aws

Guida per l'utente

AWS Batch

AWS Batch: Guida per l'utente

Table of Contents

Cosa è AWS Batch?

AWS Batch ti aiuta a eseguire carichi di lavoro di elaborazione in batch in Cloud AWS. Il Batch computing è un modo comune per sviluppatori, scienziati e ingegneri di accedere a grandi quantità di risorse di elaborazione. AWS Batchelimina il carico indifferenziato di configurazione e 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 Batchfornisce automaticamente le risorse di elaborazione e ottimizza la distribuzione del carico di lavoro in base alla quantità e alla scala dei carichi di lavoro. ConAWS 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.

Argomenti

Componenti di AWS Batch

AWS Batchsemplifica l'esecuzione di processi 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.

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 sulle AWS Fargate nostre risorse Amazon EC2 nel tuo ambiente di calcolo, utilizzando i parametri specificati in una definizione di processo. I processi possono fare riferimento ad altri processi tramite

il nome o l'ID e possono dipendere dal completamento di altri processi. Per ulteriori informazioni, consulta Processi.

Definizioni dei processi

Una definizione di processo specifica come devono essere eseguiti i processi. 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, consultare Definizioni del lavoro

Code di processi

Quando si invia un AWS Batch lavoro, lo si invia a una particolare coda di lavoro, dove il lavoro rimane fino a quando non viene pianificato 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.

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 istanza EC2, 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 vCPU 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 Batchavvia, 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 Ambiente di elaborazione.

Nozioni di base

Inizia a utilizzare AWS Batch creando la definizione di un processo, un ambiente di calcolo e una coda di processi nella console AWS Batch.

La procedura guidata alla AWS Batch prima esecuzione ti offre la possibilità di creare un ambiente di calcolo e una coda di lavoro e inviare un esempio di lavoro Hello World. Se hai già un'immagine Docker da lanciareAWS Batch, puoi creare una definizione di lavoro con quell'immagine e inviarla invece alla coda. Per ulteriori informazioni, consulta Guida introduttiva con AWS Batch.

Dashboard

Nella 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 della dashboard:

- Panoramica dei lavori: per ulteriori informazioni sui AWS Batch lavori, vedereProcessi.
- 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. Ambiente di elaborazione

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

Coda di lavoro singola

Questo widget 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, seleziona Regione AWS quello che desideri.
3. Nel pannello di navigazione seleziona Pannello di controllo.
4. Scegli 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.

CloudWatch Container Insights

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

Per aggiungere questo widget, segui questi passaggi.

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

Job log

Questo widget mostra diversi log dei tuoi lavori in un'unica comoda posizione. Per ulteriori informazioni sui registri dei lavori, consulta. the section called “ Job log”

Per aggiungere questo widget, segui questi passaggi.

1. Apri la AWS Batch console.
2. Dalla barra di navigazione, seleziona Regione AWS quello che desideri.
3. Nel pannello di navigazione seleziona Pannello di controllo.
4. Scegli Aggiungi widget.
5. Per Job logs, scegli Aggiungi widget.
6. Per Job id, inserisci l'ID del lavoro che desideri.

7. Scegli Aggiungi.

Configurazione con AWS Batch

Se ti sei già registrato ad Amazon Web Services (AWS) e utilizzi Amazon Elastic Compute Cloud (Amazon EC2) Elastic 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 console Amazon EC2 per molti di questi passaggi.

Completa le seguenti attività per cui prepararti. AWS Batch Se hai già completato uno di questi passaggi, puoi passare direttamente all' AWS CLI installazione di.

Argomenti

- Iscriviti per un Account AWS
- Crea un utente con accesso amministrativo
- Crea ruoli IAM per i tuoi ambienti di calcolo e le istanze di container
- Creazione di una coppia di chiavi
- Crea un VPC
- Creazione di un gruppo di sicurezza
- Installa il AWS CLI

Iscriviti per un Account AWS

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

Per iscriverti a un Account AWS

1. Apri la pagina all'indirizzo https://portal.aws.amazon.com/billing/signup.

2. Segui le istruzioni online.

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

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

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

Crea un utente con accesso amministrativo

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

Proteggi i tuoi Utente root dell'account AWS

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

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

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

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

Crea un utente con accesso amministrativo

1. Abilita Centro identità IAM.

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

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

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

Accedi come utente con accesso amministrativo

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

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

Assegna l'accesso ad altri utenti

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

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

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

Crea 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 altre AWS API per tuo conto. Crea un ruolo IAM che fornisca queste credenziali ai tuoi ambienti di calcolo e alle istanze di container, quindi associa quel ruolo ai tuoi ambienti di elaborazione.

> **Note**
>
> I ruoli dell'ambiente di AWS Batch calcolo e 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 Ruolo dell'istanza Amazon ECS prima della creazione del primo ambiente di calcolo. Utilizzo di ruoli collegati ai servizi per AWS Batch

Creazione di una coppia di chiavi

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 coppia di key pair, puoi crearne una utilizzando la console Amazon EC2. 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 Guida per l'utente di Amazon EC2.

Per creare una coppia di chiavi

1. Apri la console Amazon EC2 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
```

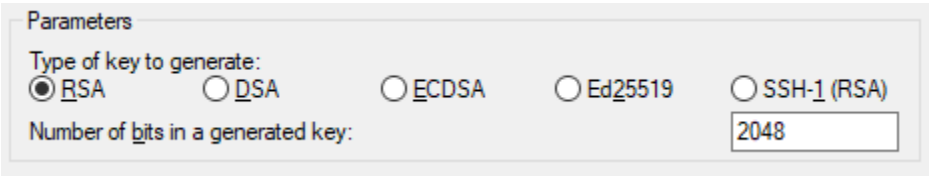
Per ulteriori informazioni, consulta Amazon EC2 Key Pairs nella Amazon EC2 User Guide.

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

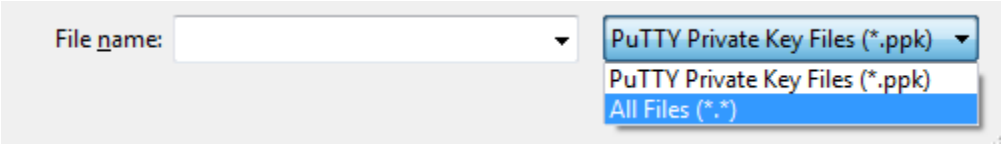
Per connettere l'istanza Linux da un computer che esegue Mac o Linux, specifica il file `.pem` nel client SSH con l'opzione `-i` e il percorso della chiave privata. Per connetterti alla tua istanza Linux da un computer che esegue Windows, usa uno dei due MindTerm o PuTTY. Se si prevede di utilizzare 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. Avviate PuTTYgen (ad esempio, dal menu Start, scegliete Tutti i programmi, PuTTY e PuTTYgen).
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. Scegliere Load (Carica). Di default, PuTTYgen visualizza solo i file con 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. PuTTYgen visualizza un avviso relativo al salvataggio della chiave senza 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.
IPv4 CIDR block (Blocco CIDR IPv4)	Input manuale CIDR IPv4 La dimensione del blocco CIDR deve essere compresa tra /16 e /28.
IPv6 CIDR block (Blocco CIDR IPv6)	Nessun blocco CIDR IPv6

Opzione	Valore	
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 Guida per l'utente di Amazon EC2.

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

Per creare un gruppo di sicurezza tramite console

1. Accedere alla 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 My 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, vedete 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 con AWS Batch

Puoi utilizzare la procedura guidata alla AWS Batch prima esecuzione per iniziare rapidamente a. 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.

Prerequisiti

Assicurati di fare quanto segue prima di avviare la procedura guidata per la AWS Batch prima esecuzione:

- Completare i passaggi descritti in. Configurazione con AWS Batch
- Verifica di disporre Account AWS delle autorizzazioni richieste.

Guida introduttiva - Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2): fornisce capacità di calcolo scalabile e sicura in Cloud AWS. L'utilizzo Amazon EC2 elimina la necessità di investimenti anticipati in hardware e ti permette di sviluppare e distribuire più rapidamente le applicazioni.

Puoi utilizzare Amazon EC2 per avviare il numero di server virtuali necessari, configurare la sicurezza e i servizi di rete, nonché gestire l'archiviazione. Amazon EC2 consente di dimensionarsi verso l'alto o verso il basso per gestire le variazioni a livello di requisiti o i picchi di popolarità, riducendo la necessità di elaborare previsioni relative al traffico.

Crea un ambiente di elaborazione

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

1. Apri la procedura guidata per la prima esecuzione della console.AWS Batch
2. Per Seleziona il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon EC2).
3. Seleziona Avanti.

4. Nella sezione Configurazione dell'ambiente di calcolo per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
5. Per il ruolo Instance, scegli un profilo di istanza esistente a cui siano associate le autorizzazioni IAM richieste. Questo profilo 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.
6. (Facoltativo) Un tag è un'etichetta assegnata a una risorsa. Per aggiungere un tag o un tag Amazon EC2, espandi Tag, quindi scegli Aggiungi tag. Inserisci una coppia chiave-valore, quindi scegli nuovamente Aggiungi tag.

   > **Important**
   >
   > Se scegli Aggiungi tag, devi inserire una coppia chiave-valore e scegliere nuovamente Aggiungi tag o scegliere Rimuovi tag.

7. (Facoltativo) Nella sezione Configurazione dell'istanza per Usa le istanze Spot di Amazon EC2, attiva Abilita l'utilizzo delle istanze Spot.
8. (Solo Spot) Per il prezzo massimo% su richiesta, inserisci la percentuale massima di prezzi on demand che desideri pagare per le risorse Spot.
9. (Facoltativo) (solo Spot) Per il ruolo della flotta Spot, scegli un ruolo IAM della flotta Spot di Amazon EC2 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 spot di Amazon EC2.

   > **Important**
   >
   > Per etichettare le istanze Spot al momento della creazione, il ruolo IAM della flotta Spot di Amazon EC2 deve utilizzare la nuova policy gestita di AmazonEC2. SpotFleetTaggingRole La policy SpotFleetRole gestita di AmazonEC2 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 “Tagging delle risorse ”.

10. Per Minimum vCPU, scegli il numero minimo di vCPU EC2 che il tuo ambiente di elaborazione mantiene, indipendentemente dalla domanda di lavoro in coda.

11. Per le vCPU desiderate, scegli il numero di vCPU EC2 con cui avviare il tuo ambiente di elaborazione. All'aumentare della domanda di job queue, AWS Batch aumenta il numero desiderato di vCPU e vengono aggiunte istanze EC2. Il numero di vCPU può aumentare fino al numero massimo di vCPU. Al diminuire della domanda, AWS Batch diminuisce il numero desiderato di vCPU e rimuove le istanze. Il numero di riduzioni fino al numero minimo di vCPU.
12. Per Maximum vCPUs (vCPU massime), scegli il numero massimo di vCPU EC2 per consentire la scalabilità orizzontale dell'ambiente di calcolo, indipendentemente dalla domanda della coda dei processi.
13. Per i tipi di istanze consentiti, scegli i tipi di istanza Amazon EC2 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`. Puoi anche `optimal` scegliere di selezionare i tipi di istanze (tra le famiglie di R4 istanze C4M4, e) che soddisfano la domanda delle tue code di lavoro.

> **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 ridimensiona le GPU in base alla quantità richiesta nelle code di lavoro. Per utilizzare la pianificazione GPU, l'ambiente di calcolo deve includere tipi di istanze della famiglia`p2`,`p3`, `p4`,`p5`, `g3` or. `g3s` `g4` `g5`

> **Note**
>
> Attualmente, `optimal` utilizza i tipi di istanze delle famiglie di istanze C4M4, eR4.
> In Regioni AWS questo caso non ci sono tipi di istanze di quelle famiglie di istanze, vengono utilizzati tipi di istanze di C5M5, e famiglie di R5 istanze.

14. Espandere Additional configuration (Configurazione aggiuntiva).
15. (Facoltativo) Per Gruppo di collocamento, inserite il nome del gruppo di posizionamento per raggruppare le risorse nell'ambiente di calcolo.
16. (Facoltativo) Per la coppia di chiavi EC2, scegli una coppia di chiavi pubblica e privata come credenziali di sicurezza quando ti connetti all'istanza. Per ulteriori informazioni sulle coppie di chiavi Amazon EC2, consulta Coppie di chiavi Amazon EC2 e istanze Linux.
17. 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 e SPOT_CAPACITY_OPTIMIZED per gli ambienti di calcolo Spot EC2. Per ulteriori informazioni, consulta the section called “Strategie di allocazione”.
18. (Facoltativo) Per la configurazione EC2, scegli Aggiungi configurazione EC2. Scegli i valori di sovrascrittura del tipo di immagine e dell'ID immagine per AWS Batch fornire informazioni su come selezionare Amazon Machine Images (AMI) 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

Può essere usato per famiglie di istanze non GPU e non Graviton. AWS Il supporto standard per le AMI Amazon Linux è terminato. Per ulteriori informazioni, consulta AMI Amazon Linux.

> **Note**
>
> L'AMI che scegli per un ambiente di calcolo deve corrispondere all'architettura dei tipi di istanza che desideri utilizzare per quell'ambiente di calcolo. Ad esempio, se il tuo ambiente di calcolo utilizza tipi di A1 istanze, l'AMI delle risorse di calcolo che scegli deve supportare Arm le istanze. Amazon ECS vende entrambe x86 le Arm versioni 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.

19. (Facoltativo) Per Launch template, seleziona un modello di lancio Amazon EC2 esistente per configurare le tue risorse di calcolo. La versione predefinita del modello viene compilata automaticamente. Per ulteriori informazioni, consulta Supporto modello di avvio.

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

20. (Facoltativo) Per Launch template version (Versione modello di avvio), immettere `$Default`, `$Latest` o un determinato numero di versione da utilizzare.

> **Important**
>
> Dopo la creazione dell'ambiente di calcolo, la versione del modello di avvio utilizzata non viene modificata anche se la `$Latest` versione `$Default` o la versione del modello di avvio viene aggiornata. Per utilizzare una nuova versione del modello di avvio, è necessario innanzitutto creare un nuovo ambiente di calcolo, quindi aggiungere il nuovo ambiente di calcolo alla coda di lavoro esistente. Quindi, rimuovi il vecchio ambiente di calcolo dalla coda dei lavori ed elimina il vecchio ambiente di calcolo.

21. Nella sezione Configurazione di rete:

    a. Per l'ID Virtual Private Cloud (VPC), scegli un Amazon VPC.

    b. Per le sottoreti, sono elencate le tue sottoreti. Account AWS Se desideri creare un set personalizzato di sottoreti, scegli Cancella sottoreti, quindi scegli le sottoreti che desideri.

> **Important**
>
> Le risorse di elaborazione devono comunicare con l'endpoint VPC di Amazon ECS tramite un endpoint VPC o più indirizzi IP pubblici. Per ulteriori informazioni, consulta Endpoint AWS PrivateLinkVPC dell'interfaccia Amazon ECS (). Se la tua istanza non ha un endpoint VPC configurato o un indirizzo IP pubblico, puoi utilizzare la traduzione degli indirizzi di rete (NAT). Per ulteriori informazioni su NAT, consulta Gateway NAT e. Creazione di un cloud privato virtuale

c. Per i gruppi di sicurezza, scegli i gruppi di sicurezza Amazon EC2 che desideri associare all'istanza. Se desideri creare un set personalizzato di gruppi di sicurezza, scegli Cancella gruppi di sicurezza. Quindi, scegli i gruppi di sicurezza che desideri.

22. Seleziona Avanti.

Crea 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 nel tuo ambiente di calcolo. Per ulteriori informazioni, consultare Job queues

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

1. Nella sezione Job queue configuration per Name, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per Priorità, immettete un numero intero compreso tra 0 e 100 per la coda dei lavori.

> **Important**
>
> Ai valori interi più alti viene assegnata una priorità più elevata dallo Scheduler. AWS Batch

3. Seleziona Avanti.

Creazione di una definizione di processo

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. Nella sezione Configurazione generale:

   a. Nella sezione Configurazione generale per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri. Il nome può contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).

   b. (Facoltativo) Per il timeout di esecuzione, immettete il periodo di tempo (in secondi) dopo il quale termina un lavoro incompiuto.

   > **Important**
   >
   > Il timeout minimo è di 60 secondi.

   c. (Facoltativo) Un tag è un'etichetta assegnata a una risorsa. Per aggiungere un tag, espandi Tag, quindi scegli Aggiungi tag. Inserisci una coppia chiave-valore, quindi scegli nuovamente Aggiungi tag.

   > **Important**
   >
   > Se scegli Aggiungi tag, devi inserire una coppia chiave-valore e scegliere nuovamente Aggiungi tag o scegliere Rimuovi tag.

   d. (Facoltativo) Attiva i tag Propagate per propagare i tag all'attività Amazon Elastic Container Service.

2. Nella sezione Configurazione del contenitore:

   a. Per Image, inserisci il nome dell'immagine utilizzata per avviare il contenitore. Per impostazione predefinita, sono disponibili tutte le immagini nel registro Docker Hub. Puoi anche specificare altri repository nel formato repository-url/image:tag. Il parametro può avere una lunghezza massima di 255 caratteri. Il parametro può contenere lettere maiuscole e minuscole, numeri, trattini (-), caratteri di sottolineatura (_), due punti (.), barre (/) e segni

numerici (#). Il parametro viene mappato `Image` nella sezione Crea un contenitore dell'API Docker Remote e il parametro di. `IMAGE`docker run

> **Note**
>
> Dockerl'architettura dell'immagine deve corrispondere all'architettura del processore delle risorse di calcolo su cui è pianificata. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

b. Per Command (Comando) specifica il comando da passare al container. Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> Puoi utilizzare i valori e i segnaposto predefiniti per la sostituzione dei parametri 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. (Facoltativo) Per la configurazione di Job Role, scegli un ruolo IAM con autorizzazioni per le AWS API. 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.

e. (Facoltativo) Puoi aggiungere parametri alla definizione del processo come mappature chiave-valore per sovrascrivere i valori predefiniti della definizione del lavoro. Per aggiungere un parametro:

- Per Parametri, scegliete Aggiungi parametro. Immettete una coppia chiave-valore, quindi scegliete nuovamente Aggiungi parametro.

> **Important**
>
> Se scegli Aggiungi parametro, devi configurare almeno un parametro o scegliere Rimuovi parametro.

f. Nella sezione Configurazione dell'ambiente per le vCPU, specificare il numero di vCPU 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.

g. Per Memoria, specificare il limite rigido (in MiB) di memoria da presentare al contenitore del lavoro. 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.

h. Per Numero di GPU, scegli il numero di GPU da riservare per il contenitore.

i. (Facoltativo) Per la configurazione delle variabili di ambiente, scegli Aggiungi variabili di ambiente per aggiungere variabili di ambiente da passare al contenitore. Questo parametro

è mappato a `Env` nella sezione Create a container di Docker Remote API e l'opzione `--env` a docker run.

j. (Facoltativo) Per Segreti, scegli Aggiungi segreto per aggiungere segreti come coppie nome-valore. Questi segreti sono esposti nel contenitore. Per ulteriori informazioni, vedere SecretOptions inParametri di definizione del lavoro per ContainerProperties.

k. (Facoltativo) Nella sezione di configurazione Linux:

   i. 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.

   ii. Per assegnare al job container autorizzazioni elevate sull'istanza host (simili 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.

   iii. Attiva Enable init process per eseguire un **`init`** processo all'interno del contenitore. Questo processo inoltra segnali e raccoglie processi.

l. (Facoltativo) Nella sezione di configurazione del file system:

   i. Attiva Abilita il filesystem di sola lettura per rimuovere l'accesso in scrittura al volume.

   ii. In Dimensione della memoria condivisa, inserisci la dimensione (in MiB) del `/dev/shm` volume.

   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. Per ulteriori informazioni, vedere swappiness in. Parametri di definizione del lavoro per ContainerProperties

   v. (Facoltativo) Espandi Configurazione aggiuntiva.

   vi. Per Tmpfs, scegli Aggiungi tmpfs per aggiungere un mount. `tmpfs`

   vii. Per 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.

viii. (Facoltativo) Per la configurazione Ulimits, scegli Aggiungi ulimit per aggiungere un valore per il contenitore. `ulimits` Inserisci i valori Name, Soft limit e Hard limit, quindi scegli Aggiungi ulimit.

3. Seleziona Avanti.

Crea un processo.

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 contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Seleziona Avanti.

Rivedi e crea

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

Guida introduttiva - Fargate

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.

Crea un ambiente di elaborazione

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 Seleziona il tipo di orchestrazione, scegli Fargate.

3. Seleziona Avanti.
4. Nella sezione Configurazione dell'ambiente di calcolo per Nome, specifica un nome univoco per il tuo ambiente di calcolo. Il nome può contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
5. (Facoltativo) Un tag è un'etichetta assegnata a una risorsa. Per aggiungere un tag, espandi Tag, quindi scegli Aggiungi tag. Inserisci una coppia chiave-valore, quindi scegli nuovamente Aggiungi tag.

   > **Important**
   >
   > Se scegli Aggiungi tag, devi inserire una coppia chiave-valore e scegliere nuovamente Aggiungi tag o scegliere Rimuovi tag.

6. (Facoltativo) Nella sezione Configurazione dell'istanza per Usa la capacità Spot Fargate, attiva Abilita l'utilizzo delle istanze Spot.
7. Per Numero massimo di vCPU, immettere il numero massimo di vCPU che l'istanza può utilizzare.
8. Nella sezione Configurazione di rete:

   a. Per l'ID Virtual Private Cloud (VPC), scegli un Amazon VPC.

   b. Per le sottoreti, sono elencate le tue sottoreti. Account AWS Se desideri creare un set personalizzato di sottoreti, scegli Cancella sottoreti, quindi scegli le sottoreti che desideri.

      > **Important**
      >
      > Le risorse di elaborazione devono comunicare con l'endpoint VPC di Amazon ECS tramite un endpoint VPC o più indirizzi IP pubblici. Per ulteriori informazioni, consulta Endpoint AWS PrivateLinkVPC dell'interfaccia Amazon ECS (). Se la tua istanza non ha un endpoint VPC configurato o un indirizzo IP pubblico, puoi utilizzare la traduzione degli indirizzi di rete (NAT). Per ulteriori informazioni su NAT, consulta Gateway NAT e. Creazione di un cloud privato virtuale

   c. Per i gruppi di sicurezza, scegli i gruppi di sicurezza Amazon EC2 che desideri associare all'istanza. Se desideri creare un set personalizzato di gruppi di sicurezza, scegli Cancella gruppi di sicurezza. Quindi, scegli i gruppi di sicurezza che desideri.

9. Seleziona Avanti.

Crea 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 nel 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 contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Per Priorità, immettete un numero intero compreso tra 0 e 100 per la coda dei lavori.

   > **Important**
   >
   > Ai valori interi più alti viene assegnata una priorità più elevata dallo Scheduler. AWS Batch

3. Seleziona Avanti.

Creazione di una definizione di processo

Per creare la definizione del processo:

1. Nella sezione Configurazione generale:

   a. In Nome, inserisci un nome di definizione del processo personalizzato.

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

   b. (Facoltativo) Per Timeout di esecuzione, immettete il periodo di tempo (in secondi) dopo il quale termina un lavoro incompiuto.

      > **Important**
      >
      > Il timeout minimo è di 60 secondi.

c. (Facoltativo) Un tag è un'etichetta assegnata a una risorsa. Per aggiungere un tag, espandi Tag, quindi scegli Aggiungi tag. Inserisci una coppia chiave-valore, quindi scegli nuovamente Aggiungi tag.

> **Important**
>
> Se scegli Aggiungi tag, devi inserire una coppia chiave-valore e scegliere nuovamente Aggiungi tag o scegliere Rimuovi tag.

d. (Facoltativo) Attiva i tag Propagate per propagare i tag all'attività Amazon Elastic Container Service.

2. Nella sezione di configurazione della piattaforma Fargate:

a. (Facoltativo) Per la versione della piattaforma Fargate, inserite l'ambiente di runtime specifico che desiderate.

b. Per la piattaforma Runtime, selezionate LINUX o Windows.

c. (Solo Windows) Per la famiglia di sistemi operativi, selezionate un sistema operativo.

d. Per l'architettura della CPU, selezionate l'architettura della CPU desiderata.

e. (Facoltativo) Attiva Assegna IP pubblico per assegnare un indirizzo IP pubblico.

f. Per Archiviazione temporanea, inserisci la quantità di spazio di archiviazione temporanea che desideri.

> **Note**
>
> Per impostazione predefinita, vengono utilizzati 20 GiB di storage temporaneo. Per utilizzare uno storage temporaneo aggiuntivo, immettere un valore compreso tra 21 GiB e 100 GiB.

g. 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. Ad esempio, puoi scegliere ecsTaskExecutionRuolo.

3. Nella sezione Configurazione del contenitore:

a. Per Image, inserisci il nome dell'immagine utilizzata per avviare il contenitore. Per impostazione predefinita, sono disponibili tutte le immagini nel registro Docker Hub. Puoi anche specificare altri repository nel formato repository-url/image:tag. Il parametro può avere

una lunghezza massima di 255 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini bassi (-), caratteri di sottolineatura (_), due punti (:), punti (.), barre (/) e simboli di numero (#). Il parametro è mappato `Image` nella sezione Crea un contenitore dell'API Docker Remote e il `IMAGE` parametro di docker run.

> **Note**
>
> Dockerl'architettura dell'immagine deve corrispondere all'architettura del processore delle risorse di calcolo su cui è pianificata. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

b. Per Command (Comando) specifica il comando da passare al container. Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> Puoi utilizzare i valori e i segnaposto predefiniti per la sostituzione dei parametri nel comando. Per ulteriori informazioni, consulta Parametri.

> **Tip**
>
> Scegli Info per esaminare gli esempi di codice Bash e JSON.

c. (Facoltativo) È possibile aggiungere parametri alla definizione del processo come mappature chiave-valore per sovrascrivere i valori predefiniti della definizione del processo. Per aggiungere un parametro:

   - Per Parametri, scegliete Aggiungi parametro. Immettete una coppia chiave-valore, quindi scegliete nuovamente Aggiungi parametro.

     > **Important**
     >
     > Se scegli Aggiungi parametro, devi configurare almeno un parametro o scegliere Rimuovi parametro.

d. (Facoltativo) Nella sezione Configurazione dell'ambiente per la configurazione del ruolo Job, scegli un ruolo IAM che fornisca l'autorizzazione all'uso delle AWS API.

e. Nella sezione Configurazione dell'ambiente per le vCPU, specificare il numero di vCPU 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.

f. Per Memoria, specificare il limite rigido (in MiB) di memoria da presentare al contenitore del lavoro. 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.

g. (Facoltativo) Per le variabili di ambiente, scegliete Aggiungi variabili di ambiente per aggiungere variabili di ambiente da passare al contenitore. Questo parametro è mappato a `Env` nella sezione Create a container di Docker Remote API e l'opzione `--env` a docker run.

4. Seleziona Avanti.

Crea un processo.

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 contenere fino a 128 caratteri. Deve contenere lettere maiuscole e minuscole, numeri, trattini (-) e caratteri di sottolineatura (_).
2. Seleziona Avanti.

Rivedi e crea

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

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

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 AWS 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.
- Creazione di un cluster Amazon EKS — Per ulteriori informazioni, consulta la sezione Guida introduttiva ad Amazon EKS `eksctl` nella Guida per l'utente di Amazon EKS.

> **Note**
>
> AWS Batch supporta solo cluster Amazon EKS con endpoint server API ad accesso pubblico, accessibili alla rete Internet pubblica. Per impostazione predefinita, gli endpoint del server API dei cluster Amazon EKS hanno accesso pubblico. Per ulteriori informazioni, consulta Amazon EKS Cluster Endpoint Access Control nella Amazon EKS User Guide.

> **Note**
>
> AWS Batch non fornisce l'orchestrazione dei nodi gestiti per CoredNS o altri pod di distribuzione. Se hai bisogno di CoreDNS, consulta Aggiungere il componente aggiuntivo CoreDNS Amazon EKS nella Guida per l'utente di Amazon EKS. Oppure, usa `eksctl create cluster create` per creare il cluster, include CoredNS per impostazione predefinita.

- 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. L'utilizzo AWS Management Console di per creare una risorsa di calcolo utilizzando le risorse Amazon EKS richiede le autorizzazioni per entrambi `eks:DescribeCluster` e. `eks:ListClusters`

Fase 1: Preparazione del 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.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo dell'autorizzazione RBAC, consulta Using RBAC Authorization nella User Guide. Kubernetes

