utente di Amazon EKS
- Risolvi i problemi relativi EC2 alle istanze nella Amazon User Guide EC2

Indice

AWS Batch

Consulta i seguenti argomenti per trovare i processi di revisione e le potenziali soluzioni ai problemi più comuni che potresti riscontrare durante l'utilizzo. AWS Batch

Argomenti

Configurazione ottimale del tipo di istanza per ricevere aggiornamenti automatici sulla famiglia di istanze

> **Note**
>
> A partire dall'11/01/2025, il comportamento di `optimal` verrà modificato per corrispondere. `default_x86_64` Durante la modifica, le famiglie di istanze potrebbero essere aggiornate a una nuova generazione. Non è necessario eseguire alcuna azione affinché l'aggiornamento avvenga.

AWS Batch supportava un'unica opzione in InstanceTypes `optimal` per soddisfare la domanda delle vostre code di lavoro. Abbiamo introdotto due nuove opzioni di tipo di istanza: e. `default_x86_64` `default_arm64` Le useremo `default_x86_64` se non effettui alcuna selezione del tipo di istanza. Queste nuove opzioni selezioneranno automaticamente tipi di istanze convenienti tra diverse famiglie e generazioni in base ai requisiti della coda di lavoro, consentendoti di far funzionare rapidamente i tuoi carichi di lavoro.

Non appena sarà disponibile una capacità sufficiente per nuovi tipi di istanze in un Regione AWS, il pool predefinito corrispondente verrà aggiornato automaticamente con il nuovo tipo di istanza. L'`optimal`opzione esistente continuerà a essere supportata e non è obsoleta, in quanto sarà supportata dai pool predefiniti sottostanti per fornire istanze aggiornate in futuro. Se stai usando '`optimal`, non è necessaria alcuna azione da parte tua.

Tuttavia, tieni presente che solo `ENABLED` and `VALID` Compute Environments (CEs) verrà aggiornato automaticamente con nuovi tipi di istanze. Se ne hai uno `DISABLED` o `INVALID` CEs, riceveranno aggiornamenti una volta riattivati e impostati su uno `VALID` stato.

`INVALID`ambiente di calcolo

È possibile che tu abbia configurato in modo errato un ambiente di elaborazione gestito. Se l'hai fatto, l'ambiente di elaborazione entra in uno `INVALID` stato e non può accettare offerte di lavoro per il collocamento. Le sezioni seguenti descrivono le possibili cause e come risolverli in base alla causa.

> **Important**
>
> AWS Batch crea e gestisce più AWS risorse per tuo conto e all'interno del tuo account, tra cui Amazon EC2 Launch Templates, Amazon EC2 Auto Scaling Groups, Amazon

EC2 Spot Fleets e Amazon ECS Clusters. Queste risorse gestite sono configurate specificamente per garantire un funzionamento ottimale. AWS Batch La modifica manuale di queste risorse gestite in batch, a meno che non sia esplicitamente indicato nella AWS Batch documentazione, può comportare un comportamento imprevisto con conseguente ambiente di `INVALID` calcolo, un comportamento di scalabilità delle istanze non ottimale, un'elaborazione ritardata del carico di lavoro o costi imprevisti. Queste modifiche manuali non possono essere supportate in modo deterministico dal servizio. AWS Batch Usa sempre il Batch supportato APIs o la console Batch per gestire i tuoi ambienti di elaborazione.

Nome del ruolo o ARN errati

La causa più comune per cui un ambiente di elaborazione entra in uno `INVALID` stato è che il ruolo di AWS Batch servizio o il ruolo Amazon EC2 Spot Fleet hanno un nome o un Amazon Resource Name (ARN) errati. Questo è più comune negli ambienti di elaborazione creati utilizzando o il AWS CLI . AWS SDKs Quando crei un ambiente di elaborazione in Console di gestione AWS, ti AWS Batch aiuta a scegliere il servizio o i ruoli corretti di Spot Fleet. Tuttavia, si supponga di immettere manualmente il nome o l'ARN e di inserirli in modo errato. Quindi, lo è anche l'ambiente di calcolo risultante. `INVALID`

Tuttavia, supponiamo di inserire manualmente il nome o l'ARN di una risorsa IAM in AWS CLI un comando o nel codice SDK. In questo caso, non è AWS Batch possibile convalidare la stringa. Invece, AWS Batch deve accettare il valore errato e tentare di creare l'ambiente. Se AWS Batch non riesce a creare l'ambiente, l'ambiente passa a uno `INVALID` stato e vengono visualizzati i seguenti errori.

In caso di ruolo del servizio non valido:

```
CLIENT_ERROR - Not authorized to perform sts:AssumeRole (Service:
AWSSecurityTokenService; Status Code: 403; Error Code: AccessDenied;
Request ID: dc0e2d28-2e99-11e7-b372-7fcc6fb65fe7)
```

In caso di parco istanze Spot non valido:

```
CLIENT_ERROR - Parameter: SpotFleetRequestConfig.IamFleetRole
is invalid. (Service: AmazonEC2; Status Code: 400; Error Code:
InvalidSpotFleetRequestConfig; Request ID: 331205f0-5ae3-4cea-
bac4-897769639f8d) Parameter: SpotFleetRequestConfig.IamFleetRole is
invalid
```

Una delle cause più comuni di questo problema è lo scenario seguente. Specifichi il nome di un ruolo IAM solo quando usi AWS CLI o il AWS SDKs, anziché l'Amazon Resource Name (ARN) completo. A seconda di come è stato creato il ruolo, l'ARN potrebbe contenere un prefisso di `aws-service-role` percorso. Ad esempio, se si crea manualmente il ruolo di AWS Batch servizio utilizzando le procedure in[Utilizzo di ruoli collegati ai servizi per AWS Batch](), l'ARN del ruolo di servizio potrebbe essere simile al seguente.

```
arn:aws:iam::123456789012:role/AWSBatchServiceRole
```

Tuttavia, se oggi hai creato il ruolo di servizio come parte della procedura guidata per la prima esecuzione della console, il tuo ruolo di servizio ARN potrebbe essere simile al seguente.

```
arn:aws:iam::123456789012:role/aws-service-role/AWSBatchServiceRole
```

Questo problema può verificarsi anche se si associa la policy a AWS Batch livello di servizio (`AWSBatchServiceRole`) a un ruolo non di servizio. Ad esempio, in questo scenario è possibile che venga visualizzato un messaggio di errore analogo al seguente:

```
CLIENT_ERROR - User: arn:aws:sts::account_number:assumed-role/batch-replacement-role/
aws-batch is not
   authorized to perform: action on resource ...
```

Per risolvere questo problema, effettuate una delle seguenti operazioni.

- Usa una stringa vuota per il ruolo di servizio quando crei l'ambiente di AWS Batch calcolo.
- Specificare il ruolo di servizio nel seguente formato:`arn:aws:iam::account_number:role/aws-service-role/batch.amazonaws.com/AWSServiceRoleForBatch`.