```
$ 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: ["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 AWS Batch la gestione e il ciclo 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-cluster-name \
    --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Output:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account>: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 2: creazione di 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. Puoi utilizzare le istanze EC2 On-Demand o le 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.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<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

- Il `serviceRole` parametro non deve essere specificato, quindi verrà utilizzato il ruolo AWS Batch collegato al servizio. AWS Batch su Amazon EKS supporta solo il ruolo AWS Batch collegato al servizio.
- Le strategie di `SPOT_PRICE_CAPACITY_OPTIMIZED` allocazione sono supportate solo `BEST_FIT_PROGRESSIVE` per gli ambienti di calcolo Amazon EKS. `SPOT_CAPACITY_OPTIMIZED`

  > **Note**
  >
  > Ti consigliamo di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` piuttosto che `SPOT_CAPACITY_OPTIMIZED` nella maggior parte dei casi.

- Per la`instanceRole`, 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.
- Un modo per ottenere sottoreti funzionanti per il `subnets` parametro consiste nell'utilizzare le sottoreti pubbliche dei gruppi di nodi gestiti di Amazon EKS create durante la `eksctl` creazione

di un cluster Amazon EKS. Altrimenti, utilizza sottoreti con un percorso di rete che supporta l'estrazione di immagini.

- Il `securityGroupIds` parametro può utilizzare lo stesso gruppo di sicurezza del cluster Amazon EKS. Questo comando recupera l'ID del gruppo di sicurezza per il cluster.

```
$ eks describe-cluster \
    --name <cluster-name> \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

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

> **Important**
>
> È importante confermare che l'ambiente di calcolo 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.

Fase 3: Creare una coda di lavoro e collegare l'ambiente di calcolo

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

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 4: Creare una definizione di lavoro

```
$ 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

- Sono supportati solo i lavori in un singolo contenitore.
- Esistono considerazioni per i `memory` parametri `cpu` and. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.

Fase 5: Inviare un lavoro

```
$ aws batch submit-job --job-queue My-Eks-JQ1 \
    --job-definition MyJobOnEks_Sleep --job-name My-Eks-Job1
$ aws batch describe-jobs --job <jobId-from-submit-response>
```

Note

- Sono supportati solo i lavori in un singolo contenitore.
- Assicurati di conoscere tutte le considerazioni pertinenti relative ai `memory` parametri `cpu` and. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.
- Per ulteriori informazioni sull'esecuzione di processi sulle risorse di Amazon EKS, consultaOfferte di lavoro Amazon EKS.

(Facoltativo) Invia un lavoro con eccezioni

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

- 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.
- 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.

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. Il supporto per i cluster Amazon EKS privati su AWS Batch è generalmente disponibile in ambito commerciale, Regioni AWS laddove AWS Batch disponibile.

I cluster solo privati di Amazon EKS non dispongono di accesso a Internet in entrata/uscita e dispongono solo di sottoreti private. 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 (uriform resource idetifier) 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 esiste una nuova versione dell'immagine e aggiorna il registro privato fino a una volta ogni 24 ore.

Indice

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. È inoltre necessario creare tutte le risorse necessarie tra cui VPC, sottoreti, tabelle di routing, endpoint VPC e cluster Amazon EKS. È necessario utilizzare il. AWS CLI

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

- **`kubectl`**— Uno strumento da riga di comando per lavorare con Kubernetes i cluster. 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 richieste AWS Identity and Access Management (IAM): 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 AWS 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.
- Creazione di un cluster Amazon EKS — Per ulteriori informazioni, consulta la sezione Guida introduttiva ad Amazon EKS `eksctl` nella Guida per l'utente di Amazon EKS.

  > **Note**
  >
  > AWS Batch non fornisce l'orchestrazione dei nodi gestiti per CoredNS o altri pod di distribuzione. Se hai bisogno di CoreDNS, consulta Aggiungere il componente aggiuntivo CoreDNS Amazon EKS nella Guida per l'utente di Amazon EKS. Oppure, usa `eksctl create cluster create` per creare il cluster, include CoredNS per impostazione predefinita.

- 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. L'utilizzo AWS Management Console di per

creare una risorsa di calcolo utilizzando le risorse Amazon EKS richiede le autorizzazioni per entrambi `eks:DescribeCluster` e. `eks:ListClusters`

- Crea un cluster EKS privato nella regione us-east-1 utilizzando il file di configurazione di esempio. `eksctl`

```
kind: ClusterConfig
apiVersion: eksctl.io/v1alpha5
availabilityZones:
    - us-east-1a
    - us-east-1b
    - us-east-1d
managedNodeGroups:
    privateNetworking: true
privateCluster:
    enabled: true
    skipEndpointCreation: false
```

  Crea le tue risorse usando il comando: `eksctl create cluster -f clusterConfig.yaml`

- I nodi gestiti in batch devono essere distribuiti su sottoreti che dispongono degli endpoint di interfaccia VPC richiesti. Per ulteriori informazioni, consulta Requisiti del cluster privato.

Fase 1: Preparazione del cluster 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 Amazon EKS.

> **Note**
>
> Per ulteriori informazioni sull'utilizzo dell'autorizzazione RBAC, consulta Using RBAC Authorization nella documentazione. Kubernetes

```
$ 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: ["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 AWS Batch la gestione e il ciclo 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-cluster-name \
    --arn "arn:aws:iam::<your-account>:role/AWSServiceRoleForBatch" \
    --username aws-batch
```

Output:

```
2022-10-25 20:19:57 [#]  adding identity "arn:aws:iam::<your-account>: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 2: creazione di 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. Puoi utilizzare le istanze EC2 On-Demand o le 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.