Quando si specifica solo il nome di un ruolo IAM quando si utilizza AWS CLI o il AWS SDKs, si AWS Batch presuppone che l'ARN non utilizzi il prefisso `aws-service-role` del percorso. Per questo motivo, ti consigliamo di specificare l'ARN completo per i tuoi ruoli IAM quando crei ambienti di calcolo.

Per riparare un ambiente di calcolo configurato in modo errato in questo modo, consulta. [Riparazione di un ambiente di elaborazione `INVALID`]()

Riparazione di un ambiente di elaborazione **`INVALID`**

Quando hai un ambiente di calcolo in uno `INVALID` stato, aggiornalo per correggere il parametro non valido. Ad esempioNome del ruolo o ARN errati, aggiorna l'ambiente di calcolo utilizzando il ruolo di servizio corretto.

Riparazione di un ambiente di calcolo configurato in modo errato

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Dalla barra di navigazione, seleziona l'opzione Regione AWS da utilizzare.
3. Nel riquadro di navigazione, seleziona Compute environments (Ambienti di calcolo).
4. Nella pagina Compute environments (Ambienti di calcolo), seleziona il pulsante accanto all'ambiente di calcolo da modificare, quindi scegliere Edit (Modifica).
5. Nella pagina Aggiorna ambiente di calcolo, per il ruolo di servizio, scegli il ruolo IAM da utilizzare con il tuo ambiente di calcolo. La console di AWS Batch visualizza solo i ruoli con una relazione di trust corretta per gli ambienti di calcolo.

   > **Tip**
   >
   > Per istruzioni su come creare un ruolo collegato al servizio, consulta. Utilizzo dei ruoli per AWS Batch

6. Seleziona Save (Salva) per aggiornare l'ambiente di calcolo.

Lavori bloccati in uno **`RUNNABLE`** status

Supponiamo che l'ambiente di elaborazione contenga risorse di elaborazione, ma che i lavori non vadano oltre lo stato. `RUNNABLE` Quindi, è probabile che qualcosa impedisca l'inserimento dei lavori su una risorsa di elaborazione e causi il blocco delle code di lavoro. Ecco come sapere se il lavoro è in attesa del suo turno o se è bloccato e blocca la coda.

Se AWS Batch rileva che hai un `RUNNABLE` lavoro a capo e blocca la coda, riceverai un Eventi bloccati in Job queue evento da Amazon CloudWatch Events con il motivo. Lo stesso motivo viene aggiornato anche nel `statusReason` campo come parte delle chiamate `ListJobs` `DescribeJobs` API.

Facoltativamente, puoi configurare il `jobStateTimeLimitActions` parametro tramite `CreateJobQueue` azioni `UpdateJobQueue`API.

> **Note**
>
> Attualmente, l'unica azione che puoi eseguire `jobStateLimitActions.action` è annullare un lavoro.

Il `jobStateTimeLimitActions` parametro viene utilizzato per specificare una serie di azioni da AWS Batch eseguire sui lavori in uno stato specifico. È possibile impostare una soglia temporale in secondi tramite il `maxTimeSeconds` campo.

Quando un lavoro si trova in uno `RUNNABLE` stato definito`statusReason`, AWS Batch esegue l'azione specificata dopo `maxTimeSeconds` che è trascorsa.

Ad esempio, è possibile impostare il `jobStateTimeLimitActions` parametro in modo che attenda fino a 4 ore per qualsiasi lavoro nello `RUNNABLE` stato in cui è in attesa che diventi disponibile una capacità sufficiente. È possibile farlo impostando `statusReason` su `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY` e `maxTimeSeconds` su 144000 prima di annullare il lavoro e consentire al lavoro successivo di passare in testa alla coda dei lavori.

Di seguito sono riportati i motivi che AWS Batch fornisce quando rileva che una coda di lavori è bloccata. Questo elenco fornisce i messaggi restituiti dalle azioni `ListJobs` e `DescribeJobs` API. Questi sono anche gli stessi valori che è possibile definire per il `jobStateLimitActions.statusReason` parametro.

1. Motivo: tutti gli ambienti di elaborazione connessi presentano errori di capacità insufficienti. Quando richiesto, AWS Batch rileva le EC2 istanze Amazon che presentano errori di capacità insufficiente. L'annullamento manuale del lavoro consentirà al lavoro successivo di passare in testa alla coda, ma senza risolvere il problema o i problemi relativi al ruolo di servizio, è probabile che anche il lavoro successivo venga bloccato. È consigliabile esaminare e risolvere il problema manualmente.
   - **`statusReason`**messaggio mentre il lavoro è bloccato: `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY - Service cannot fulfill the capacity requested for instance type [instanceTypeName]`
   - **`reason`**usato per**`jobStateTimeLimitActions`**: `CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`
   - **`statusReason`**messaggio dopo l'annullamento del lavoro: `Canceled by JobStateTimeLimit action due to reason: CAPACITY:INSUFFICIENT_INSTANCE_CAPACITY`

Nota:

a. Il ruolo AWS Batch di servizio richiede `autoscaling:DescribeScalingActivities` l'autorizzazione affinché questo rilevamento funzioni. Se utilizzi il ruolo Utilizzo di ruoli collegati ai servizi per AWS Batch collegato al servizio (SLR) o la policy AWS politica gestita: AWSBatchServiceRolepolitica gestita, non devi intraprendere alcuna azione perché le relative politiche di autorizzazione vengono aggiornate.

b. Se si utilizza la SLR o la policy gestita, è necessario aggiungere le `ec2:DescribeSpotFleetRequestHistory` autorizzazioni `autoscaling:DescribeScalingActivities` e in modo da poter ricevere gli eventi bloccati della coda dei lavori e lo stato aggiornato dei lavori quando si accede. `RUNNABLE` Inoltre, sono AWS Batch necessarie queste autorizzazioni per eseguire `cancellation` azioni tramite il `jobStateTimeLimitActions` parametro anche se sono configurate nella coda dei lavori.

c. Nel caso di un lavoro parallelo a più nodi (MNP), se l'ambiente di EC2 calcolo Amazon ad alta priorità collegato presenta `insufficient capacity` errori, blocca la coda anche se un ambiente di calcolo con priorità inferiore presenta questo errore.

2. Motivo: tutti gli ambienti di elaborazione hanno un `maxvCpus`parametro inferiore ai requisiti del lavoro. L'annullamento del lavoro, manualmente o impostando il `jobStateTimeLimitActions` parametro on`statusReason`, consente al lavoro successivo di passare in testa alla coda. Facoltativamente, è possibile aumentare il `maxvCpus` parametro dell'ambiente di calcolo primario per soddisfare le esigenze del lavoro bloccato.

   - **`statusReason`**messaggio mentre il lavoro è bloccato: `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE - CE(s) associated with the job queue cannot meet the CPU requirement of the job.`
   - **`reason`**usato per**`jobStateTimeLimitActions`**: `MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`
   - **`statusReason`**messaggio dopo l'annullamento del lavoro: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:COMPUTE_ENVIRONMENT_MAX_RESOURCE`

3. Motivo: nessuno degli ambienti di elaborazione dispone di istanze che soddisfano i requisiti del lavoro. Quando un lavoro richiede risorse, AWS Batch rileva che nessun ambiente di calcolo collegato è in grado di ospitare il lavoro in arrivo. L'annullamento del lavoro, manualmente o impostando il `jobStateTimeLimitActions` parametro on`statusReason`, consente al lavoro successivo di passare in testa alla coda. Facoltativamente, puoi ridefinire i tipi di istanze consentiti nell'ambiente di calcolo per aggiungere le risorse di lavoro necessarie.

- **`statusReason`**messaggio mentre il lavoro è bloccato: `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT - The job resource requirement (vCPU/memory/GPU) is higher than that can be met by the CE(s) attached to the job queue.`
- **`reason`**usato per**`jobStateTimeLimitActions`**: `MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`
- **`statusReason`**messaggio dopo l'annullamento del lavoro: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:JOB_RESOURCE_REQUIREMENT`

4. Motivo: tutti gli ambienti di elaborazione presentano problemi relativi ai ruoli di servizio. Per risolvere questo problema, confronta le autorizzazioni del ruolo di servizio con le AWS politiche gestite per AWS Batch e risolvi eventuali lacune. Nota: non è possibile configurare un'azione programmabile tramite il parametro per risolvere questo errore. `jobStateTimeLimitActions`

   È consigliabile utilizzare il per evitare errori similiUtilizzo di ruoli collegati ai servizi per AWS Batch.

   L'annullamento del lavoro, manualmente o impostando il `jobStateTimeLimitActions` parametro on`statusReason`, consente al lavoro successivo di passare in testa alla coda. Senza risolvere il problema o i problemi relativi al ruolo di servizio, è probabile che anche il lavoro successivo venga bloccato. È consigliabile esaminare e risolvere il problema manualmente.

   - **`statusReason`**messaggio mentre il lavoro è bloccato: `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS – Batch service role has a permission issue.`

5. Motivo: il tuo ambiente di calcolo ha una configurazione del tipo di istanza non supportata. Ciò può verificarsi quando i tipi di istanza non sono disponibili nelle zone di disponibilità selezionate o quando il modello o la configurazione di avvio contengono impostazioni incompatibili con i tipi di istanza specificati. Per risolvere il problema, verifica che i tipi di istanza siano supportati nelle zone di disponibilità specificate Regione AWS e nelle zone di disponibilità, verifica che le impostazioni del modello di lancio siano compatibili con i tipi di istanza e valuta la possibilità di eseguire l'aggiornamento a tipi di istanza di nuova generazione. Per ulteriori informazioni sulla ricerca dei tipi di istanza supportati, consulta Finding an Amazon EC2 istance type nella Amazon EC2 User Guide.

   - **`statusReason`**messaggio mentre il lavoro è bloccato: `MISCONFIGURATION:EC2_INSTANCE_CONFIGURATION_UNSUPPORTED - Your compute environment associated with this job queue has an unsupported instance type configuration.`

6. Motivo: tutti gli ambienti di elaborazione non sono validi. Per ulteriori informazioni, consulta `INVALID`ambiente di calcolo. Nota: non è possibile configurare un'azione programmabile tramite il `jobStateTimeLimitActions` parametro per risolvere questo errore.
   - **`statusReason`**messaggio mentre il lavoro è bloccato: `ACTION_REQUIRED - CE(s) associated with the job queue are invalid.`
7. Motivo: AWS Batch ha rilevato una coda bloccata, ma non è in grado di determinarne il motivo. Nota: non è possibile configurare un'azione programmabile tramite il `jobStateTimeLimitActions` parametro per risolvere questo errore. Per ulteriori informazioni sulla risoluzione dei problemi, vedi Perché il mio AWS Batch lavoro è bloccato in RUNNABLE on in Re:post. AWS
   - **`statusReason`**messaggio mentre il lavoro è bloccato: `UNDETERMINED - Batch job is blocked, root cause is undetermined.`

Se non hai ricevuto un evento da CloudWatch Events o hai ricevuto l'evento con motivo sconosciuto, ecco alcune cause comuni di questo problema.

Il driver di **`awslogs`** registro non è configurato sulle tue risorse di calcolo

AWS Batch i job inviano le proprie informazioni di registro a CloudWatch Logs. Per abilitare questa opzione, è necessario configurare le risorse di calcolo per utilizzare il driver di log `awslogs`. Supponiamo di basare l'AMI delle risorse di calcolo sull'AMI ottimizzata per Amazon ECS (o Amazon Linux). Quindi, questo driver viene registrato per impostazione predefinita nel pacchetto. `ecs-init` Supponiamo ora di utilizzare un'AMI di base diversa. Quindi, è necessario verificare che il driver di `awslogs` log sia specificato come driver di registro disponibile con la variabile di `ECS_AVAILABLE_LOGGING_DRIVERS` ambiente all'avvio dell'agente container Amazon ECS. Per ulteriori informazioni, consultare Specifiche dell'AMI delle risorse di calcolo e Tutorial: Creare un'AMI per risorse di calcolo.

Risorse insufficienti

Se le definizioni dei processi specificano più risorse di CPU o memoria di quelle che le risorse di elaborazione possono allocare, i lavori non vengono mai collocati. Ad esempio, supponiamo che il tuo job specifichi 4 GiB di memoria e che le tue risorse di calcolo abbiano meno di quella disponibile. Quindi accade che il lavoro non possa essere collocato su quelle risorse di calcolo. In tal caso, è necessario ridurre la quantità di memoria specificata nella definizione di processo o aggiungere risorse di calcolo maggiori all'ambiente. Parte della memoria è riservata all'agente container Amazon ECS e ad altri processi di sistema critici. Per ulteriori informazioni, consulta Gestione della memoria delle risorse di calcolo.

Nessun accesso a Internet per le risorse di elaborazione

Le risorse di calcolo richiedono un accesso per comunicare con l'endpoint del servizio Amazon ECS. Ciò può avvenire attraverso un endpoint VPC di interfaccia o tramite risorse di calcolo con indirizzi IP pubblici.

Per ulteriori informazioni sugli endpoint di interfaccia Amazon ECR, consulta Endpoint VPC dell'interfaccia Amazon ECS (AWS PrivateLink) nella Guida per gli sviluppatori di Amazon Elastic Container Service.

Se non disponi di un endpoint VPC di interfaccia configurato e le risorse di calcolo non dispongono di indirizzi IP pubblici, per fornire questo accesso devono utilizzare il processo Network Address Translation (NAT). Per ulteriori informazioni, consulta Gateway NATnella Guida per l'utente di Amazon VPC. Per ulteriori informazioni, consulta the section called "Crea un VPC".