```
$ cat <<EOF > ./batch-eks-compute-environment.json
{
  "computeEnvironmentName": "My-Eks-CE1",
  "type": "MANAGED",
  "state": "ENABLED",
  "eksConfiguration": {
    "eksClusterArn": "arn:aws:eks:<region>:123456789012:cluster/<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-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

- Il `serviceRole` parametro non deve essere specificato, quindi verrà utilizzato il ruolo AWS Batch collegato al servizio. AWS Batch su Amazon EKS supporta solo il ruolo AWS Batch collegato al servizio.
- Le strategie di `SPOT_PRICE_CAPACITY_OPTIMIZED` allocazione sono supportate solo `BEST_FIT_PROGRESSIVE` per gli ambienti di calcolo Amazon EKS. `SPOT_CAPACITY_OPTIMIZED`

  > **Note**
  >
  > Ti consigliamo di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` anziché `SPOT_CAPACITY_OPTIMIZED` n nella maggior parte dei casi.

- Per la`instanceRole`, 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.
- Un modo per ottenere sottoreti funzionanti per il `subnets` parametro consiste nell'utilizzare le sottoreti pubbliche dei gruppi di nodi gestiti di Amazon EKS create durante la `eksctl` creazione di un cluster Amazon EKS. Altrimenti, utilizza sottoreti con un percorso di rete che supporta l'estrazione di immagini.
- Il `securityGroupIds` parametro può utilizzare lo stesso gruppo di sicurezza del cluster Amazon EKS. Questo comando recupera l'ID del gruppo di sicurezza per il cluster.

```
$ eks describe-cluster \
    --name <cluster-name> \
    --query cluster.resourcesVpcConfig.clusterSecurityGroupId
```

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

Important

È importante confermare 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.

Fase 3: Creare una coda di lavoro e collegare l'ambiente di calcolo

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

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 4: Creare una definizione di lavoro

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

```
$ cat <<EOF > ./batch-eks-job-definition.json
```

```
{
  "jobDefinitionName": "MyJobOnEks_Sleep",
  "type": "container",
  "eksProperties": {
    "podProperties": {
      "hostNetwork": true,
      "containers": [
        {
          "image": "account-id.dkr.ecr.region.amazonaws.com/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
```

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.

Fase 5: Inviare un lavoro

```
$ aws batch submit-job - -job-queue My-Eks-JQ1 \
    - -job-definition MyJobOnEks_Sleep - -job-name My-Eks-Job1
$ aws batch describe-jobs - -job <jobId-from-submit-response>
```

Note

- Sono supportati solo i lavori in un singolo contenitore.
- Assicurati di conoscere tutte le considerazioni pertinenti relative ai `memory` parametri `cpu` and. Per ulteriori informazioni, consulta Considerazioni su AWS Batch memoria e vCPU per Amazon EKS.
- Per ulteriori informazioni sull'esecuzione di processi sulle risorse di Amazon EKS, consultaOfferte di lavoro Amazon EKS.

(Facoltativo) Invia un lavoro con eccezioni

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

- 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.

- 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.

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, consulta Come posso risolvere lo stato del pod in Amazon EKS? .

Processi

I lavori sono l'unità di lavoro da cui si inizia. AWS Batch 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 Parametri di definizione del lavoro per ContainerProperties.

È possibile inviare un numero elevato di processi semplici e indipendenti.

Argomenti

Invio di un lavoro

Dopo aver registrato una definizione di lavoro, puoi inviarla come lavoro a una coda di AWS Batch lavoro. È possibile sovrascrivere molti dei parametri specificati nella definizione del processo in fase di esecuzione.

Per inviare un processo

1. Aprire 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 scegliere Jobs (Processi).
4. Scegli Invia nuovo lavoro.
5. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può contenere fino a 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 Creazione di una definizione di processo a nodo singolo .
7. Per Job queue, scegli una coda lavori esistente. Per ulteriori informazioni, consulta Creazione di 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 AWS Batch tentativi 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 (Comando) specifica il comando da passare al container. Per comandi semplici, inserisci il comando come per un prompt dei comandi. Per comandi più complicati, ad esempio con caratteri speciali), utilizzate la sintassi JSON.

    > **Note**
    >
    > Questo parametro non può contenere una stringa vuota.

    b. Per le vCPU, inserire il numero di vCPU 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 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, dai la priorità alla memoria per i lavori di un tipo di istanza specifico. Per ulteriori informazioni, consulta Risorsa di calcoloGestione della memoria.

d. (Facoltativo) Per Numero di GPU, scegli il numero di GPU 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.

Stati del processo

Quando invii 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 inviato alla coda e non 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 status `RUNNABLE` 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. Il tempo necessario per estrarre un'immagine per il lavoro equivale al 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 lavoro per i `SUCCEEDED` lavori viene mantenuto 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 lavoro per i `FAILED` lavori viene mantenuto 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 del contenitore. 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 Tutorial: Utilizzo dell'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, consulta gli UID nella 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 `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, ma è impostata sull'indirizzo IPv4 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'istanza Amazon EC2
- 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 OnExit viene specificato valuate, può contenere fino a 5 strategie di nuovo tentativo. Se OnExit viene specificato valuate, 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 Stati del processo.

> **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 Riprova la strategiaCreazione di una definizione di processo a nodo singolo , Invio di un lavoro e Codici di errore delle attività interrotte.

Dipendenze dal lavoro

Quando invii un AWS Batch lavoro, puoi specificare gli ID del lavoro da cui dipende il lavoro. In questo caso, il pianificatore di AWS Batch garantirà che il processo venga eseguito solo dopo che le dipendenze specificate sono state completate correttamente. 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 Array.

Per inviare un AWS Batch lavoro con dipendenze, vediInvio di un lavoro.

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, è possibile che il completamento di un processo noto richieda solo 15 minuti. 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. Invio di 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 PropertiesOverride parametro eks 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.

Mappa un job in esecuzione su un pod e un nodo

Il `podProperties` contenuto di un processo in esecuzione `podName` e i `nodeName` parametri 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.

Come riportare 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 Array
- Job timeout
- Ritentativi di lavoro automatizzati
- Pianificazione equa delle quote

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
- Per creare un job basato su GPU sulle risorse di Amazon EKS
- Lavori bloccati in uno status `RUNNABLE`

Lavori Array

Un processo in array è un processo che condivide parametri comuni, come ad esempio la definizione del processo, le vCPU e la memoria. Viene eseguito come una raccolta di processi di base correlati ma separati che potrebbero essere distribuiti su più host e potrebbero essere 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.

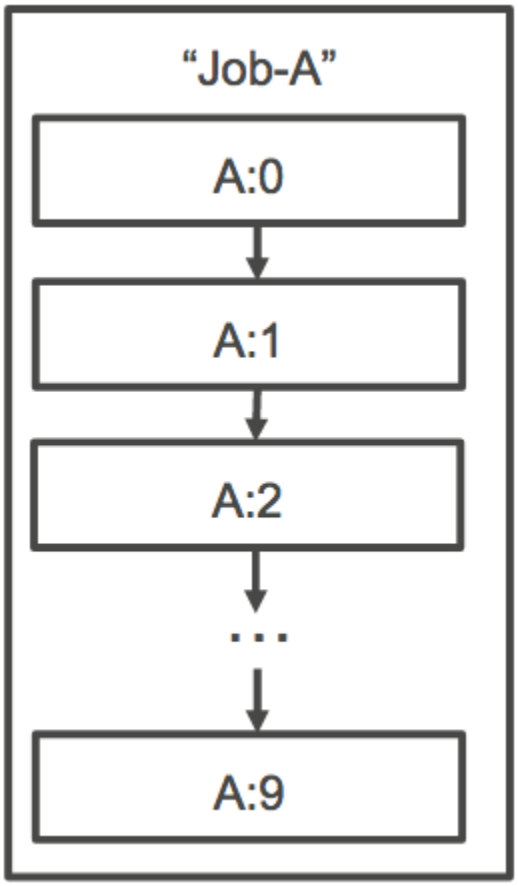
Quando si invia un processo di array, il job dell'array principale ottiene un ID di AWS Batch lavoro normale. Ogni job secondario ha lo stesso ID di base. Tuttavia, l'indice dell'array per il job secondario viene aggiunto alla fine dell'ID principale, ad esempio *`example_job_ID`*`:0` per il primo job 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.

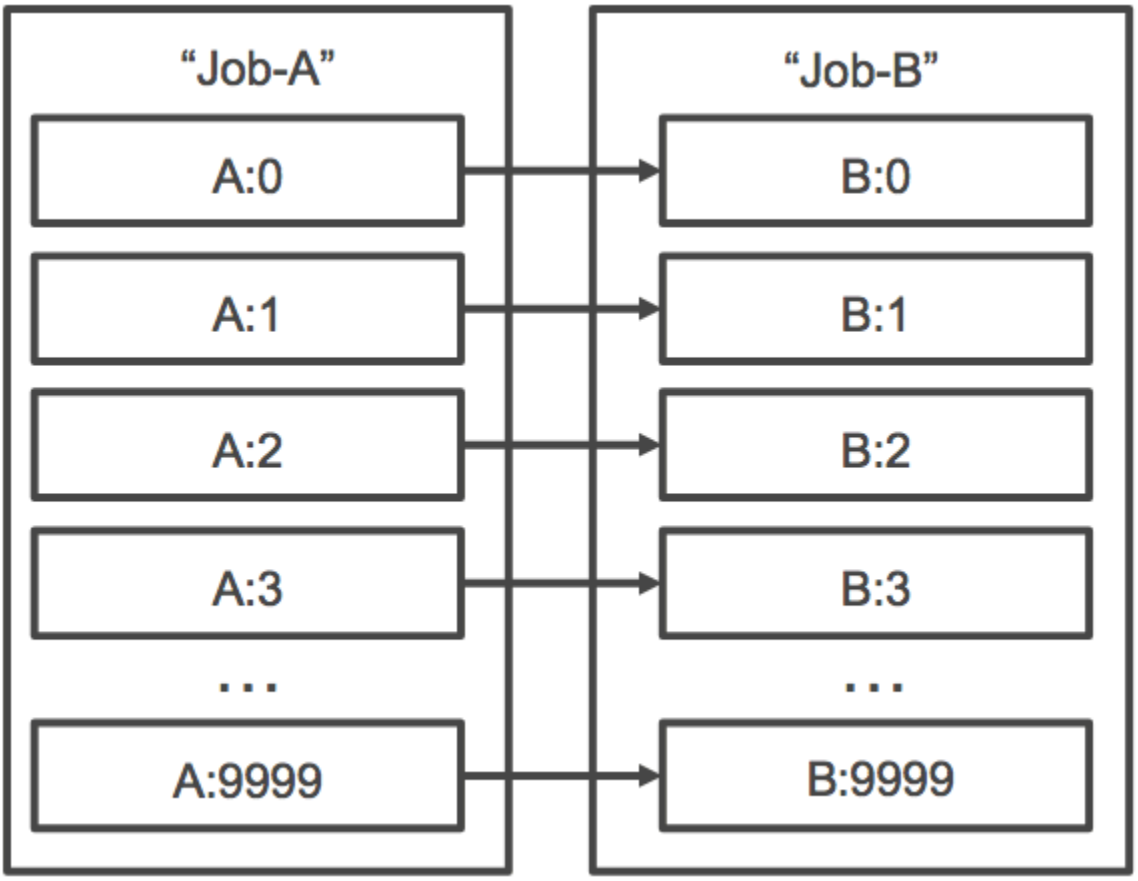
In fase di runtime, la variabile di ambiente `AWS_BATCH_JOB_ARRAY_INDEX` è impostata sul numero di indice dell'array del processo corrispondente del container. Il primo array job index è numerato e `0` i tentativi successivi sono in ordine crescente (ad esempio, 1, 2 e 3). È possibile utilizzare questo valore di indice per controllare il modo in cui vengono indicati i diversi elementi figlio dei processi in array. Per ulteriori informazioni, consulta Tutorial: Utilizzo dell'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 annulli o termini un processo padre in array, tutti i processi figlio verranno annullati o terminati. Puoi annullare o terminare singoli processi figlio (che passeranno allo stato `FAILED`) senza influire sugli altri processi figlio. Tuttavia, se un processo di array secondario fallisce (da solo o annullandolo o terminandolo manualmente), anche il processo principale fallisce.

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.

Per 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://JobD-Output-Bucket",
        "Recipient": "pipeline-notifications@mycompany.com"
    },
    "dependsOn": [
        {
            "jobId": "JobD_job_ID"
        }
    ]
}
```

Tutorial: Utilizzo dell'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.

Prerequisiti

Di seguito sono elencati i requisiti per questo tutorial:

- Un ambiente di AWS Batch elaborazione. Per ulteriori informazioni, consulta Creazione di un ambiente di elaborazione.
- Una coda AWS Batch di lavoro e l'ambiente di elaborazione associato. Per ulteriori informazioni, consulta Creazione di una coda di lavoro.
- È AWS CLI installato sul sistema locale. Per ulteriori informazioni, consulta Installazione dell' AWS Command Line Interface nella Guida per l'utente dell'AWS Command Line Interface .
- Il Docker installato sul sistema locale. Per ulteriori informazioni, consulta la sezione relativa al Docker CE nella documentazione del Docker.

Fase 1: Creare 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.
```

Fase 2: 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. AWS Management Console 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
```

Fase 3: Creare e registrare 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 AWS Management Console. Per ulteriori informazioni, consulta Creazione di 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
```

Fase 4: Inviare 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`.

Fase 5: Visualizzate i registri dei lavori dell'array

Dopo che il lavoro ha raggiunto `SUCCEEDED` lo stato, 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.

Lavori paralleli multinodo

Puoi utilizzare processi paralleli a più nodi per eseguire singoli processi che si estendono su più istanze Amazon EC2. Con i processi paralleli AWS Batch multinodo, 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 processi paralleli a più nodi vengono inviati come un singolo processo. 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. Una volta che il nodo principale è disponibile, i nodi figlio vengono lanciati 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 istanza Amazon EC2 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 lavoro, descrivi *aws_batch_job_id #1 utilizzando* l'operazione API. AWS Batch DescribeJobs 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.

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'indirizzo privato IPv4 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 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 a più nodi devono utilizzare lo stesso tipo di istanza.

Ciclo di vita del lavoro

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

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 Gruppi di collocamento nella Guida per l'utente di Amazon EC2.
- 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. A questa vengono applicati anche tutti i gruppi di sicurezza applicati alle risorse di calcolo. Per ulteriori informazioni, consulta Task Networking with the awsvpc Network Mode nella Amazon Elastic Container Service Developer Guide.
- Il tuo ambiente di elaborazione potrebbe non avere più di cinque gruppi di sicurezza associati.
- 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. Le comunicazioni tra nodi devono utilizzare l'indirizzo IP privato o il nome host DNS per il 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 Creazione di 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.
- Il tuo ambiente di calcolo deve disporre di un numero massimo sufficiente di vCPU per supportare il processo parallelo a più nodi.

- La quota di istanze Amazon EC2 include il numero di istanze necessarie per eseguire il 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 GPU

I job GPU consentono di eseguire processi che utilizzano le GPU di un'istanza.

Sono supportati i seguenti tipi di istanza Amazon EC2 basati su GPU. Per ulteriori informazioni, consulta Istanze Amazon EC2 G3, Istanze Amazon EC2 G4, Istanze Amazon EC2 G5, IstanzeAmazon EC2 P2, Istanze Amazon EC2 P3,Istanze Amazon EC2 P4d e IstanzeAmazon EC2 P5.

Tipo di istanza	GPU	Memoria GPU	vCPU	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
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

Tipo di istanza	GPU	Memoria GPU	vCPU	Memoria	Larghezza di banda di rete
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
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
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	4x100 Gbps
p5.48xlarge	8	640 GiB	192	2 TiB	32x100 Gbps

Note

Per i job GPU in sono supportati solo i tipi di istanze che supportano una GPU NVIDIA e utilizzano un'architettura x86_64. AWS Batch Ad esempio, le famiglie di istanze G4adand G5gnon sono supportate.

Il parametro ResourceRequirements per la definizione del processo specifica il numero di GPU da aggiungere al contenitore. Questo numero di GPU non è disponibile per nessun altro processo eseguito su quell'istanza per la durata di tale processo. Tutti i tipi di istanze in un ambiente di calcolo che esegue processi GPU devono appartenere alle famiglie di istanze p2p3,p4,p5,g3,g3s,g4, og5. Se ciò non viene fatto, un job GPU potrebbe rimanere bloccato nello stato. RUNNABLE

I lavori che non utilizzano le GPU 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.

Per creare un job basato su GPU sulle risorse di Amazon EKS

Questa sezione spiega come eseguire un carico di lavoro di GPU Amazon EKS su. AWS Batch

Indice

Per creare Kubernetes cluster basati 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 istanza con GPU NVIDIA.
- 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
```

Per creare 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 GPU nella documentazione. Kubernetes

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
```

Per eseguire 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"
}
```

Cerca e filtra i lavori AWS Batch

È possibile elencare i lavori in una coda di lavoro utilizzando la AWS Batch console. Tuttavia, se ci sono molti lavori nella coda dei lavori, potrebbe essere difficile trovare un lavoro specifico.

È possibile utilizzare Ricerca e filtraggio per elencare i lavori che corrispondono ai criteri di ricerca specificati.

1. Apri la AWS Batch console.
2. Scegliere Jobs (Processi).
3. Attiva Ricerca e filtraggio.

> **Note**
>
> Se hai diversi lavori, questo processo potrebbe richiedere alcuni minuti.

4. Nella casella Seleziona una coda di lavoro, scegli la coda dei lavori che desideri cercare.
5. Nella casella Filtra risorse per proprietà o valore, scegli una delle proprietà elencate.
6. Scegliete l'operatore che desiderate utilizzare. Ad esempio, scegli Stato =.

> **Tip**
>
> Per utilizzare una proprietà o un operatore diverso, chiudete i criteri correnti. Scegliete quindi la proprietà e l'operatore che desiderate.

7. Immettete o scegliete il valore di una proprietà. Ad esempio, inserisci tutto o parte del nome di un lavoro o scegli Status = RUNNABLE.
8. Scegli il lavoro che desideri nell'elenco filtrato.

> **Tip**
>
> Se non vedi il lavoro che desideri, scorri l'elenco filtrato.

Job log

È possibile configurare i AWS Batch lavori per inviare le informazioni di registro ai CloudWatch registri. In questo modo, puoi visualizzare diversi registri 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).

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 Cerca e filtra i lavori AWS Batch.

4. In Status, scegli lo stato del lavoro che desideri.
5. Scegli il lavoro che desideri.
6. Nella pagina Dettagli, scorri verso il basso fino a Job Logs.
7. Scegli Recupera registri.
8. Per Autorizzazione obbligatoria, inserisci**OK**, quindi scegli Autorizza per accettare gli CloudWatch addebiti 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. Seleziona Salvataggio delle modifiche.

9. Esamina i dati di registro del AWS Batch lavoro.

> **Tip**
>
> È possibile filtrare il registro in base a Parole chiave, Numero massimo di risultati e Ordinamento. Puoi anche scegliere uno degli intervalli di tempo predefiniti o creare un intervallo personalizzato per personalizzare i risultati.

Informazioni sul 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 Ricerca e filtro per trovarli più velocemente. Per ulteriori informazioni, consulta Cerca e filtra i lavori AWS Batch.

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

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 runtime.

Indice

Alcuni degli attributi specificati in una definizione di processo includono:

- Immagine Docker da utilizzare con il container del processo
- Numero di vCPU e quantità di memoria da utilizzare con il container
- Il comando che il container deve eseguire all'avvio
- Variabili di ambiente (se necessarie) da passare al container all'avvio
- Tutti i volumi di dati da utilizzare con il container
- Quale ruolo IAM (se esiste) deve essere utilizzato dal tuo job per le AWS autorizzazioni

Per una descrizione completa dei parametri disponibili in una definizione di processo, consultare Parametri di definizione del lavoro per ContainerProperties.

Creazione di una definizione di processo a nodo singolo

Per poter eseguire un processo in AWS Batch, è necessario creare una definizione del 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.

Puoi creare una definizione di processo parallelo multinodo sulle risorse di Amazon Elastic Container Service. Per ulteriori informazioni, consulta the section called “Creazione di una definizione di processo parallelo a più nodi”.

Argomenti

Creazione di una definizione di processo a nodo singolo sulle risorse Amazon EC2

Per creare una nuova definizione di lavoro sulle risorse Amazon EC2:

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 Job definition.
4. Scegli Crea.
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon EC2).
6. Per la configurazione della piattaforma EC2, disattiva Enable multi-node parallel processing.
7. In Nome, inserisci un nome univoco per la definizione del lavoro. Il nome può contenere fino a 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 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, inserisci il numero di volte in cui 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.
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`*. Il nome può contenere fino a 225 caratteri. Può contenere lettere maiuscole e minuscole, numeri, trattini (-), caratteri di sottolineatura (_), due punti (:), barre (/) e segni numerici (#). Questo parametro è mappato a `Image` nella sezione Crea un container dell'API remota Docker e al parametro `IMAGE` di docker run.

    > **Note**
    >
    > Dockerl'architettura delle immagini deve corrispondere all'architettura del processore delle risorse di elaborazione su cui sono pianificate. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

    - 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

b. Per la sintassi dei comandi, scegli Bash o JSON.

c. Per Command (Comando) specifica il comando da passare al container. Per comandi più semplici, inserite il comando come per un prompt dei comandi. Verificate quindi che il JSON risultato sia corretto e passato a. Docker daemon Per comandi più complicati (ad esempio, con caratteri speciali), utilizzate la sintassi JSON.

> **Tip**
>
> Scegliete Info per visualizzare Bash e JSON codificare gli esempi.

Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

d. (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.

e. Per la configurazione di Job Role, scegli un ruolo IAM con autorizzazioni per le AWS API. 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 le vCPU, inserire il numero di vCPU 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 Risorsa di calcoloGestione della memoria.

    c. Per Numero di GPU, scegli il numero di GPU da 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, vedere SecretOptions inParametri di definizione del lavoro per ContainerProperties.

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 processi.

20. (Facoltativo) Nella sezione di configurazione del file system:

    a. Attiva Abilita il 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, vedere swappiness in. Parametri di definizione del lavoro per ContainerProperties

    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. (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, consulta LogDriver inParametri di definizione del lavoro per ContainerProperties.

> **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, consulta SecretOptions in. Parametri di definizione del lavoro per ContainerProperties

22. Scegli Pagina successiva.

23. 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.

Creazione di una definizione di lavoro a nodo singolo sulle risorse AWS Fargate

Per creare una nuova definizione di lavoro sulle AWS Fargate risorse:

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 Job definition.
4. Scegli Crea.
5. Per il tipo di orchestrazione, scegli Fargate. Per ulteriori informazioni, consulta AWS Batchsu AWS Fargate.
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. 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 Image, scegli l'immagine Docker 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 avere una lunghezza massima di 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**
>
> Dockerl'architettura dell'immagine deve corrispondere all'architettura del processore delle risorse di elaborazione su cui è pianificata. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

b. Per la sintassi dei comandi, scegli Bash o JSON.

c. Per Command (Comando) specifica il comando da passare al container. Per i comandi semplici, inserite il comando come per un prompt dei comandi, quindi verificate che il JSON risultato sia corretto. Viene passato al Docker demone. Per comandi più complicati (ad esempio, con caratteri speciali), usa la sintassi JSON.

> **Tip**
>
> Scegliete Info per visualizzare Bash e JSON codificare gli esempi.

Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

> **Note**
>
> È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

d. (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

e. Nella sezione Configurazione dell'ambiente:

   i. Per la configurazione del ruolo Job, scegli un ruolo IAM con autorizzazioni per le AWS API. 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 le vCPU, inserire il numero di vCPU 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 Risorsa di calcoloGestione della memoria.

f. (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.

g. (Facoltativo) Per Segreti, scegliete Aggiungi segreto per aggiungere segreti come coppie nome-valore. Questi segreti sono esposti nel contenitore. Per ulteriori informazioni, vedere SecretOptions inParametri di definizione del lavoro per ContainerProperties.

h. 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 di inizializzazione per eseguire un processo di inizializzazione all'interno del contenitore. Questo processo inoltra segnali e raccoglie processi.

c. Attiva Abilita il 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 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, consulta LogDriver inParametri di definizione del lavoro per ContainerProperties.

> **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, consulta SecretOptions in. Parametri di definizione del lavoro per ContainerProperties

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.

Creazione di una definizione di processo a nodo singolo sulle risorse Amazon EKS

Per creare una nuova definizione di lavoro sulle risorse di Amazon Elastic Kubernetes Service:

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 Job definition.
4. Scegli 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 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
    - Predefinito: pod eredita la configurazione di risoluzione dei nomi dal nodo su cui viene eseguita.

> **Note**
>
> Se non viene specificata una politica DNS, quella predefinita non è la politica DNS predefinita. ClusterFirstViene invece utilizzato.

- 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 le etichette dei 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/`

e. Scegli Pagina successiva.

f. 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**
>
> Dockerl'architettura dell'immagine deve corrispondere all'architettura del processore delle risorse di calcolo su cui è pianificata. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

iii. (Facoltativo) Per la policy Image pull, scegli quando estrarre le immagini.

iv. (Facoltativo) In Comando, inserite un JSON comando Bash or 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.

g. (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

h. Nella sezione Configurazione dell'ambiente:

i. Per le vCPU, inserire il numero di vCPU 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 Risorsa di calcoloGestione della memoria.

i. (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.

j. (Opzionale) 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.

iii. Scegliete Sola lettura per rimuovere le autorizzazioni di scrittura sul volume.

iv. Scegli Aggiungi supporto per volume.

k. (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.

l. (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.

m. (Facoltativo) Per assegnare al contenitore del lavoro autorizzazioni elevate sull'istanza host (analogamente all'`root`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.

n. (Facoltativo) Attiva il filesystem root di sola lettura per rimuovere l'accesso in scrittura al filesystem root.

o. (Facoltativo) Attiva Esegui come utente non root t per eseguire i contenitori in come utente non root. pod

> **Note**
>
> Se l'opzione Run as non-root è attivata, kubelet convalida l'immagine in fase di esecuzione per verificare che non venga eseguita come UID 0.

p. 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.

Creazione di una definizione di processo parallelo a più nodi

Per poter eseguire un processo in AWS Batch, è necessario creare una definizione del processo. Questo processo varia leggermente tra processi paralleli a nodo singolo e multinodo. In questo argomento viene illustrato in particolare come creare una definizione di processo per un processo parallelo a AWS Batch più nodi. Per ulteriori informazioni, consulta Lavori paralleli multinodo.

> **Note**
>
> AWSFargate non supporta i lavori paralleli multinodo.

Creazione di una definizione di processo parallelo a più nodi sulle risorse Amazon EC2

Per creare una definizione di processo parallelo a nodo singolo, consulta Creazione di una definizione di processo a nodo singolo .

Per creare una definizione di processo parallelo multinodo sulle risorse di Amazon Elastic Compute Cloud:

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nella barra di navigazione, seleziona la Regione AWS da utilizzare.
3. Nel riquadro di navigazione, scegli Job definition.
4. Scegli Crea.
5. Per il tipo di orchestrazione, scegli Amazon Elastic Compute Cloud (Amazon EC2).
6. Per Abilita il 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 segni numerici (#). Questo parametro è mappato a `Image` nella sezione Crea un container dell'API remota Docker e al parametro `IMAGE` di docker run.

    > **Note**
    >
    > Dockerl'architettura delle immagini deve corrispondere all'architettura del processore delle risorse di elaborazione su cui sono pianificate. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `quay.io/assemblyline/ubuntu`).

d. Per la sintassi dei comandi, scegli Bash o JSON.

e. Per Command (Comando) specifica il comando da passare al container. Per i comandi semplici, è possibile immettere il comando come si fa al prompt dei comandi nella scheda delimitata da spazi. Verificate quindi che il JSON risultato sia corretto. Il risultato JSON viene passato a. Docker daemon Per comandi più complessi (ad esempio con caratteri speciali), puoi passare alla scheda JSON e immettere qui l'array di stringhe equivalente.

   Questo parametro è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

   > **Note**
   >
   > È possibile utilizzare valori predefiniti per la sostituzione dei parametri e i segnaposto nel comando. Per ulteriori informazioni, consulta Parametri.

f. Per vCPUs (vCPU) specifica il numero di vCPU da prenotare per il container. 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.

g. 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 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, è possibile fornire ai job quanta più memoria possibile per un particolare tipo di istanza. Per ulteriori informazioni, consulta Risorsa di calcoloGestione della memoria.

h. (Facoltativo) Per Numero di GPU, specifica il numero di GPU utilizzate dal job. Il processo viene eseguito su un contenitore con il numero specificato di GPU collegate a quel contenitore.

i. (Facoltativo) Per Job role, puoi specificare un ruolo IAM che fornisca al contenitore del tuo job le autorizzazioni per utilizzare le AWS API. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, inclusi 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.

j. (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 le AWS API. Questa funzionalità utilizza i ruoli IAM di Amazon ECS per la funzionalità delle attività. Per ulteriori informazioni, inclusi 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, vedere SecretOptions inParametri di definizione del lavoro per ContainerProperties.

21. Nella sezione di configurazione Linux:

    a. Attiva Abilita il 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, vedere swappiness in. Parametri di definizione del lavoro per ContainerProperties

    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. (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, consulta LogDriver inParametri di definizione del lavoro per ContainerProperties.

    > **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, consulta SecretOptions in. Parametri di definizione del lavoro per ContainerProperties

27. Scegli Pagina successiva.
28. 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.

Creazione di definizioni di lavoro utilizzando 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 Parametri di definizione del lavoro per ContainerProperties.

```
{
    "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
                    }
                }
            ]
        }
    }
}
```

> **Note**
>
> È possibile generare un modello di definizione del processo a contenitore singolo con il seguente comando: AWS CLI
>
> ```
> $ aws batch register-job-definition --generate-cli-skeleton
> ```

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 lavoro
- 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 lavoro 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 lavoro

Indice

- Propaga i tag
- Proprietà del nodo
- Riprova la strategia
- Priorità di pianificazione
- Tag
- Timeout

Nome della definizione del Job

`jobDefinitionName`

Quando si registra una definizione di processo, è necessario specificare un nome, Il nome può contenere fino a 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

Type

`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 the section called “Creazione di una definizione di processo parallelo a più nodi”.

-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 è mappato a `Cmd` nella sezione Crea un container dell'API remota Docker e al parametro `COMMAND` di docker run. Per ulteriori informazioni sul parametro Docker `CMD`, consulta https://docs.docker.com/engine/reference/builder/#cmd.

```
"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 esecuzione (ruolo IAM).

- Tipo: stringa

Campo obbligatorio: no

`fargatePlatformConfiguration`

La configurazione della piattaforma per i lavori eseguiti sulle risorse di Fargate. I processi in esecuzione su risorse EC2 non devono specificare questo parametro.

Tipo: oggetto FargatePlatformdi configurazione

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**
>
> Dockerl'architettura dell'immagine deve corrispondere all'architettura del processore delle risorse di elaborazione su cui è pianificata. Ad esempio, Docker le immagini Arm basate possono essere eseguite solo su risorse di elaborazione Arm basate.

- 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 archivi online vengono ulteriormente qualificate da un nome di dominio (ad esempi, `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 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 in esecuzione su risorse 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 in esecuzione su risorse 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 in esecuzione su risorse 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**
  >
  > Le AMI ottimizzate per Amazon ECS 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 memoria per funzionare come spazio di swap in un'istanza Amazon EC2 utilizzando un file di swap? .

- I parametri dello spazio di swap sono supportati solo per le definizioni dei processi che utilizzano risorse EC2.
- 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 in esecuzione su risorse 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 in esecuzione su risorse 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 Utilizzo del driver di log awslogs il driver di registrazione Amazon CloudWatch Logs nella documentazione 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 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 Specifica di 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.

Ad esempio`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 processi in esecuzione su risorse EC2 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. Questa operazione è necessaria se il lavoro 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 in esecuzione su risorse 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 GPU fisici da prenotare per il container. Il numero di GPU riservate per tutti i contenitori di un processo non può superare il numero di GPU disponibili sulla risorsa di elaborazione su cui viene avviato il processo.

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. Risorsa di calcoloGestione della memoria

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 vCPU prenotate per il processo. 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 risorse EC2, 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 del conteggio risorse vCPU on demand di Fargate è 6 vCPU. 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 Specifica di 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 vCPU prenotate per il container.

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 in esecuzione su risorse 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 Utilizzo dei 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 Risorsa di calcoloGestione della memoria.

cpu

Il numero di CPU prenotate per il container. 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 GPU prenotate per il container. 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. Risorsa di calcoloGestione della memoria

cpu

Il numero di vCPU prenotate per il container. 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 GPU prenotate per il container. 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 Kubernetes documentazione.

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 `dnsPolicy` nell'operazione RegisterJobDefinition API, non viene restituito alcun valore `dnsPolicy` dalle DescribeJobdefinizioni o dalle operazioni

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 Dir EksEmpty

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

Limite di dimensione

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

Speciifica 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 Kubernetes documentazione.

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 lavoro. 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 quando l'attività viene creata. 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 dei nodi 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 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.

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

Riprova la strategia

`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 del `ExitCode` valore 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 processi in coda di lavoro con una policy di ripartizione equa. 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 Tagging delle risorse AWS Batch.

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

Creazione di definizioni di lavoro utilizzando EcsProperties

Utilizzando le definizioni dei AWS Batch processi `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 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 alle API AWS Batch

Di seguito vengono descritte ulteriormente alcune delle principali differenze nell'utilizzo dei tipi di dati `EcsProperties` e delle `EcsProperties` API:

- Molti dei parametri utilizzati all'interno `ContainerProperties` appaiono all'interno. `TaskContainerProperties` Alcuni esempi includonocommand,`image`, `privilegedsecrets`, `eusers`. 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, vedere EcsTaskDetailse 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 EcsPropertiesla Guida di riferimento dell'AWS Batch API.
- `taskProperties`contiene le proprietà definite per il task Amazon ECS. Per ulteriori informazioni, consulta EcsPropertiesla Guida di riferimento delle AWS Batch API.
- `containers`include informazioni simili a quelle contenute `containerProperties` nella definizione del contenitore singolo. La differenza principale è che `containers` consente di definire fino a dieci contenitori. Per ulteriori informazioni, consulta ECS:Containers TaskProperties nella Guida di AWS Batch riferimento delle API.
- `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 pod possono condividere lo stesso spazio dei nomi del processo. I valori predefiniti sono. `false` Questa impostazione consente `true` ai contenitori di visualizzare e segnalare i processi in altri contenitori che si trovano nello stesso contenitore.

- Ogni contenitore è importante. Tutti i contenitori devono essere completati correttamente (uscire come 0) affinché il lavoro abbia successo. 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.