Limite di EC2 istanze Amazon raggiunto

Il numero di EC2 istanze Amazon in cui il tuo account può avviare Regione AWS è determinato dalla quota di EC2 istanze. Alcuni tipi di istanze hanno anche una per-instance-type quota. Per ulteriori informazioni sulla quota di EC2 istanze Amazon del tuo account, incluso come richiedere un aumento del limite, consulta Amazon EC2 Service Limits nella Amazon EC2 User Guide.

L'agente container Amazon ECS non è installato

L'agente contenitore Amazon ECS deve essere installato su Amazon Machine Image (AMI) per consentire l' AWS Batch esecuzione dei lavori. L'agente contenitore Amazon ECS è installato per impostazione predefinita su Amazon ECS ottimizzato. AMIs Per ulteriori informazioni sull'agente container Amazon ECS, consulta l'agente container Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Per ulteriori informazioni, consulta Perché il mio AWS Batch lavoro è bloccato nello `RUNNABLE` status? in Re:post.

Istanze Spot non taggate al momento della creazione

L'etichettatura delle istanze Spot per le risorse di AWS Batch calcolo è supportata a partire dal 25 ottobre 2017. In precedenza, la policy gestita IAM consigliata (`AmazonEC2SpotFleetRole`) per il ruolo Amazon EC2 Spot Fleet non conteneva le autorizzazioni per etichettare le istanze Spot al momento del lancio. Viene chiamata la nuova policy gestita IAM consigliata. `AmazonEC2SpotFleetTaggingRole` Supporta l'etichettatura delle istanze Spot al momento del lancio.

Per correggere il tagging delle istanze Spot al momento della creazione, segui la seguente procedura per applicare l'attuale policy gestita IAM consigliata al tuo ruolo di Amazon EC2 Spot Fleet. In questo modo, tutte le future istanze Spot create con quel ruolo dispongono delle autorizzazioni per applicare i tag delle istanze al momento della creazione.

Per applicare l'attuale policy gestita da IAM al tuo ruolo nella flotta Amazon EC2 Spot

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Ruoli e scegli il tuo ruolo in Amazon EC2 Spot Fleet.
3. Scegli Collega policy.
4. Seleziona Amazon EC2 SpotFleetTaggingRole e scegli la politica di Allega.
5. Scegli nuovamente il tuo ruolo in Amazon EC2 Spot Fleet per rimuovere la politica precedente.
6. Seleziona la x a destra della EC2 SpotFleetRole politica di Amazon e scegli Detach.

Le istanze Spot non si ridimensionano

AWS Batch ha introdotto il ruolo AWSServiceRoleForBatchcollegato ai servizi il 10 marzo 2021. Se non viene specificato alcun ruolo nel `serviceRole` parametro dell'ambiente di calcolo, questo ruolo collegato al servizio viene utilizzato come ruolo di servizio. Tuttavia, supponiamo che il ruolo collegato al servizio venga utilizzato in un ambiente di calcolo EC2 Spot, ma che il ruolo Spot utilizzato non includa la policy gestita da Amazon. EC2 SpotFleetTaggingRole Quindi, l'istanza Spot non viene ridimensionata. Di conseguenza, riceverai un errore con il seguente messaggio: «Non sei autorizzato a eseguire questa operazione». Utilizza i seguenti passaggi per aggiornare il ruolo Spot Fleet che utilizzi nel `spotIamFleetRole` parametro. Per ulteriori informazioni, consulta Utilizzo dei ruoli collegati ai servizi e Creazione di un ruolo per delegare le autorizzazioni a un AWS servizio nella Guida per l'utente IAM.

Argomenti

- Associa la politica EC2 SpotFleetTaggingRole gestita da Amazon al tuo ruolo nella flotta Spot nel Console di gestione AWS
- Associa la politica EC2 SpotFleetTaggingRole gestita da Amazon al tuo ruolo nella flotta Spot con AWS CLI

Associa la politica EC2 SpotFleetTaggingRole gestita da Amazon al tuo ruolo nella flotta Spot nel Console di gestione AWS

Per applicare l'attuale policy gestita da IAM al tuo ruolo nella flotta Amazon EC2 Spot

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli Ruoli e scegli il tuo ruolo in Amazon EC2 Spot Fleet.
3. Scegli Collega policy.
4. Seleziona Amazon EC2 SpotFleetTaggingRole e scegli la politica di Allega.
5. Scegli nuovamente il tuo ruolo in Amazon EC2 Spot Fleet per rimuovere la politica precedente.
6. Seleziona la x a destra della EC2 SpotFleetRole politica di Amazon e scegli Detach.

Associa la politica EC2 SpotFleetTaggingRole gestita da Amazon al tuo ruolo nella flotta Spot con AWS CLI

I comandi di esempio presuppongono che il tuo ruolo Amazon EC2 Spot Fleet abbia un nome*AmazonEC2SpotFleetRole*. Se il tuo ruolo utilizza un nome diverso, modifica i comandi in modo che corrispondano.

Per associare la politica EC2 SpotFleetTaggingRole gestita da Amazon al tuo ruolo di Flotta Spot

1. Per collegare la policy Amazon EC2 SpotFleetTaggingRole managed IAM al tuo *AmazonEC2SpotFleetRole* ruolo, esegui il comando seguente utilizzando AWS CLI.

```
$ aws iam attach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetTaggingRole
 \
    --role-name AmazonEC2SpotFleetRole
```

2. Per scollegare la policy Amazon EC2 SpotFleetRole managed IAM dal tuo *AmazonEC2SpotFleetRole* ruolo, esegui il comando seguente utilizzando. AWS CLI

```
$ aws iam detach-role-policy \
    --policy-arn arn:aws:iam::aws:policy/service-role/AmazonEC2SpotFleetRole \
    --role-name AmazonEC2SpotFleetRole
```

Impossibile recuperare i segreti di Secrets Manager

Se utilizzi un'AMI con un agente Amazon ECS precedente alla versione 1.16.0-1, devi utilizzare la variabile `ECS_ENABLE_AWSLOGS_EXECUTIONROLE_OVERRIDE=true` di configurazione dell'agente Amazon ECS per utilizzare questa funzionalità. Puoi aggiungerla al `./etc/ecs/ecs.config` file di una nuova istanza di contenitore quando crei quell'istanza. In alternativa, puoi aggiungerlo a un'istanza esistente. Se lo aggiungi a un'istanza esistente, devi riavviare l'agente ECS dopo averlo aggiunto. Per ulteriori informazioni, consulta Configurazione dell'agente del container Amazon ECS nella Guida per gli sviluppatori di Amazon Elastic Container Service.