AWS Batch scenari di lavoro utilizzando 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 `RegisterJobDefinition`payload. 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 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": true,
              "command": [
                "echo",
                "hello world"
              ],
              "image": "public.ecr.aws/amazonlinux/amazonlinux:latest",
              "resourceRequirements": [
                {
                  "type": "VCPU",
                  "value": "6"
                },
                {
                  "type": "MEMORY",
                  "value": "12288"
                }
              ]
            }
          ]
        }
      ]
  }
}
```

AWS Batch lavoro per Amazon ECS 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 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 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"
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
```

```
  }
}
```

Utilizzo del driver di log 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 `STDOUT` flussi di `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.

Opzioni disponibili per il driver di log awslogs

Il driver di `awslogs` registro supporta le seguenti opzioni nelle definizioni dei processi. AWS Batch 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 del 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, vedere 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.

Specificare 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 Creazione di una definizione di processo a nodo singolo .

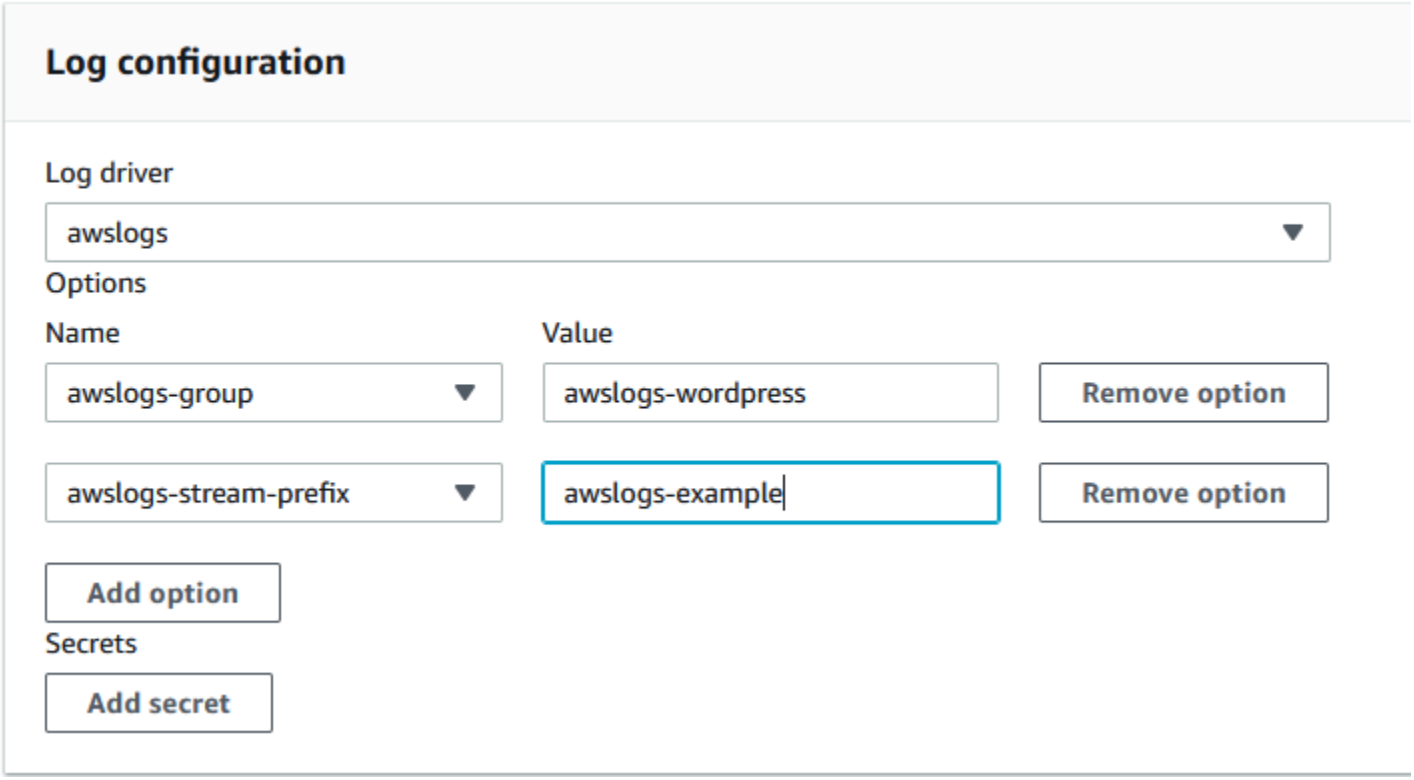
I seguenti frammenti JSON di configurazione del registro hanno un `logConfiguration` oggetto specificato per ogni processo. Uno è per un WordPress processo che invia i log a un gruppo di log chiamato `awslogs-wordpress` e un altro è per un contenitore MySQL che invia i log a un gruppo di log chiamato. `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 Invio di un lavoro.

Specifica di dati sensibili

ConAWS Batch, puoi inserire dati sensibili nei tuoi lavori archiviandoli in AWS Secrets Manager segreti o AWS Systems Manager parametri 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

Specifica di dati sensibili tramite Secrets Manager

Con AWS Batch, puoi inserire dati sensibili nei tuoi lavori archiviandoli in modo AWS Secrets Manager segreto 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 AWS Secrets Manager nella Guida per l'utente di AWS Secrets Manager .

Considerazioni sulla specifica di dati sensibili con 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 AWS Secrets Manager .
- 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 esecuzione (ruolo 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 ulteriori informazioni, consulta Aggiungere e rimuovere le politiche IAM nella Guida per l'utente 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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
```

```
    }
  ]
}
```

Inserimento di dati sensibili come una variabile di ambiente

All'interno della definizione del lavoro, puoi specificare i seguenti elementi:

- 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, vedere Termini e concetti chiave per AWS Secrets Manager nella Guida per l'AWS Secrets Manager 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.

Gli ID di versione vengono utilizzati per tenere traccia di diverse versioni di un segreto quando vengono aggiornati o ruotati. Ogni versione di un segreto ha un ID. Per ulteriori informazioni, vedere Termini e concetti chiave per AWS Secrets Manager nella Guida per l'AWS Secrets Manager 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-valuecomando 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 output di esempio da un comando describe-secret che visualizza il contenuto non crittografato di un segreto insieme ai metadati per tutte le versioni del segreto.

```
{
    "ARN": "arn:aws:secretsmanager:region:aws_account_id:secret:appauthexample-AbCdEf",
    "Name": "appauthexample",
    "Description": "Example of a secret containing application authorization data.",
    "RotationEnabled": false,
    "LastChangedDate": 1581968848.926,
    "LastAccessedDate": 1581897600.0,
    "Tags": [],
    "VersionIdsToStages": {
        "871d9eca-18aa-46a9-8785-981dd39ab30c": [
            "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"
    }]
  }]
}
```

Inserimento di dati sensibili in una configurazione di log

All'interno della definizione del processo, quando si specifica a `logConfiguration` 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"
      }]
    }]
  }]
```

```
}
```

Creare un segreto AWS Secrets Manager

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 AWS Secrets Manager .

Come creare un segreto di base

Utilizza Secrets Manager per creare un segreto per i dati sensibili.

1. Apri la console di 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. Scegli la chiave di AWS KMS crittografia che desideri 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. Seleziona Avanti.
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 AWS Secrets Manager Secrets.
9. Verifica le tue impostazioni e poi scegli Archivia segreto per salvare tutti i dati immessi come nuovo segreto in Secrets Manager.

Specifica dei dati sensibili tramite l'archivio parametri di Systems Manager

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 per la specifica di dati sensibili tramite l'archivio parametri di Systems Manager
- Autorizzazioni IAM richieste per i segreti AWS Batch
- Inserimento di dati sensibili come una variabile di ambiente
- Inserimento di dati sensibili in una configurazione di log
- Creazione di un parametro Parameter Store AWS Systems Manager

Considerazioni per la specifica di dati sensibili tramite l'archivio parametri di Systems Manager

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 esecuzione (ruolo 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 ulteriori informazioni, consulta Aggiungere e rimuovere le politiche IAM nella Guida per l'utente 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:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ssm:GetParameters",
        "secretsmanager:GetSecretValue",
        "kms:Decrypt"
      ],
      "Resource": [
        "arn:aws:ssm:<region>:<aws_account_id>:parameter/<parameter_name>",
        "arn:aws:secretsmanager:<region>:<aws_account_id>:secret:<secret_name>",
```

```
        "arn:aws:kms:<region>:<aws_account_id>:key/<key_id>"
      ]
    }
  ]
}
```

Inserimento di dati sensibili come una 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"
    }]
  }]
}
```

Inserimento di dati sensibili in una configurazione di log

Nella definizione del container, quando specifichi un `logConfiguration` è possibile specificare `secretOptions` con il nome dell'opzione del driver di log per impostare il container e l'ARN completo del parametro dell'archivio parametri di Systems Manager contenente i dati sensibili da presentare al container.

> **Important**
>
> Se il parametro 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"
      }]
    }]
  }]
}
```

Creazione di 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 o SecureString. StringList

> **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 AWS Secrets Manager consente di archiviare le credenziali in modo sicuro e quindi di farvi riferimento nella definizione del lavoro. 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 istanze Amazon EC2 e Fargate.

Important

Se la definizione del lavoro 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 istanze Amazon EC2, 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 AWS Management Console, 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 esecuzione (ruolo 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "secretsmanager:GetSecretValue"
      ],
      "Resource": [
        "arn:aws:secretsmanager:region:123456789012:secret:secret_name",
        "arn:aws:kms:region:123456789012:key/key_id"
      ]
    }
  ]
}
```

Uso dell'autenticazione dei registri privati

Come creare un segreto di base

AWS Secrets Manager Utilizzatelo per creare un segreto per le credenziali del registro privato.

1. Apri la AWS Secrets Manager 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. Seleziona Avanti.
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 AWS Secrets Manager 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 conAWS Batch.

Considerazioni sui volumi Amazon EFS

Quando usi i volumi Amazon EFS, tieni presente le considerazioni seguenti:

- Per i lavori che utilizzano risorse EC2, il supporto del file system Amazon EFS è stato aggiunto come anteprima pubblica con la versione AMI ottimizzata per Amazon ECS `20191212` con agente container versione 1.35.0. Tuttavia, il supporto del file system 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.

Utilizzo dei 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 Batchhost 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 Batchhost 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 sull'esecuzione dei ruoli IAM, consultaAWS Batch esecuzione (ruolo IAM).

Definizioni di lavoro di esempio

Le seguenti definizioni dei processi di esempio illustrano come utilizzare modelli comuni, quali variabili di ambiente, sostituzione di parametri e montaggi di volume.

Usa 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 specifico è disponibile nel post del blog sul calcolo dedicato alla creazione di un processo AWS Batch "Fetch & Run" semplice. 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://my-batch-scripts/myjob.sh"
            },
            {
                "name": "BATCH_FILE_TYPE",
                "value": "script"
            }
        ],
        "user": "nobody"
    }
}
```

Utilizzo della 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 Utilizzo di 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"
        }
      }
    ]
  }
}
```

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 Spot Amazon EC2 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

Creazione di una coda di lavoro

Prima di inviare processi in AWS Batch, è necessario creare una coda di processi. Quando si crea una coda di lavoro, si associano uno o più ambienti di calcolo 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.

Creazione di una coda di lavoro Fargate

Per creare una coda di lavoro a Fargate

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nella barra di navigazione, seleziona la Regione AWS da utilizzare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli 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 Batchnon 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.

Creazione di una coda di lavoro Amazon EC2

Per creare una coda di lavoro Amazon EC2

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Nella barra di navigazione, seleziona la Regione AWS da utilizzare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli 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 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. 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 Batchnon 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.

Creazione di una coda di lavoro 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. Nella barra di navigazione, seleziona la Regione AWS da utilizzare.
3. Nel riquadro di navigazione, scegli Job queues.
4. Scegli 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 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 Batchnon supporta la combinazione di modelli di provisioning in un'unica 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.

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.

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.

```
{
   "computeEnvironmentOrder": [
      {
         "computeEnvironment": "",
         "order": 0
      }
   ],
   "jobQueueName": "",
   "jobStateTimeLimitActions": [
      {
         "state": "RUNNABLE",
         "action": "CANCEL",
   "maxTimeSeconds": 0,
         "reason": ""

      }
   ],
   "priority": 0,
   "schedulingPolicyArn": "",
   "state": "ENABLED",
   "tags": {
      "KeyName": ""
   }
}
```

Note

È possibile generare il modello di coda dei lavori precedente con il seguente AWS CLI comando.

```
$ aws batch create-job-queue --generate-cli-skeleton
```

Parametri della coda Job

Le code di lavoro sono suddivise in quattro componenti di base: nome, stato, priorità e ordine dell'ambiente di calcolo. Questa sezione illustra i componenti associati a questi componenti.

Argomenti

Nome della coda Job

`jobQueueName`

Il nome della coda di processi. Può essere costituito da un massimo di 128 lettere (maiuscole e minuscole), numeri e caratteri di sottolineatura.

Tipo: stringa

Campo obbligatorio: sì

Azioni relative al limite temporale dello stato della coda Job

`jobStateTimeLimitActions`

L'insieme di azioni AWS Batch eseguite sui lavori che rimangono in testa alla coda dei lavori nello stato specificato più a lungo dei tempi specificati. AWS Batch eseguirà ogni azione dopo che `maxTimeSeconds` è passata. (Nota: il valore minimo per `maxTimeSeconds` è 600 (10 minuti) e il valore massimo è 86.400 (24 ore).)

Tipo: matrice di oggetti `JobStateTimeLimitActions`

Campo obbligatorio: no

Priority (Priorità)

priority

Priorità della coda di processi. Le code di processi con priorità più elevata (o un valore intero maggiore per il parametro `priority`) vengono valutate per prime quando associate allo stesso ambiente di calcolo. La priorità è 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. Tutti gli ambienti di elaborazione devono essere Amazon EC2 `EC2` (`SPOTo`) o `FARGATE` Fargate (o). `FARGATE_SPOT` Gli ambienti di elaborazione Amazon EC2 e Fargate non possono essere combinati.

Tipo: integer

Campo obbligatorio: sì

Politica di pianificazione

schedulingPolicyArn

L'Amazon Resource Name (ARN) della politica di pianificazione per la coda dei lavori. Le code di lavoro che non dispongono di un criterio di pianificazione vengono pianificate secondo un modello FIFO (first-in, first-out). Dopo che una coda di lavoro ha una politica di pianificazione, può essere sostituita ma non può essere rimossa. Una coda di lavoro senza un criterio di pianificazione viene pianificata come coda di lavoro FIFO e non può essere aggiunta una politica di pianificazione. Le code di lavoro con una politica di pianificazione possono avere un massimo di 500 identificatori di fair share attivi. Una volta raggiunto il limite, gli invii di tutti i lavori che aggiungono un nuovo identificatore di fair share hanno esito negativo.

Tipo: string

Campo obbligatorio: no

Stato

`state`

Stato della coda di processi. Se lo stato della coda dei lavori è `ENABLED` (il valore predefinito), può accettare lavori. Se lo stato della coda dei processi è `DISABLED`, non è possibile aggiungere nuovi processi alla coda, ma è possibile concludere i processi già presenti nella coda.

Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Ordine dell'ambiente di calcolo

`computeEnvironmentOrder`

Il set di ambienti di calcolo mappati a una coda di processi e l'ordine relativo di ognuno rispetto agli altri. Il pianificatore di processi usa questo parametro per determinare quale ambiente di calcolo deve eseguire un determinato processo. Gli ambienti di calcolo devono trovarsi nello stato `VALID` per poter essere associati a una coda di processi. Puoi associare fino a tre ambienti di calcolo a una coda dei processi. Tutti gli ambienti di elaborazione devono essere Amazon EC2 `EC2` (`SPOTo`) o `FARGATE` Fargate (o). `FARGATE_SPOT` Gli ambienti di elaborazione Amazon EC2 e Fargate non possono essere combinati.

> **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.

Tipo: matrice di oggetti ComputeEnvironmentOrder

Campo obbligatorio: sì

`computeEnvironment`

Amazon Resource Name (ARN) dell'ambiente di calcolo.

Tipo: stringa

Campo obbligatorio: sì

`order`

Ordine dell'ambiente di calcolo. Gli ambienti di calcolo vengono scelti in ordine crescente. Se, ad esempio, due ambienti di calcolo sono associati a una coda di processi, l'ambiente di calcolo con un valore intero `order` inferiore viene scelto per primo per provare a collocare il processo.

Tag

`tags`

Tag di coppia chiave-valore da associare alla coda dei lavori. Per ulteriori informazioni, consulta Tagging delle risorse AWS Batch.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Visualizzazione dello stato della coda dei lavori

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.

Visualizzazione delle informazioni sulla coda dei lavori

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, utilizzate 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 le code di lavoro AWS Batch Fair Share Scheduling (FSS), l'ordine dei lavori si basa sulla priorità del lavoro e sull'utilizzo della condivisione.

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.

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

Job scheduling

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 della 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 Politiche di pianificazione.

Condividi gli identificatori

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 fair share 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 Fair Share senza sostituirlo.

> **Note**
>
> Gli identificatori di condivisione sono univoci all'interno di una coda di lavoro e non vengono aggregati tra le code di lavoro.

È possibile impostare la priorità di pianificazione 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, 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.

> **Note**
>
> Le risorse di elaborazione allegate vengono allocate equamente tra tutti gli identificatori di condivisione, a meno che non vengano esplicitamente sovrascritte.

Pianificazione equa delle quote

La pianificazione della condivisione equa fornisce una serie di controlli per aiutare a pianificare i lavori.

> **Note**
>
> Per ulteriori informazioni sui parametri delle politiche di pianificazione, vedere. Parametri della politica di pianificazione

- Secondi di decadimento delle quote: il periodo di tempo (in secondi) utilizzato dallo AWS Batch scheduler per calcolare una percentuale di fair share per ogni identificatore di quota equa. 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* ActiveFairShares dov'è il numero di identificatori di fair share 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, vedere Visualizzazione dello stato della coda dei job.

Ambiente di elaborazione

Le code dei processi sono mappate a uno o più ambienti di calcolo. 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 definite al momento della creazione dell'ambiente di calcolo. Puoi scegliere di utilizzare le istanze On-Demand di Amazon EC2 e le istanze Spot di Amazon EC2. 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.

Gli ambienti di elaborazione gestiti avviano le istanze Amazon EC2 nel VPC e nelle sottoreti specificate, quindi le registrano in un cluster Amazon ECS. Le istanze Amazon EC2 richiedono l'accesso alla rete esterna per comunicare con l'endpoint del servizio Amazon ECS. Alcune sottoreti non forniscono alle istanze Amazon EC2 indirizzi IP pubblici. Se le tue istanze Amazon EC2 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, consulta. Creazione di 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 AMI per risorse di calcolo.

> **Note**
>
> AWS Batch non aggiorna automaticamente le AMI in un ambiente di calcolo dopo la creazione. Ad esempio, non aggiorna le AMI nel tuo ambiente 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 Aggiornamento degli ambienti di elaborazione. Per utilizzare l'aggiornamento avanzato degli ambienti di calcolo per aggiornare le AMI, segui queste regole:

- O 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 disponibile`$Default`) o aggiungendo una nuova versione al modello di lancio (se lo è`$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, consulta Specifiche AMI delle risorse di calcolo e Creazione di una risorsa di calcolo AMI.

> **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 di Amazon EC2. Sostituire *ecsClusterArn*con l'ARN del cluster ottenuto con il comando precedente.

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

AMI per risorse di calcolo

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 istanze Amazon EC2 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 EC2 supportati

> **Note**
>
> Dopo la creazione di un ambiente di calcolo, AWS Batch non aggiorna le AMI nell'ambiente di calcolo. AWS Batch inoltre, non aggiorna le AMI 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 Aggiornamento degli ambienti di elaborazione. Per utilizzare l'aggiornamento avanzato degli ambienti di calcolo per aggiornare le AMI, segui queste regole:

- O 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 AMI delle risorse di calcolo

- Creazione di una risorsa di calcolo AMI
- Utilizzo di un'AMI per carichi di lavoro GPU
- Deprecazione di Amazon Linux

Specifiche 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 tali AMI. Per ulteriori informazioni, consulta l'AMI ottimizzata per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Creazione di una risorsa di calcolo AMI

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 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 che scegli per un ambiente di calcolo deve corrispondere all'architettura dei tipi di istanza che desideri utilizzare per quell'ambiente di calcolo. Ad esempio, se il tuo

ambiente di calcolo utilizza tipi di A1 istanze, l'AMI delle risorse di calcolo che scegli deve supportare Arm le istanze. Amazon ECS vende entrambe x86 le Arm versioni 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'istanza Amazon Linux 2 EC2 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 Utilizzo di un'AMI per carichi di lavoro GPU.

Important

Puoi scegliere un'AMI di base che non supporti il `ecs-init` pacchetto. Tuttavia, in tal caso, è necessario 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 Guida per l'utente di Amazon EC2.
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 guida per l'utente di Amazon EC2.

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 nell'ID immagine sovrascrivi la casella quando crei l'ambiente di AWS Batch calcolo. Per ulteriori informazioni, vederethe section called “Per creare un ambiente di elaborazione gestito utilizzando le risorse EC2”.

   > **Note**
   >
   > L'AMI che scegli per un ambiente di calcolo deve corrispondere all'architettura dei tipi di istanza che desideri utilizzare per quell'ambiente di calcolo. Ad esempio, se il tuo ambiente di calcolo utilizza tipi di A1 istanze, l'AMI delle risorse di calcolo che scegli deve supportare Arm le istanze. Amazon ECS vende entrambe x86 le Arm versioni 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 Creazione di una 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.

3. (Facoltativo) Invia un processo di esempio alla nuova coda di processi. Per ulteriori informazioni, consulta Definizioni di lavoro di esempio, Creazione di una definizione di processo a nodo singolo e Invio di un lavoro.

Utilizzo di un'AMI per carichi di lavoro GPU

Per eseguire i carichi di lavoro della GPU sulle risorse di elaborazione AWS Batch, è necessario utilizzare un'AMI con il supporto per GPU. Per ulteriori informazioni, consulta Working with GPU on Amazon ECS e AMI ottimizzate per Amazon ECS nella Amazon Elastic Container Service Developer Guide.

Negli ambienti di elaborazione gestiti, se l'ambiente di calcolo specifica tipi o famiglie di istanze `p2`,`p3`,`p4`,`p5`, `g3` `g3sg4`, o tipi di `g5` 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 GetParametersByPath operations per recuperare GetParameters i metadati per le AMI ottimizzate per GPU Amazon ECS consigliate. AWS Command Line Interface

> **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 includere`p5`, 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 GPU on Amazon ECS nella 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

Supporto modello di avvio

AWS Batch supporta l'utilizzo di modelli di lancio di Amazon EC2 con i tuoi ambienti di calcolo EC2. Con i modelli di avvio, puoi modificare la configurazione predefinita delle tue risorse di AWS Batch calcolo senza dover creare AMI personalizzate.

> **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 console Amazon EC2. 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 tag Amazon EC2) 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 tag Amazon EC2 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 chiavi Amazon EC2
- ID AMI Amazon EC2
- ID gruppo di sicurezza
- Tag Amazon EC2

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 Aggiornamento degli ambienti di elaborazione.

Dati utente di Amazon EC2 nei modelli di lancio

Puoi fornire i dati utente di Amazon EC2 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 utente di Amazon EC2 nei modelli di avvio 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.

Di seguito sono riportati alcuni esempi di file MIME multiparte che è possibile utilizzare per crearne di personalizzati.

> **Note**
>
> Se aggiungi dati utente a un modello di lancio nella console Amazon EC2, puoi incollarli come testo semplice. In alternativa, 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..."
>     }
> }
> ```

Esempi

- Esempio: monta un file system Amazon EFS esistente
- Esempio: sovrascrivi la configurazione predefinita dell'agente container Amazon ECS
- Esempio: montaggio di 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: montaggio di un file system Amazon FSx for Lustre esistente

Example

Questo file MIME multiparte di esempio 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 Guida per l'utente di Amazon FSx for Lustre. 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"
        }
    ],
}
```

Creazione di 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 le istanze Amazon EC2 AWS o le risorse 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 dell'istanza Amazon EC2 all'interno dell'ambiente.

Important

Le istanze Fargate Spot non sono supportate nei seguenti scenari:

- Su contenitori Amazon Linux con architettura ARM64.
- 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.

Indice

Per creare un ambiente di elaborazione gestito utilizzando le risorse AWS Fargate

1. Aprire 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 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 Autorizzazioni di ruolo collegate al servizio 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. Selezionare 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 EC2 Spot e Fargate_spot.

   b. Per Maximum vCPU, scegli il numero massimo di vCPU verso 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.

Per creare un ambiente di elaborazione gestito utilizzando le risorse EC2

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 Crea.
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. (Facoltativo) 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 Autorizzazioni di ruolo collegate al servizio 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.
    g. (Facoltativo) Per i tag EC2, 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. Selezionare Aggiungi tag.
    h. (Facoltativo) Per Tag, scegli Aggiungi tag. Quindi, inserisci un nome chiave e un valore opzionale. Selezionare Aggiungi tag.

       Per ulteriori informazioni, consulta Tagging delle risorse AWS Batch.
    i. Scegli Pagina successiva.
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 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.

c. (Solo Spot) Per il ruolo della flotta Spot, scegli un ruolo IAM della flotta Spot di Amazon EC2 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 spot di Amazon EC2.

> **Important**
>
> Per etichettare le istanze Spot al momento della creazione, il ruolo IAM della flotta Spot di Amazon EC2 deve utilizzare la nuova policy gestita di AmazonEC2. SpotFleet TaggingRole La policy gestita dai SpotFleetruoli di AmazonEC2 non dispone delle autorizzazioni necessarie per etichettare le istanze Spot. Per ulteriori informazioni, consulta Istanze Spot non taggate al momento della creazione e the section called “Tagging delle risorse ”.

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

e. Per le vCPU desiderate, scegli il numero di vCPU con cui avviare il tuo ambiente di elaborazione. Se la domanda della tua coda dei processi aumenta, AWS Batch è in grado di incrementare il numero desiderato di vCPU nel tuo ambiente di calcolo e aggiungere istanze EC2, fino al numero massimo di vCPU. Se la domanda diminuisce, AWS Batch è in grado di ridurre il numero desiderato di vCPU nel tuo ambiente di calcolo e rimuovere istanze, fino al numero minimo di vCPU.

f. Per Maximum vCPU, scegli il numero massimo di vCPU verso 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 istanza Amazon EC2 che possono essere avviati. Puoi specificare famiglie di istanze per avviare qualsiasi tipo di istanza all'interno di tali famiglie (ad esempio`c5`,`c5n`, o`p3`). 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`. Puoi anche `optimal` scegliere di selezionare i tipi di istanze (tra le famiglie di R4 istanze C4M4, e) che soddisfano la domanda delle tue code di lavoro.

> **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 scalerà le GPU 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 famiglie`p2`,,,`p3`,`p4`, `p5` `g3`, `g3s` or. `g4` `g5`

> **Note**
>
> Attualmente, `optimal` utilizza i tipi di istanze delle famiglie di istanze C4M4, eR4. In Regioni AWS questo caso non ci sono tipi di istanze di quelle famiglie di istanze, vengono utilizzati tipi di istanze di C5M5, e famiglie di R5 istanze.

h. Espandere Additional configuration (Configurazione aggiuntiva).

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

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

k. 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".

l. (Facoltativo) Per la configurazione EC2, scegli i valori di sovrascrittura del tipo di immagine e dell'ID immagine per AWS Batch fornire informazioni su come selezionare Amazon Machine Images (AMI) 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

Può essere usato per famiglie di istanze non GPU e non Graviton. AWS Il supporto standard per le AMI Amazon Linux è terminato. Per ulteriori informazioni, consulta AMI Amazon Linux.

> **Note**
>
> L'AMI che scegli per un ambiente di calcolo deve corrispondere all'architettura dei tipi di istanza che desideri utilizzare per quell'ambiente di calcolo. Ad esempio, se il tuo ambiente di calcolo utilizza tipi di A1 istanze, l'AMI delle risorse di calcolo che scegli deve supportare Arm le istanze. Amazon ECS vende entrambe x86 le

Arm versioni 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.

m. (Facoltativo) Per Launch template, seleziona un modello di lancio Amazon EC2 esistente per configurare le tue risorse di calcolo. La versione predefinita del modello viene compilata automaticamente. Per ulteriori informazioni, consulta Supporto modello di avvio.

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

n. (Facoltativo) Per Launch template version (Versione modello di avvio), immettere `$Default`, `$Latest` o un determinato numero di versione da utilizzare.

> **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 “Aggiornamento dell'ID AMI”.

o. Scegli Pagina successiva.

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 AWS Outposts nella Amazon Elastic Container Service 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.

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.

Per creare un ambiente di elaborazione non gestito utilizzando le risorse EC2

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. 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. (Facoltativo) Per il ruolo di servizio, scegli un ruolo che consenta al AWS Batch servizio di effettuare chiamate alle operazioni AWS API richieste per tuo conto. Ad esempio, scegli AWSBatchServiceRole. Per ulteriori informazioni, consulta the section called “Utilizzo di ruoli collegati ai servizi”..
7. Per Maximum vCPU, scegli il numero massimo di vCPU verso 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. Selezionare Aggiungi tag. Per ulteriori informazioni, consulta Tagging delle risorse AWS Batch.
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.

Per creare un ambiente di elaborazione gestito utilizzando le risorse Amazon EKS

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 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 Spot EC2, attiva Abilita l'utilizzo delle istanze Spot per utilizzare le istanze Spot di Amazon EC2.
13. (Solo Spot) Per ottenere 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 Spot di Amazon EC2 per l'ambiente di `SPOT` calcolo.

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

15. (Facoltativo) Per Minimum vCPU, scegli il numero minimo di vCPU che il tuo ambiente di elaborazione mantiene, indipendentemente dalla domanda di lavoro in coda.
16. (Facoltativo) Per Maximum vCPU, scegli il numero massimo di vCPU verso cui il tuo ambiente di elaborazione può scalare orizzontalmente, indipendentemente dalla domanda di lavoro in coda.
17. Per i tipi di istanze consentiti, scegli i tipi di istanza Amazon EC2 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`. Puoi anche `optimal` scegliere di selezionare i tipi di istanze (tra le famiglie di R4 istanze C4M4, e) perché hai bisogno che corrispondano alla domanda delle tue code di lavoro.

> **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 ridimensiona le GPU in base alla quantità richiesta nelle code di lavoro. Per utilizzare la pianificazione GPU, l'ambiente di calcolo deve includere tipi di istanze appartenenti alle famiglie`p2`,,,,`p3`, `p4p5`, `g3` o. `g3s` `g4` `g5`

> **Note**
>
> Attualmente, `optimal` utilizza i tipi di istanza delle famiglie di istanze C4, M4 e R4. In Regioni AWS questo caso non sono presenti tipi di istanze di tali famiglie di istanze, vengono utilizzati tipi di istanze di C5M5, e famiglie di R5 istanze.

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 (AMI), scegli Aggiungi configurazione Amazon Machine Images (amis). Quindi, scegli un tipo di immagine, inserisci una sovrascrittura dell'ID dell'immagine e la versione. Kubernetes

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

> **Note**
>
> 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.
>
> Amazon Linux 2
>
> È predefinita per tutte le famiglie di istanze AWS basate su Graviton (ad esempio, `C6g` `M6gR6g`, `eT4g`) e può essere utilizzata per tutti i tipi di istanze non GPU.
>
> Amazon Linux 2 (GPU)
>
> È predefinita per tutte le famiglie di istanze GPU (ad esempio, `P4` and`G4`) e può essere utilizzata per tutti i tipi di istanze non basati su Graviton. AWS

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

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

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.

Modello di ambiente di calcolo

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