Impossibile sovrascrivere i requisiti di risorse per la definizione del processo

Le sostituzioni di memoria e vCPU specificate nella struttura ContainerOverrides `memory` e dei `vcpus` membri della struttura ContainerOverrides, passata a, non SubmitJobpossono sovrascrivere i requisiti di memoria e vCPU specificati nella struttura ResourceRequirements nella definizione del processo.

Se si tenta di ignorare questi requisiti di risorse, è possibile che venga visualizzato il seguente messaggio di errore:

«Questo valore è stato inviato in una chiave obsoleta e potrebbe essere in conflitto con il valore fornito dai requisiti di risorse della definizione del processo».

Per correggere questo problema, specifica i requisiti di memoria e vCPU nel membro ResourceRequirements di ContainerOverrides. Ad esempio, se le sostituzioni di memoria e vCPU sono specificate nelle righe seguenti.

```
"containerOverrides": {
   "memory": 8192,
   "vcpus": 4
}
```

Modificali come segue:

```
"containerOverrides": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "8192"
```

```
      },
      {
         "type": "VCPU",
         "value": "4"
      }
   ],
}
```

Effettua la stessa modifica ai requisiti di memoria e vCPU specificati nell'oggetto ContainerProperties nella definizione del processo. Ad esempio, se i requisiti di memoria e vCPU sono specificati nelle righe seguenti.

```
{
   "containerProperties": {
      "memory": 4096,
      "vcpus": 2,
}
```

Modificali nel modo seguente:

```
"containerProperties": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "4096"
      },
      {
         "type": "VCPU",
         "value": "2"
      }
   ],
}
```

Messaggio di errore quando si aggiorna l'**desiredvCpus**impostazione

Quando si utilizza l' AWS Batch API per aggiornare l'impostazione v CPUs (`desiredvCpus`) desiderata, viene visualizzato il seguente messaggio di errore.

```
Manually scaling down compute environment is not supported. Disconnecting job queues from compute environment will cause it to scale-down to minvCpus.
```

Questo problema si verifica se il `desiredvCpus` valore aggiornato è inferiore al `desiredvCpus` valore corrente. Quando si aggiorna il `desiredvCpus` valore, devono essere soddisfatte entrambe le seguenti condizioni:

- Il `desiredvCpus` valore deve essere compreso tra i `maxvCpus` valori `minvCpus` e.
- Il `desiredvCpus` valore aggiornato deve essere maggiore o uguale al `desiredvCpus` valore corrente.

AWS Batch su Amazon EKS

Argomenti

Consulta i seguenti argomenti per trovare i processi di revisione e le potenziali soluzioni ai problemi più comuni che potresti riscontrare durante l'utilizzo AWS Batch su Amazon Elastic Kubernetes Service.

`INVALID`ambiente di calcolo

È possibile che tu abbia configurato in modo errato un ambiente di elaborazione gestito. Se l'hai fatto, l'ambiente di elaborazione entra in uno `INVALID` stato e non può accettare offerte di lavoro per il collocamento. Le sezioni seguenti descrivono le possibili cause e come risolverli in base alla causa.

Versione non supportata Kubernetes

È possibile che venga visualizzato un messaggio di errore analogo al seguente quando si utilizza l'operazione `CreateComputeEnvironment` API o l'operazione `UpdateComputeEnvironment` API per creare o aggiornare un ambiente di calcolo. Questo problema si verifica se si specifica una versione non supportata Kubernetes in. `EC2Configuration`

```
At least one imageKubernetesVersion in EC2Configuration is not supported.
```

Per risolvere questo problema, elimina l'ambiente di calcolo e quindi ricrealo con una versione supportata. Kubernetes

Puoi eseguire un aggiornamento di versione minore sul tuo cluster Amazon EKS. Ad esempio, puoi aggiornare il cluster da `1.xx` a `1.yy` anche se la versione secondaria non è supportata.

Tuttavia, lo stato dell'ambiente di calcolo potrebbe cambiare `INVALID` dopo un aggiornamento della versione principale. Ad esempio, se si esegue un aggiornamento della versione principale da `1.xx` a`2.yy`. Se la versione principale non è supportata da AWS Batch, viene visualizzato un messaggio di errore analogo al seguente.

```
reason=CLIENT_ERROR - ... EKS Cluster version [2.yy] is unsupported
```

Per risolvere questo problema, specifica una Kubernetes versione supportata quando utilizzi un'operazione API per creare o aggiornare un ambiente di calcolo.

AWS Batch su Amazon EKS attualmente supporta le seguenti Kubernetes versioni:

- `1.33`
- `1.32`
- `1.31`
- `1.30`
- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`

Il profilo dell'istanza non esiste

Se il profilo dell'istanza specificato non esiste, lo stato dell'ambiente di calcolo AWS Batch su Amazon EKS viene modificato in`INVALID`. Nel `statusReason` parametro viene visualizzato un errore simile al seguente.

```
CLIENT_ERROR - Instance profile arn:aws:iam::...:instance-profile/<name> does not exist
```

Per risolvere questo problema, specifica o crea un profilo dell'istanza di lavoro. Per ulteriori informazioni, consulta Ruolo IAM del nodo di Amazon EKS nella Guida per l'utente di Amazon EKS.

Namespace non valido Kubernetes

Se AWS Batch su Amazon EKS non è possibile convalidare lo spazio dei nomi per l'ambiente di calcolo, lo stato dell'ambiente di calcolo viene modificato in. `INVALID` Ad esempio, questo problema può verificarsi se lo spazio dei nomi non esiste.

Nel `statusReason` parametro viene visualizzato un messaggio di errore simile al seguente.

```
CLIENT_ERROR - Unable to validate Kubernetes Namespace
```

Questo problema può verificarsi se si verifica una delle seguenti condizioni:

- La stringa Kubernetes dello spazio dei nomi nella `CreateComputeEnvironment` chiamata non esiste. Per ulteriori informazioni, consulta CreateComputeEnvironment.
- Le autorizzazioni RBAC (Role-Based Access Control) necessarie per gestire lo spazio dei nomi non sono configurate correttamente.
- AWS Batch non ha accesso all'endpoint del server Kubernetes API Amazon EKS.

Per risolvere il problema, consulta Verificare che `aws-auth ConfigMap` sia configurato correttamente. Per ulteriori informazioni, consulta Guida introduttiva ad AWS Batch Amazon EKS.

Ambiente di elaborazione eliminato

Supponiamo di eliminare un cluster Amazon EKS prima di eliminare l'ambiente di calcolo allegato AWS Batch su Amazon EKS. Quindi, lo stato dell'ambiente di calcolo viene modificato in. `INVALID` In questo scenario, l'ambiente di calcolo non funziona correttamente se si ricrea il cluster Amazon EKS con lo stesso nome.

Per risolvere questo problema, elimina e ricrea l'ambiente di calcolo AWS Batch su Amazon EKS.

I nodi non entrano a far parte del cluster Amazon EKS

AWS Batch su Amazon EKS ridimensiona un ambiente di elaborazione se determina che non tutti i nodi si sono uniti al cluster Amazon EKS. Quando AWS Batch Amazon EKS ridimensiona l'ambiente di elaborazione, lo stato dell'ambiente di calcolo viene modificato in. `INVALID`

> **Note**
>
> AWS Batch non modifica immediatamente lo stato dell'ambiente di calcolo in modo da poter eseguire il debug del problema.

Nel `statusReason` parametro viene visualizzato un messaggio di errore simile a uno dei seguenti:

```
Your compute environment has been INVALIDATED and scaled down because
none of the instances joined the underlying ECS Cluster. Common issues
preventing instances joining are the following: VPC/Subnet configuration
preventing communication to ECS, incorrect Instance Profile policy
preventing authorization to ECS, or customized AMI or LaunchTemplate
configurations affecting ECS agent.
```

```
Your compute environment has been INVALIDATED and scaled down because
none of the nodes joined the underlying Amazon EKS Cluster. Common issues
preventing nodes joining are the following: networking configuration
preventing communication to Amazon EKS Cluster, incorrect Amazon EKS
Instance Profile or Kubernetes RBAC policy preventing authorization
to Amazon EKS Cluster, customized AMI or LaunchTemplate configurations
affecting Amazon EKS/Kubernetes node bootstrap.
```

Quando si utilizza un'AMI Amazon EKS predefinita, le cause più comuni di questo problema sono le seguenti:

- Il ruolo dell'istanza non è configurato correttamente. Per ulteriori informazioni, consulta Ruolo IAM del nodo di Amazon EKS nella Guida per l'utente di Amazon EKS.
- Le sottoreti non sono configurate correttamente. Per ulteriori informazioni, consulta i requisiti e le considerazioni su VPC e sottorete di Amazon EKS nella Guida per l'utente di Amazon EKS.
- Il gruppo di sicurezza non è configurato correttamente. Per ulteriori informazioni, consulta i requisiti e le considerazioni dei gruppi di sicurezza di Amazon EKS nella Guida per l'utente di Amazon EKS.

> **Note**
>
> È inoltre possibile visualizzare una notifica di errore nella Personal Health Dashboard (PHD).

AWS Batch su Amazon EKS il lavoro è bloccato **RUNNABLE**

Uno `aws-auth` `ConfigMap` viene creato e applicato automaticamente al cluster quando si crea un gruppo di nodi gestito o un gruppo di nodi utilizzando. `eksctl` Un `aws-auth` `ConfigMap` viene inizialmente creato per consentire ai nodi di unirsi al cluster. Tuttavia, lo si utilizza anche `aws-auth` `ConfigMap` per aggiungere l'accesso RBAC (Role-Based Access Control) a utenti e ruoli.

Per verificare che sia configurato correttamente: `aws-auth` `ConfigMap`

1. Recupera i ruoli mappati in: `aws-auth` `ConfigMap`

   ```
   $ kubectl get configmap -n kube-system aws-auth -o yaml
   ```

2. Verificare che `roleARN` sia configurato come segue.

   ```
   rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
   ```

   > **Note**
   >
   > Puoi anche esaminare i log del piano di controllo di Amazon EKS. Per ulteriori informazioni, consulta la registrazione del piano di controllo di Amazon EKS nella Guida per l'utente di Amazon EKS.

Per risolvere un problema a causa del quale un lavoro è bloccato in uno `RUNNABLE` stato, ti consigliamo di utilizzare questa opzione `kubectl` per riapplicare il manifesto. Per ulteriori informazioni, consulta Fase 2: prepara il cluster Amazon EKS per AWS Batch. In alternativa, puoi utilizzare `kubectl` per modificare manualmente il `aws-authConfigMap`. Per ulteriori informazioni, consulta Abilitare l'accesso di utenti e ruoli IAM al cluster nella Amazon EKS User Guide.

AWS Batch su Amazon EKS il lavoro è bloccato **STARTING**

Un Job può rimanere in `STARTING` stato quando il Pod è bloccato `ContainerCreating` per eventuali richieste di lunga durata provenienti dal kubelet (`pull`, `logexec`, `eattach`) fino alla risoluzione del problema di avvio del Pod o alla fine del Job. `PENDING` Negli scenari idonei riportati di seguito, il lavoro AWS Batch verrà interrotto per conto dell'utente, altrimenti il lavoro dovrà essere terminato manualmente utilizzando l'API. TerminateJob

Per verificare il motivo per cui un Job può essere bloccato`STARTING`, usa Tutorial: mappa un job in esecuzione su un pod e un nodo per trovare e descrivere il Pod: `podName`

```
% kubectl describe pod aws-batch.000c8190-87df-31e7-8819-176fe017a24a -n my-aws-batch-
namespace
Name:             aws-batch.000c8190-87df-31e7-8819-176fe017a24a
Namespace:        my-aws-batch-namespace
...
Containers:
  default:
...
    State:          Waiting
      Reason:       ContainerCreating
    Ready:          False
...
Conditions:
  Type                        Status
  PodReadyToStartContainers   False
  Initialized                 True
  Ready                       False
  ContainersReady             False
  PodScheduled                True
...
Events:
  Type     Reason       Age    From     Message
  ----     ------       ----   ----     -------
  Warning  FailedMount  2m32s  kubelet  Unable to attach or mount volumes: ...
```

Prendi in considerazione la possibilità di configurare il tuo cluster EKS per inviare i log del piano di controllo ai CloudWatch registri per una visibilità completa.

Scenario: errore persistente di richiesta di volume, collegamento o montaggio

I lavori che utilizzano Persistent Volume Claims in cui il volume non viene collegato o montato possono essere interrotti. Questo può essere il risultato di una Job Definition configurata in modo errato. Per ulteriori dettagli, consulta Crea una definizione di processo a nodo singolo sulle risorse Amazon EKS.

Verificare che **`aws-auth ConfigMap`** sia configurato correttamente

Per verificare che `aws-auth ConfigMap` sia configurato correttamente:

1. Recupera i ruoli mappati in. `aws-auth ConfigMap`

```
$ kubectl get configmap -n kube-system aws-auth -o yaml
```

2. Verificare che `roleARN` sia configurato come segue.

```
rolearn: arn:aws:iam::aws_account_number:role/AWSServiceRoleForBatch
```

> **Note**
>
> Il percorso `aws-service-role/batch.amazonaws.com/` è stato rimosso dall'ARN del ruolo collegato al servizio. Ciò è dovuto a un problema con la `aws-auth` mappa di configurazione. Per ulteriori informazioni, vedere Ruoli con percorsi non funzionano quando il percorso è incluso nel relativo ARN in. aws-auth configmap

> **Note**
>
> Puoi anche esaminare i log del piano di controllo di Amazon EKS. Per ulteriori informazioni, consulta la registrazione del piano di controllo di Amazon EKS nella Guida per l'utente di Amazon EKS.