```
{
    "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": ""
    }
}
```

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

Parametri dell'ambiente di calcolo

Gli ambienti di calcolo sono suddivisi in diversi componenti di base: il nome, il tipo e lo stato dell'ambiente di calcolo, la definizione delle risorse di calcolo (se si tratta di un ambiente di calcolo gestito), la configurazione di Amazon EKS (se utilizza risorse Amazon EKS), il ruolo di servizio da utilizzare per fornire le autorizzazioni IAM e i tag per AWS Batch l'ambiente di calcolo.

Argomenti

- Tag

Nome dell'ambiente di calcolo

`computeEnvironmentName`

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

Tipo: stringa

Campo obbligatorio: sì

Type

`type`

Il tipo di ambiente di calcolo. Scegli `MANAGED` di AWS Batch gestire le risorse di calcolo EC2 o Fargate che definisci. Per ulteriori informazioni, consulta Risorse di calcolo. Scegli `UNMANAGED` di gestire le tue risorse di calcolo EC2.

-Tipo: stringa

Valori validi: `MANAGED` | `UNMANAGED`

Campo obbligatorio: sì

Stato

`state`

Lo stato dell'ambiente di calcolo.

Se lo stato è`ENABLED`, lo AWS Batch scheduler tenta di collocare i lavori all'interno dell'ambiente. Questi lavori provengono da una coda di lavoro associata sulle risorse di elaborazione.
Se l'ambiente di elaborazione è gestito, le istanze vengono scalate orizzontalmente o automaticamente in base alla domanda di lavoro in coda.

Se lo stato è`DISABLED`, lo AWS Batch scheduler non tenta di inserire lavori all'interno dell'ambiente. I lavori che si trovano in uno `RUNNING` stato `STARTING` o continuano a progredire

normalmente. Gli ambienti di elaborazione gestiti che si trovano nello `DISABLED` stato non sono scalabili orizzontalmente.

> **Note**
>
> Gli ambienti di elaborazione in uno `DISABLED` stato potrebbero continuare a comportare costi di fatturazione. Per evitare costi aggiuntivi, disattiva e quindi elimina l'ambiente di elaborazione. Per ulteriori informazioni, consulta DeleteComputeEnvironmentle sezioni AWS Batch API Reference e Evitare addebiti imprevisti nella Guida per l'AWS Billing utente.

Quando un'istanza è inattiva, viene ridimensionata fino al `minvCpus` valore. Tuttavia, la dimensione dell'istanza non cambia. Ad esempio, considerate un'`c5.8xlarge`istanza con un `minvCpus` valore di `4` e un `desiredvCpus` valore di`36`. Questa istanza non si riduce a un'`c5.large`istanza.

-Tipo: stringa

Valori validi: `ENABLED` | `DISABLED`

Campo obbligatorio: no

Risorse di calcolo

`computeResources`

Dettagli delle risorse di calcolo gestite dall'ambiente di calcolo. Per ulteriori informazioni, consulta Ambiente di elaborazione.

Tipo: oggetto ComputeResource

Obbligatorio: questo parametro è necessario per gli ambienti di elaborazione gestiti

`type`

Tipo di ambiente di calcolo. Puoi scegliere di utilizzare le istanze On-Demand EC2 (`EC2`) e le istanze Spot EC2 () oppure di utilizzare la capacità Fargate (`SPOT`) e la capacità Fargate Spot (`FARGATE`) nel tuo ambiente di elaborazione gestito. `FARGATE_SPOT` Se scegli l'opzione

`SPOT`, devi specificare un ruolo del parco istanze Spot di Amazon EC2 con il parametro `spotIamFleetRole`. Per ulteriori informazioni, consulta Ruolo della flotta spot di Amazon EC2.

Valori validi: `EC2`| `SPOT`| `FARGATE`| `FARGATE_SPOT`

Campo obbligatorio: sì

`allocationStrategy`

La strategia di allocazione da utilizzare per la risorsa di calcolo se non è possibile allocare un numero sufficiente di istanze del tipo di istanza EC2 più adatto. Ciò potrebbe essere dovuto alla disponibilità del tipo di istanza nei limiti del Regione AWS servizio Amazon EC2. Per ulteriori informazioni, consulta Strategie di allocazione.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

`BEST_FIT` (predefinito)

AWS Batch seleziona il tipo di istanza che meglio si adatta alle esigenze dei lavori 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 stai raggiungendo i limiti del servizio Amazon EC2, i lavori aggiuntivi vengono eseguiti solo dopo il 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. Le risorse di calcolo che utilizzano una strategia di `BEST_FIT` allocazione non supportano gli aggiornamenti dell'infrastruttura e non possono aggiornare alcuni parametri. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **Note**
>
> `BEST_FIT`non è supportato per gli ambienti di elaborazione che utilizzano risorse Amazon EKS.

`BEST_FIT_PROGRESSIVE`

Utilizza tipi di istanze aggiuntivi sufficientemente grandi da soddisfare i requisiti dei lavori in coda. Preferisci 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.

`SPOT_CAPACITY_OPTIMIZED`

(Disponibile solo per le risorse di calcolo delle istanze Spot) Utilizza tipi di istanze aggiuntivi sufficientemente grandi da soddisfare i requisiti dei lavori in coda. Preferisci i tipi di istanza che hanno meno probabilità di essere interrotte.

`SPOT_PRICE_CAPACITY_OPTIMIZED`

(Disponibile solo per le risorse di calcolo delle istanze Spot) La strategia di allocazione ottimizzata per prezzo e capacità tiene conto sia del prezzo che della capacità per selezionare i pool di istanze Spot che hanno meno probabilità di subire interruzioni e hanno il prezzo più basso possibile.

> **Note**
>
> Ti consigliamo di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` piuttosto che `SPOT_CAPACITY_OPTIMIZED` nella maggior parte dei casi.

Con `BEST_FIT_PROGRESSIVE`, `SPOT_CAPACITY_OPTIMIZED`, e `SPOT_PRICE_CAPACITY_OPTIMIZED` le strategie che utilizzano istanze On-Demand o Spot e la `BEST_FIT` strategia che utilizza istanze Spot, AWS Batch potrebbe essere necessario superare i requisiti `maxvCpus` di capacità. In questo caso, AWS Batch non supera mai di più `maxvCpus` di una singola istanza.

Valori validi: `BEST_FIT`| `BEST_FIT_PROGRESSIVE`| `SPOT_CAPACITY_OPTIMIZED`| `SPOT_PRICE_CAPACITY_OPTIMIZED`

Campo obbligatorio: no

`minvCpus`

Il numero minimo di vCPU che un ambiente mantiene anche se un ambiente di elaborazione lo è. `DISABLED`

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

Tipo: integer

Campo obbligatorio: no

`maxvCpus`

Il numero massimo di vCPU che l'ambiente di elaborazione può AWS Batch supportare.

> **Note**
>
> Per soddisfare i requisiti di capacità `BEST_FIT_PROGRESSIVESPOT_CAPACITY_OPTIMIZED`, AWS Batch potrebbe essere necessario superare le strategie di `SPOT_PRICE_CAPACITY_OPTIMIZED` allocazione che utilizzano istanze On-Demand o Spot e la `BEST_FIT` strategia che utilizza istanze Spot. `maxvCpus` In questo caso, AWS Batch non supera mai di più di una singola `maxvCpus` istanza. Ad esempio, non AWS Batch utilizza più di una singola istanza tra quelle specificate nell'ambiente di calcolo.

Tipo: integer

Campo obbligatorio: no

`desiredvCpus`

Il numero desiderato di vCPU nell'ambiente di elaborazione. AWS Batch modifica questo valore tra i valori minimo e massimo in base alla domanda della coda di lavoro.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

Tipo: integer

Campo obbligatorio: no

`instanceTypes`

I tipi di istanza che possono essere avviati. Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate. Non specificarlo. È possibile 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`). Nota che i tipi di istanze in metallo non rientrano nelle famiglie di istanze (ad esempio `c5` non include)`c5.metal`. È inoltre possibile scegliere `optimal` per selezionare i tipi di istanza (dalle famiglie di istanze C4, M4 e R4) che corrispondono alla domanda delle code dei processi.

> **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**
>
> Attualmente, `optimal` utilizza i tipi di istanza delle famiglie di istanze C4, M4 e R4. Poiché non Regioni AWS esistono tipi di istanze appartenenti a quelle famiglie di istanze, vengono utilizzati tipi di istanze delle famiglie di istanze C5, M5 e R5.

Tipo: matrice di stringhe

Obbligatorio: sì

`imageId`

Questo parametro è obsoleto.

L'ID dell'AMI (Amazon Machine Image) utilizzato per istanze avviate nell'ambiente di calcolo. Questo parametro viene sovrascritto dal membro `imageId0verride` della struttura `Ec2Configuration`.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

> **Note**
>
> L'AMI che scegli per un ambiente di calcolo deve corrispondere all'architettura dei tipi di istanza che desideri utilizzare per quell'ambiente di calcolo. Ad esempio, se il tuo ambiente di calcolo utilizza tipi di A1 istanze, l'AMI delle risorse di calcolo che scegli deve supportare Arm le istanze. Amazon ECS vende entrambe x86 le Arm versioni 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.

- Tipo: stringa

Campo obbligatorio: no

`subnets`

Le sottoreti del VPC in cui vengono avviate le risorse di calcolo. Queste sottoreti devono trovarsi all'interno dello stesso VPC. Le risorse di calcolo Fargate possono contenere un massimo di 16 sottoreti. Per ulteriori informazioni, consulta VPC e sottoreti nella Guida per l'utente di Amazon VPC.

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

Quando si aggiornano gli ambienti di calcolo, se si fornisce un elenco vuoto di sottoreti VPC, il comportamento risultante differisce tra le risorse di calcolo Fargate ed EC2. Per le risorse di calcolo Fargate, se si fornisce un elenco vuoto non viene apportata alcuna modifica. Per le risorse di calcolo EC2, fornendo un elenco vuoto si rimuovono le sottoreti VPC dalla risorsa di calcolo. Se si modificano le sottoreti VPC, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Questo vale sia per le risorse di calcolo Fargate che per quelle di EC2. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

Tipo: matrice di stringhe

Campo obbligatorio: sì

`securityGroupIds`

Gruppi di sicurezza Amazon EC2 associati alle istanze avviate nell'ambiente di calcolo. È necessario specificare uno o più gruppi di sicurezza, in `securityGroupIds` o utilizzando un modello di avvio a cui si fa riferimento in `launchTemplate`. Questo parametro è obbligatorio per i lavori eseguiti su risorse Fargate e deve contenere almeno un gruppo di sicurezza. (Fargate non supporta i modelli di avvio). Se i gruppi di sicurezza vengono specificati utilizzando `securityGroupIds` e `launchTemplate`, verranno utilizzati i valori in `securityGroupIds`.

Quando si aggiornano gli ambienti di calcolo, se si fornisce un elenco vuoto di gruppi di sicurezza, il comportamento risultante differisce tra le risorse di calcolo Fargate ed EC2. Per le risorse di calcolo Fargate, se si fornisce un elenco vuoto non viene apportata alcuna modifica. Per le risorse di calcolo EC2, fornendo un elenco vuoto si rimuovono i gruppi di sicurezza dalla risorsa di calcolo. Se si modificano i gruppi di sicurezza, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Questo vale sia per le risorse di calcolo Fargate che per quelle di EC2. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

Tipo: matrice di stringhe

Campo obbligatorio: sì

`ec2KeyPair`

La coppia di chiavi EC2 utilizzata per le istanze lanciate nell'ambiente di calcolo. Puoi utilizzare questa coppia di chiavi per accedere alle tue istanze con SSH. Quando si aggiorna un ambiente di calcolo, se si modifica la coppia di key pair EC2, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

- Tipo: stringa

Campo obbligatorio: no

`instanceRole`

Il profilo dell'istanza Amazon ECS da collegare alle istanze Amazon EC2 in un ambiente di elaborazione. Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate. Non specificarlo. Puoi specificare un Amazon Resource Name (ARN) abbreviato o completo per un profilo dell'istanza. Ad esempio `ecsInstanceRole` o `arn:aws:iam::`*`aws_account_id`*`:instance-profile/ecsInstanceRole`. Per ulteriori informazioni, consulta Ruolo dell'istanza Amazon ECS.

Quando si aggiorna un ambiente di calcolo, se si modifica questa impostazione, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

-Tipo: stringa

Campo obbligatorio: no

`tags`

Tag di coppia chiave-valore da applicare alle istanze EC2 avviate nell'ambiente di calcolo. Ad esempio, puoi specificare `"Name": "AWS Batch Instance - C4OnDemand"` come tag in modo che ciascuna istanza nel tuo ambiente di calcolo abbia lo stesso nome. Ciò è utile per riconoscere le AWS Batch istanze nella console Amazon EC2. Questi tag non vengono visualizzati quando si utilizza l' AWS BatchListTagsForResourceoperazione API.

Quando si aggiorna un ambiente di calcolo, se si modificano i tag EC2, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

`placementGroup`

Il gruppo di collocamento Amazon EC2 da associare alle risorse di calcolo. Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate. Non specificarlo. Se intendi inviare lavori paralleli multinodo al tuo ambiente di elaborazione, prendi in considerazione la creazione di un gruppo di collocamento del cluster e associalo alle tue risorse di elaborazione. Questo mantiene il processo parallelo a più nodi in un gruppo logico di istanze all'interno di una singola zona di disponibilità con un potenziale di flusso di rete elevato. Per ulteriori informazioni, consulta Gruppi di collocamento nella Guida per l'utente di Amazon EC2 per le istanze Linux.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

- Tipo: stringa

Campo obbligatorio: no

`bidPercentage`

La percentuale massima che può raggiungere il prezzo di un'istanza Spot EC2 rispetto al prezzo On-Demand per quel tipo di istanza prima del lancio delle istanze. Ad esempio, se la percentuale massima 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. Per la maggior parte dei casi d'uso, è preferibile lasciare vuoto questo campo.

Quando si aggiorna un ambiente di elaborazione, se si modifica la percentuale di offerta, è necessario un aggiornamento dell'infrastruttura dell'ambiente di elaborazione. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

Campo obbligatorio: no

`spotIamFleetRole`

L'Amazon Resource Name (ARN) del ruolo IAM per il parco istanze Spot Amazon EC2 applicato a un ambiente di calcolo `SPOT`. Questo ruolo è obbligatorio se la strategia di allocazione è impostata su `BEST_FIT` o se la strategia di allocazione non è specificata. Per ulteriori informazioni, consulta Ruolo della flotta spot di Amazon EC2.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

> **Important**
>
> Per etichettare le istanze Spot al momento della creazione, il ruolo IAM di Spot Fleet qui specificato deve utilizzare la nuova politica gestita di AmazonEC2. SpotFleet TaggingRole La politica di gestione dei SpotFleetruoli di AmazonEC2 precedentemente consigliata non dispone delle autorizzazioni necessarie per etichettare le istanze Spot. Per ulteriori informazioni, consulta Istanze Spot non taggate al momento della creazione.

-Tipo: stringa

Obbligatorio: questo parametro è obbligatorio per gli ambienti di calcolo `SPOT`.

`launchTemplate`

Un modello di avvio facoltativo da associare alle risorse di calcolo. Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate. Non specificarlo. Qualsiasi altro parametro di risorsa di calcolo specificato in un'operazione CreateComputeEnvironmento UpdateComputeEnvironmentAPI sostituisce gli stessi parametri nel modello di avvio. Per usare un modello di avvio, è necessario specificare l'ID del modello di avvio o il nome del modello di avvio nella richiesta, ma non entrambi. Per ulteriori informazioni, consulta Supporto modello di avvio.

Quando aggiorni un ambiente di calcolo, per rimuovere il modello di avvio personalizzato e utilizzare il modello di avvio predefinito, imposta la specifica `launchTemplateId` o il `launchTemplateName` membro del modello di avvio su una stringa vuota. La rimozione

del modello di avvio da un ambiente di calcolo non rimuove l'AMI specificato nel modello di avvio, se era quello utilizzato. Per aggiornare l'AMI selezionato da un modello di avvio, il `updateToLatestImageVersion` parametro deve essere impostato su`true`. Quando si aggiorna un ambiente di calcolo, se si modifica il modello di avvio, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

Tipo: LaunchTemplateSpecification

oggetto

Campo obbligatorio: no

`launchTemplateId`

ID del modello di avvio.

- Tipo: stringa

Campo obbligatorio: no

`launchTemplateName`

Il nome del modello di avvio.

- Tipo: stringa

Campo obbligatorio: no

`version`

Numero di versione del modello di lancio, `$Latest` o `$Default`.

Se il valore è `$Latest`, viene utilizzata la versione più recente del modello di lancio. Se il valore è `$Default`, viene utilizzata la versione predefinita del modello di lancio. Durante un aggiornamento dell'infrastruttura, se uno dei due `$Default` è stato specificato `$Latest` o è stato specificato per l'ambiente di calcolo, AWS Batch rivaluta la versione del modello di avvio e potrebbe utilizzare una versione diversa del modello di avvio. Questo vale anche se il modello di avvio non è stato specificato nell'aggiornamento.

Default: `$Default`.

- Tipo: stringa

Campo obbligatorio: no

`ec2Configuration`

Fornisce le informazioni utilizzate per selezionare Amazon Machine Images (AMI) per le istanze nell'ambiente di calcolo EC2. Se `Ec2Configuration` non è specificato, l'impostazione predefinita è Amazon Linux 2 (`ECS_AL2`). Prima del 31 marzo 2021, questa impostazione predefinita era Amazon Linux (`ECS_AL1`) per istanze non GPU e non AWS Graviton.

Quando si aggiorna un ambiente di calcolo, se si modifica questo parametro, è necessario un aggiornamento dell'infrastruttura dell'ambiente di calcolo. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **Note**
>
> Questo parametro non è applicabile ai processi in esecuzione su risorse Fargate.

Tipo: matrice di oggetti Ec2Configuration

Campo obbligatorio: no

`imageIdOverride`

L'ID AMI utilizzato per le istanze avviate nell'ambiente di calcolo che corrisponde al tipo di immagine. Questa impostazione sostituisce l'insieme `imageId` nell'oggetto `computeResource`.

- Tipo: stringa

Campo obbligatorio: no

`imageKubernetesVersion`

La Kubernetes versione per l'ambiente di calcolo. Se non specifichi un valore, viene utilizzata la versione più recente supportata da AWS Batch .

- Tipo: stringa

Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 256 caratteri.

Campo obbligatorio: no

`imageType`

Il tipo di immagine da associare al tipo di istanza per selezionare un'AMI. I valori supportati sono diversi per risorse `ECS` e `EKS`.

ECS

Se il parametro `imageIdOverride` non è specificato, viene utilizzata un'AMI Amazon Linux 2 ottimizzata per Amazon ECS (`ECS_AL2`) recente. Se in un aggiornamento viene specificato un nuovo tipo di immagine, ma non viene specificato `imageId` né un `imageIdOverride` parametro, viene utilizzata l'AMI ottimizzata Amazon ECS più recente per quel tipo di immagine AWS Batch supportata da.

ECS_AL2

Amazon Linux 2: impostazione predefinita per tutte le famiglie di istanze non GPU.

ECS_AL2_NVIDIA

Amazon Linux 2 (GPU): impostazione predefinita per tutte le famiglie di istanze GPU (ad esempio `P4` e`G4`) e può essere utilizzata per tutti i tipi di istanze non basati su AWS Graviton.

ECS_AL1

Amazon Linux. Amazon Linux ha raggiunto end-of-life il supporto standard. Per ulteriori informazioni, consulta AMI Amazon Linux.

EKS

Se il parametro `imageIdOverride` non è specificato, viene utilizzata un'AMI Amazon Linux ottimizzata per Amazon EKS (`EKS_AL2`) recente. Se in un aggiornamento viene specificato un nuovo tipo di immagine, ma non viene specificato `imageId` né un né un `imageIdOverride` parametro, viene utilizzata l'ultima AMI ottimizzata Amazon EKS per quel tipo di immagine che AWS Batch supporta.

EKS_AL2

Amazon Linux 2: impostazione predefinita per tutte le famiglie di istanze non GPU.

EKS_AL2_NVIDIA

Amazon Linux 2 (accelerato): predefinito per tutte le famiglie di istanze GPU (ad esempio `P4` e`G4`) e può essere utilizzato per tutti i tipi di istanze non basati su AWS Graviton.

- Tipo: stringa

  Limitazioni di lunghezza: lunghezza minima pari a 1. La lunghezza massima è 256 caratteri.

  Campo obbligatorio: sì

Configurazione Amazon EKS

Configurazione per il cluster Amazon EKS che supporta l'ambiente di AWS Batch elaborazione. Il cluster deve esistere già per poter creare l'ambiente di calcolo.

`eksClusterArn`

Il nome della risorsa Amazon (ARN) del cluster Amazon EKS. Un esempio è `arn:aws:eks:`*`us-east-1`*`:`*`123456789012`*`:cluster/`*`ClusterForBatch`*.

Tipo: stringa

Campo obbligatorio: sì

`kubernetesNamespace`

Lo spazio dei nomi del cluster Amazon EKS. AWS Batch gestisce i pod in questo spazio dei nomi. Il valore non può essere vuoto o nullo. Deve contenere meno di 64 caratteri, non può essere impostato su `default`, non può iniziare con "`kube-`" e deve corrispondere a questa espressione regolare: `^[a-z0-9]([-a-z0-9]*[a-z0-9])?$`. Per ulteriori informazioni, consulta Spazi dei nomi nella documentazione di Kubernetes.

Tipo: stringa

Campo obbligatorio: sì

Tipo: oggetto EksConfiguration

Campo obbligatorio: no

Ruolo del servizio

`serviceRole`

L'Amazon Resource Name (ARN) completo del ruolo IAM che consente di AWS Batch effettuare chiamate ad altri AWS servizi per tuo conto. Per ulteriori informazioni, consulta Utilizzo di ruoli collegati ai servizi per AWS Batch. Ti consigliamo di non specificare il ruolo del servizio. In questo modo, AWS Batch utilizza il ruolo AWSServiceRoleForBatchcollegato al servizio.

> **Important**
>
> Se il tuo account ha già creato il ruolo AWS Batch collegato al servizio (AWSServiceRoleForBatch), quel ruolo viene utilizzato per impostazione predefinita per il tuo ambiente di calcolo, a meno che tu non specifichi un ruolo qui. Se il ruolo AWS Batch collegato al servizio non esiste nel tuo account e non è specificato alcun ruolo qui, il servizio tenta di creare il ruolo collegato al AWS Batch servizio nel tuo account. Per ulteriori informazioni sul ruolo collegato al servizio AWSServiceRoleForBatch, consulta Autorizzazioni di ruolo collegate al servizio per AWS Batch.
> Se l'ambiente di calcolo viene creato utilizzando il ruolo AWSServiceRoleForBatchcollegato al servizio, non può essere modificato per utilizzare un normale ruolo IAM. Allo stesso modo, se l'ambiente di calcolo viene creato con un normale ruolo IAM, non può essere modificato per utilizzare il ruolo collegato al servizio. AWSServiceRoleForBatch Per aggiornare i parametri dell'ambiente di calcolo che richiedono un aggiornamento dell'infrastruttura per essere modificati, è necessario utilizzare il ruolo collegato al AWSServiceRoleForBatchservizio. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

Se il ruolo specificato ha un percorso diverso da`/`, assicurati di specificare l'ARN completo del ruolo (consigliato) o di anteporre il percorso al nome del ruolo.

> **Note**
>
> A seconda di come hai creato il tuo ruolo AWS Batch di servizio, il relativo Amazon Resource Name (ARN) potrebbe contenere il prefisso del `service-role` percorso. Quando si specifica solo il nome del ruolo di servizio, AWS Batch si presuppone che l'ARN non utilizzi il prefisso `service-role` del percorso. Per questo motivo, ti

consigliamo di specificare l'ARN completo del ruolo di servizio al momento della creazione di ambienti di calcolo.

-Tipo: stringa

Campo obbligatorio: no

Tag

`tags`

Tag di coppia chiave-valore da associare all'ambiente di calcolo. Per ulteriori informazioni, consulta Tagging delle risorse AWS Batch.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Configurazioni EC2

AWS Batch utilizza AMI ottimizzate Amazon ECS per ambienti di calcolo EC2 ed EC2 Spot. L'impostazione predefinita è Amazon Linux 2 (`ECS_AL2`). Prima del 31 marzo 2021, questa impostazione predefinita era Amazon Linux (`ECS_AL1`) per istanze non GPU e non AWS Graviton.

> **Note**
>
> AWS Batch supporta anche Amazon Linux 2023.

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, consulta Aggiornamento degli ambienti di calcolo - AWS Batch

Strategie di allocazione

Quando viene creato un ambiente di elaborazione gestito, AWS Batch seleziona i tipi di istanza tra quelli `instanceTypes` specificati che meglio si adattano alle esigenze dei processi. 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 le istanze aggiuntive siano disponibili. Se non ci sono abbastanza istanze disponibili o se l'utente sta raggiungendo le quote di servizio Amazon EC2, 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`, è necessario specificare il ruolo IAM di Spot Fleet. `BEST_FIT`non è supportato durante l'aggiornamento degli ambienti di elaborazione. Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione.

> **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.

`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.

Aggiornamento degli ambienti di elaborazione

Dopo aver creato un ambiente di calcolo che utilizza risorse EC2, puoi aggiornare direttamente molte delle impostazioni dell'ambiente di calcolo. Tuttavia, la modifica di alcune impostazioni richiede la AWS Batch sostituzione delle istanze nell'ambiente di calcolo.

Per gli ambienti di calcolo che utilizzano risorse Fargate, è possibile aggiornare quanto segue.

- `securityGroupIds`
- `subnets`
- `desiredvCpus`
- `maxvCpus`
- `minvCpus`

AWS Batchdispone di due meccanismi di aggiornamento. Il primo è un aggiornamento di scalabilità in cui le istanze vengono aggiunte o rimosse dall'ambiente di calcolo. Il secondo è un aggiornamento dell'infrastruttura in cui le istanze nell'ambiente di calcolo vengono sostituite. Un aggiornamento dell'infrastruttura richiede molto più tempo rispetto a un aggiornamento di scalabilità.

Se si aggiornano gli ambienti di elaborazione conAWS Batch, la modifica solo di queste impostazioni provoca un aggiornamento scalabile: vCPU desiderate ()`desiredvCpus`, numero massimo di vCPU (), vCPU minimo (`maxvCpus`), ruolo di servizio () e stato (). `minvCpus` `serviceRole` `state`

> **Note**
>
> Quando si aggiorna l'`desiredvCpus`impostazione, il valore deve essere compreso tra i valori e. `minvCpus` `maxvCpus`
> Inoltre, il `desiredvCpus` valore aggiornato deve essere maggiore o uguale al `desiredvCpus` valore corrente. Per ulteriori informazioni, consulta the section called “Messaggio di errore quando si aggiorna l'`desiredvCpus`impostazione”.

Se una delle seguenti impostazioni viene modificata in un'azione UpdateComputeEnvironmentAPI, AWS Batch avvia un aggiornamento dell'infrastruttura. Un aggiornamento dell'infrastruttura richiede che il ruolo del servizio sia impostato su AWSServiceRoleForBatch(impostazione predefinita) e che la strategia di allocazione sia `BEST_FIT_PROGRESSIVE` `SPOT_CAPACITY_OPTIMIZED`, o.

`SPOT_PRICE_CAPACITY_OPTIMIZED` `BEST_FIT`non è supportato. Ad eccezione del ruolo di servizio, tutte le impostazioni che possono essere modificate per un aggiornamento di scalabilità possono essere modificate anche per un aggiornamento dell'infrastruttura.

> **Note**
>
> Si consiglia di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` anziché `SPOT_CAPACITY_OPTIMIZED` n nella maggior parte dei casi.

Durante un aggiornamento dell'infrastruttura, lo stato dell'ambiente di elaborazione cambia in. `UPDATING` Le nuove istanze vengono avviate utilizzando le impostazioni aggiornate. Sulle nuove istanze vengono pianificati nuovi lavori. I lavori attualmente in esecuzione vengono inviati in base alla politica di aggiornamento dell'infrastruttura. Per ulteriori informazioni, consulta le pagine UpdateComputeEnvironment e UpdatePolicy nella Documentazione di riferimento dell'API AWS Batch.

Nel tipo di `UpdatePolicy` dati, considera i seguenti scenari:

> **Note**
>
> In questi scenari, è vero quanto segue. Quando un'istanza viene terminata, i processi in esecuzione vengono interrotti. Per impostazione predefinita, questi processi non vengono ritentati. Per riprovare uno di questi processi dopo la chiusura di un'istanza, configura una strategia per riprovare il processo. Per ulteriori informazioni, consulta the section called "Ritentativi di lavoro automatizzati" nella Guida per l'utente di AWS Batch.

- Se l'`terminateJobsOnUpdate`impostazione è impostata su`true`, i processi in esecuzione vengono interrotti durante un aggiornamento dell'infrastruttura. L'`jobExecutionTimeoutMinutes`impostazione viene ignorata.
- Se l'`terminateJobsOnUpdate`impostazione è impostata su`false`, i lavori possono essere eseguiti per un periodo di tempo aggiuntivo dopo l'aggiornamento dell'infrastruttura. Questo tempo aggiuntivo è configurato nell'`jobExecutionTimeoutMinutes`impostazione. Per impostazione predefinita, l'`jobExecutionTimeoutMinutes`impostazione è 30 minuti.

Man mano che la capacità diventa disponibile nell'ambiente di elaborazione, vengono lanciate nuove istanze con le impostazioni aggiornate e vengono avviati i processi sulle nuove istanze. Quando tutti i

processi vengono completati sulle istanze con le impostazioni precedenti, le vecchie istanze vengono terminate. La capacità disponibile significa che il numero desiderato di vCPU è inferiore al numero massimo di vCPU di almeno un numero di vCPU pari a quello richiesto dal tipo di istanza più piccolo.

Aggiornamenti dell'infrastruttura

È necessario un aggiornamento dell'infrastruttura per modificare alcune impostazioni per un ambiente di elaborazione. Se viene modificata una delle seguenti impostazioni, viene avviato un aggiornamento dell'infrastruttura:

> **Important**
>
> L'ambiente di calcolo deve utilizzare il ruolo AWSServiceRoleForBatchcollegato al servizio per apportare modifiche che richiedono un aggiornamento dell'infrastruttura.
> Se l'ambiente di calcolo utilizza un ruolo collegato al servizio, non può essere modificato per utilizzare un normale ruolo IAM. Allo stesso modo, se l'ambiente di calcolo ha un ruolo IAM regolare, non può essere modificato per utilizzare un ruolo collegato al servizio. Pertanto, è possibile eseguire aggiornamenti dell'infrastruttura solo su ambienti di calcolo creati utilizzando un ruolo collegato al servizio.

- La strategia di allocazione (`allocationStrategy`, deve essere o`BEST_FIT_PROGRESSIVE`. `SPOT_CAPACITY_OPTIMIZED` `SPOT_PRICE_CAPACITY_OPTIMIZED` Se la strategia di allocazione originale è`BEST_FIT`, gli aggiornamenti dell'infrastruttura non sono supportati.)

  > **Note**
  >
  > Si consiglia di utilizzare `SPOT_PRICE_CAPACITY_OPTIMIZED` anziché `SPOT_CAPACITY_OPTIMIZED` n nella maggior parte dei casi.

- Percentuale di offerta () `bidPercentage`
- Configurazione EC2 () `ec2Configuration`
- Coppia di chiavi () `ec2KeyPair`
- ID immagine (`imageId`)
- Ruolo dell'istanza (`instanceRole`)
- Tipi di istanze (`instanceTypes`)
- Modello di avvio (`launchTemplate`)

- Gruppo di posizionamento (`placementGroup`)
- Gruppi di sicurezza (`securityGroupIds`)
- Sottoreti VPC () `subnets`
- Tag EC2 () `tags`
- Tipo di ambiente di calcolo (`type`, può essere uno di `EC2` o) `SPOT`
- Se eseguire l'aggiornamento all'AMI più recente supportata da AWS Batch durante un aggiornamento dell'infrastruttura `updateToLatestImageVersion`

Aggiornamento dell'ID AMI

Durante un aggiornamento dell'infrastruttura, l'ID AMI dell'ambiente di calcolo potrebbe cambiare, a seconda che le AMI siano specificate in una di queste tre impostazioni. Le AMI sono specificate nel modello `imageId` (in`computeResources`), `imageId0verride` (in`ec2Configuration`) o nel modello di avvio specificato in. `launchTemplate` Supponiamo che nessun ID AMI 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 `imageId`o`imageId0verride`, 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 gli ID AMI. Ciò richiede infatti la specificazione di una stringa vuota per quel parametro. `imageId` Perché`imageIdOverride`, 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`.

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.

Argomenti

Selezione AMI 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 sulle AMI ottimizzate per Amazon EKS, consulta le AMI Amazon Linux ottimizzate per Amazon EKS nella Guida per l'utente di Amazon EKS.

Esegui il comando seguente per vedere quale tipo di AMI AWS Batch è 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 1.29 AMI,
 depending on instance types
$ aws batch describe-compute-environments --compute-environments My-Eks-CE1 \
    | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2",
    "imageKubernetesVersion": "1.29"
  }
]
```