Per risolvere un problema a causa del quale un lavoro è bloccato in uno `RUNNABLE` stato, ti consigliamo di utilizzare questa opzione `kubectl` per riapplicare il manifesto. Per ulteriori informazioni, consulta Fase 2: prepara il cluster Amazon EKS per AWS Batch. In alternativa, puoi utilizzare `kubectl` per modificare manualmente il `aws-authConfigMap`. Per ulteriori informazioni, consulta Abilitare l'accesso di utenti e ruoli IAM al cluster nella Amazon EKS User Guide.

Le autorizzazioni o le associazioni RBAC non sono configurate correttamente

Se riscontri problemi relativi alle autorizzazioni RBAC o all'associazione, verifica che il ruolo possa accedere al `aws-batch` Kubernetes namespace: Kubernetes

```
$ kubectl get namespace namespace --as=aws-batch
```

```
$ kubectl auth can-i get ns --as=aws-batch
```

È inoltre possibile utilizzare il **`kubectl describe`** comando per visualizzare le autorizzazioni per un ruolo o un namespace del cluster. Kubernetes

```
$ kubectl describe clusterrole aws-batch-cluster-role
```

Di seguito è riportato un output di esempio.

```
Name:         aws-batch-cluster-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources                                       Non-Resource URLs  Resource Names
 Verbs
  ---------                                       -----------------  --------------
 -----
  configmaps                                      []                 []
 [get list watch]
  nodes                                           []                 []
 [get list watch]
  pods                                            []                 []
 [get list watch]
  daemonsets.apps                                 []                 []
 [get list watch]
  deployments.apps                                []                 []
 [get list watch]
  replicasets.apps                                []                 []
 [get list watch]
  statefulsets.apps                               []                 []
 [get list watch]
  clusterrolebindings.rbac.authorization.k8s.io   []                 []
 [get list]
  clusterroles.rbac.authorization.k8s.io          []                 []
 [get list]
  namespaces                                      []                 []
 [get]
  events                                          []                 []
 [list]
```

```
$ kubectl describe role aws-batch-compute-environment-role -n my-aws-batch-namespace
```

Di seguito è riportato un output di esempio.

```
Name:         aws-batch-compute-environment-role
Labels:       <none>
Annotations:  <none>
PolicyRule:
  Resources                                  Non-Resource URLs  Resource Names  Verbs
  ---------                                  -----------------  --------------  -----
  pods                                       []                 []              [create
 get list watch delete patch]
  serviceaccounts                            []                 []              [get list]
  rolebindings.rbac.authorization.k8s.io     []                 []              [get list]
  roles.rbac.authorization.k8s.io            []                 []              [get list]
```

Per risolvere questo problema, riapplica le autorizzazioni e i comandi RBAC. `rolebinding` Per ulteriori informazioni, consulta Fase 2: prepara il cluster Amazon EKS per AWS Batch.

Risorsa: quote AWS Batch di servizio

La tabella seguente fornisce le quote di servizio AWS Batch che non possono essere modificate. Ogni quota è specifica della regione.

Risorsa	Quota
Numero massimo di code di lavoro Per ulteriori informazioni, consulta Job queues.	50
Numero massimo di ambienti di elaborazione tra Amazon ECS e Amazon EKS. Per ulteriori informazioni, consulta Ambienti di calcolo per AWS Batch.	50
Numero massimo di ambienti di elaborazione per cluster Amazon EKS.	5
Numero massimo di ambienti di elaborazione per ogni coda di lavoro	3
Numero massimo di dipendenze processo per un processo	20
Dimensione massima della definizione del processo (per le operazioni API `RegisterJobDefinition`)	24 KiB
Dimensione massima del payload del processo (per le operazioni API `SubmitJob`)	30 KiB
Dimensione massima dell'array per i processi in array	10000
Numero massimo di processi nello stato `SUBMITTED`	1000000
Numero massimo di transazioni al secondo (TPS) per ogni account per le operazioni `SubmitJob`	50
Numero massimo di risorse consumabili	50.000
Numero massimo di ambienti di servizio. Per ulteriori informazioni, consulta Ambienti di servizio.	50
Numero massimo di ambienti di servizio per ogni coda di lavoro	1
Dimensione massima della richiesta `SubmitServiceJob`	30 KiB

Risorsa	Quota
Dimensione massima del payload della richiesta di servizio di lavoro (per le operazioni `SubmitServiceJob` API)	10 KiB
Numero massimo di transazioni al secondo (TPS) per ogni account per le operazioni `SubmitServiceJob`	5
Numero massimo di tentativi con strategia di riprova per un lavoro di assistenza	10

A seconda dell'utilizzo AWS Batch, potrebbero essere applicate quote aggiuntive. Per ulteriori informazioni sulle EC2 quote Amazon, consulta Amazon EC2 Service Quotas nel. Riferimenti generali di AWS Per ulteriori informazioni sulle quote Amazon ECS, consulta Amazon ECS Service Quotas nel. Riferimenti generali di AWS Per ulteriori informazioni sulle quote Amazon EKS, consulta Amazon EKS Service Quotas nel. Riferimenti generali di AWS

Cronologia dei documenti

La tabella seguente descrive le modifiche importanti alla documentazione rispetto alla versione iniziale di AWS Batch. Inoltre, aggiorniamo frequentemente la documentazione tenendo conto dei feedback ricevuti.

Modifica	Descrizione	Data
Aggiunti default_x86_64 e default_arm64	Aggiunti tipi di istanza nuovi **`default_x86_64`** e **`default_arm64`** consentiti.	15 agosto 2025
Ambienti di servizio e lavori di assistenza aggiunti	Ambienti di servizio e lavori di assistenza aggiunti da utilizzare AWS Batch con l' SageMaker intelligenza artificiale.	30 luglio 2025
Aggiunto AWSService RoleFor AWSBatch WithSagemaker e AWSBatch ServiceRo lePolicyForSageMaker	Aggiunti nuovi ruoli AWS collegati ai servizi `AWSServiceRoleForAWSBatchWithSagemaker` e policy gestite `AWSBatchServiceRolePolicyForSageMaker` che consentono di AWS Batch gestire l' SageMaker IA per tuo conto.	30 luglio 2025
Aggiunge il supporto per EKS AL 2023 AMIs	Come eseguire l'aggiornamento da EKS AL2 a EKS AL2 023	24 giugno 2025
Aggiunge il supporto per un FireLens comando ECS Exec	Aggiunge il supporto per un FireLens comando ECS Exec.	15 aprile 2025

Aggiunge il supporto per la pianificazione basata sulle risorse per AWS Batch	Aggiunge il supporto per la pianificazione basata sulle risorse AWS Batch per Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e. AWS Fargate	27 febbraio 2025
Versioni Amazon EKS AWS Batch supportate aggiornate	Sono state aggiornate le versioni di Amazon EKS che AWS Batch supportano la rimozione della versione 1.22.	11 marzo 2024
Versioni Amazon EKS AWS Batch supportate aggiornate	Sono state aggiornate le versioni di Amazon EKS che AWS Batch supportan o l'inclusione della versione 1.29.	29 febbraio 2024
Ritentativi di lavoro automatiz zati	È stato corretto l'esempio di codice.	29 febbraio 2024
Aggiunge il supporto per i lavori con più contenitori per AWS Batch	Aggiunge il supporto per i lavori multi-container AWS Batch per Amazon Elastic Container Service, Amazon Elastic Kubernetes Service e. AWS Fargate	28 febbraio 2024
Versioni Amazon EKS AWS Batch supportate aggiornate	Sono state aggiornate le versioni di Amazon EKS che AWS Batch supportano l'inclusione della versione 1.28	27 gennaio 2024

Aggiornato BatchServiceRolePolicy e AWSBatchServiceRole	BatchServiceRolePolicy Aggiornato per aggiungere il supporto per la descrizione della cronologia delle richieste e delle Amazon EC2 Auto Scaling attività di Spot Fleet. AWSBatchServiceRole Aggiornato per aggiungere dichiarazioni IDs, concedere AWS Batch autorizzazioni a `ec2:DescribeSpotFleetRequestHistory` e. `autoscaling:DescribeScalingActivities`	5 dicembre 2023
AWS Batch su Amazon EKS	AWS Batch aggiunge il supporto per l'esecuzione di lavori su cluster Amazon EKS.	25 ottobre 2022
Deputati di prevenzione confusa tra diversi servizi per AWS Batch	AWS Batch fornisce ora una soluzione alternativa al confuso problema della sicurezza secondaria, che si presenta quando un'entità (un servizio o un account) è costretta da un'altra entità a compiere un'azione.	6 giugno 2022

Endpoint VPC di interfaccia ()AWS PrivateLink	È stato aggiunto il supporto per la configurazione degli endpoint VPC dell'interfaccia con tecnologia. AWS PrivateLink Ciò significa che puoi creare una connessione privata tra il tuo VPC AWS Batch senza richiedere l'accesso tramite un'istanza NAT, una connessione VPN o. Direct Connect	15 aprile 2022
Aggiornamenti avanzati dell'ambiente di elaborazione	AWS Batch aggiornamenti di supporto avanzati per gli ambienti di elaborazione.	14 aprile 2022
AWS aggiornamenti gestiti delle politiche: aggiornamento delle politiche esistenti	AWS Batch politiche gestite esistenti aggiornate.	6 dicembre 2021
Pianificazione basata sulla condivisione equa	AWS Batch aggiunge il supporto per l'aggiunta di politiche di pianificazione alle code di lavoro.	9 novembre 2021
Amazon EFS	AWS Batch aggiunge il supporto per l'aggiunta di file system Amazon EFS alle definizioni dei processi.	1 aprile 2021
È stato aggiunto un ruolo collegato al servizio	AWS Batch aggiunge il ruolo collegato al AWSServiceRoleForBatchservizio.	10 marzo 2021
AWS Fargate supporto	AWS Batch aggiunge il supporto per l'esecuzione di lavori sulle risorse di Fargate.	3 dicembre 2020

Aggiunta di tag alle risorse	AWS Batch aggiunge il supporto per l'aggiunta di tag di metadati agli ambienti di calcolo, alle definizioni dei processi, alle code di lavoro e ai lavori.	7 ottobre 2020
Segreti	AWS Batch aggiunge il supporto per il trasferimento di segreti ai lavori.	1 ottobre 2020
Registrazione di log	AWS Batch aggiunge il supporto per specificare driver di registro aggiuntivi per i lavori.	1 ottobre 2020
Strategie di allocazione	AWS Batch aggiunge il supporto per più strategie per scegliere i tipi di istanza.	16 ottobre 2019
Supporto EFA	AWS Batch aggiunge il supporto per i dispositivi Elastic Fabric Adapter (EFA).	2 agosto 2019
Pianificazione tramite GPU	AWS Batch aggiunge la pianificazione della GPU. Con questa funzionalità, puoi specificare il numero di richieste per GPUs ogni processo e AWS Batch ridimensionare le istanze di conseguenza.	4 aprile 2019

Lavori paralleli multinodo	AWS Batch aggiunge il supporto per i lavori paralleli multinodo. Puoi utilizzar e questa funzionalità per eseguire singoli job che si estendono su più EC2 istanze Amazon.	19 novembre 2018
Autorizzazioni a livello di risorsa	AWS Batch supporta autorizza zioni a livello di risorsa su diverse operazioni API.	12 novembre 2018
Supporto per modelli Amazon EC2 Launch	AWS Batch aggiunge il supporto per l'utilizzo di modelli di lancio con ambienti di calcolo.	12 novembre 2018
AWS Batch timeout di lavoro	AWS Batch aggiunge il supporto per il timeout del lavoro. Con questo supporto, puoi configurare una durata di timeout specifica per i tuoi lavori in modo che se un lavoro dura più a lungo del dovuto, lo AWS Batch interrompa.	5 Aprile 2018
AWS Batch lavori come obiettivi EventBridge	AWS Batch i posti di lavoro sono resi disponibili come EventBridge obiettivi. Creando regole semplici, puoi abbinare gli eventi e inviare AWS Batch lavori in risposta ad essi.	1 marzo 2018
CloudTrail revisione contabile per AWS Batch	CloudTrail può controllare le chiamate effettuate alle azioni AWS Batch API.	10 gennaio 2018

lavori di array	AWS Batch aggiunge il supporto per i lavori di array. È possibile utilizzare i job di array per lo sweep dei parametri e i carichi di lavoro Monte Carlo.	28 novembre 2017
Etichettatura estesa AWS Batch	AWS Batch espande il supporto per la funzione di etichettatura. Puoi utilizzare questa funzione per specifica re i tag per le istanze Amazon EC2 Spot lanciate all'interno di ambienti di elaborazione gestiti.	26 ottobre 2017
AWS Batch flusso di eventi per EventBridge	AWS Batch aggiunge il flusso di eventi per EventBridge. Puoi utilizzare lo stream di AWS Batch eventi per ricevere notifiche quasi in tempo reale sullo stato dei lavori inviati alle tue code di lavoro.	24 ottobre 2017
Ritentativi di lavoro automatiz zati	AWS Batch aggiunge il supporto per i nuovi tentativi di lavoro. Con questo aggiornam ento, è possibile applicare una strategia di nuovi tentativi ai processi e alle definizioni dei processi che consente di riprovare automaticamente i processi in caso di esito negativo.	28 marzo 2017

AWS Batch disponibilità generale	AWS Batch viene introdotto, progettato come mezzo per eseguire carichi di lavoro di elaborazione in batch su. Cloud AWS	5 gennaio 2017

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.