L'esempio seguente è un tipo di istanza GPU.

```
# GPU CE example: indicates Batch has choosen the AL2 x86 EKS Accelerated 1.29 AMI
$ aws batch describe-compute-environments --compute-environments My-Eks-GPU-CE \
    | jq '.computeEnvironments[].computeResources.ec2Configuration'
[
  {
    "imageType": "EKS_AL2_NVIDIA",
    "imageKubernetesVersion": "1.29"
  }
]
```

Versioni di Kubernetes supportate

AWS Batch su Amazon EKS attualmente supporta le seguenti Kubernetes versioni:

- 1.29
- 1.28
- 1.27
- 1.26
- 1.25
- 1.24
- 1.23

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
```

Aggiornamento della 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=1.23}]'
```

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 Esecuzione di un DaemonSet su nodi AWS Batch gestiti.

AWS Batch non aggiorna automaticamente le AMI dell'ambiente di calcolo. È tua responsabilità aggiornarle. Esegui il comando seguente per aggiornare le tue AMI alla versione AMI 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 dell'ambiente informatico alla *1.23*.

```
$ aws batch update-compute-environment \
    --compute-environment <compute-environment-name> \
    --compute-resources \
      'ec2Configuration=[{imageType=EKS_AL2,imageKubernetesVersion=1.23}]'
```

Quando esegui l'aggiornamento a un'AMI o alla Kubernetes versione più recente, puoi 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 finiscono (`jobExecutionTimeoutMinutes`.) Per ulteriori informazioni, consulta Aggiornamento degli ambienti di elaborazione e la policy di aggiornamento dell'infrastruttura (UpdatePolicy) impostata nell'operazione UpdateComputeEnvironmentAPI.

Esecuzione di un DaemonSet su nodi AWS Batch gestiti

AWS Batch imposta contaminazioni sui Kubernetes nodi AWS Batch gestiti. 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"
```

Personalizzazione con modelli di lancio

AWS Batch su Amazon EKS supporta i modelli di lancio. Esistono dei vincoli su ciò che il modello di lancio può fare.

Important

AWS Batch corre. `/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.

Note

Se il modello di lancio viene modificato dopo la CreateComputeEnvironmentchiamata, UpdateComputeEnvironmentdeve essere chiamato per valutare la versione del modello di lancio da sostituire.

Argomenti

- Aggiungere argomenti kubelet aggiuntivi
- Configurazione del runtime del contenitore

- Montaggio di un volume Amazon EFS
- Supporto IPv6

Aggiungere 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, *`--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==--
```

Configurazione del 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, containerdconsultate la Kubernetesdocumentazione.

> **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==--
```

Montaggio di 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.

Supporto IPv6

AWS Batch supporta i cluster Amazon EKS con indirizzi IPv6. Non sono richieste personalizzazioni per il supporto. AWS Batch Tuttavia, prima di iniziare, ti consigliamo di rivedere le considerazioni e le condizioni descritte in Assegnazione di indirizzi IPv6 a pod e servizi nella Amazon EKS User Guide.

Risorsa di calcoloGestione della memoria

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 default delle risorse di calcolo AWS Batch, inoltre, riserva 32 MiB di memoria per l'agente del container Amazon ECS e altri processi di sistema critici. Questa memoria non è disponibile per l'allocazione dei lavori. Per ulteriori informazioni, consulta Allocazione della memoria di sistema.

L'agente del container di Amazon ECS utilizza la funzione Docker `ReadMemInfo()` per eseguire una query sulla memoria totale disponibile per il sistema operativo. 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.

Allocazione della memoria di sistema

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 di default delle risorse di calcolo AWS Batch riserva 32 MiB di memoria per l'agente del container di Amazon ECS e altri processi di sistema critici.

Visualizzazione della memoria della risorsa 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 all'indirizzo 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 registrata è quello che la risorsa di elaborazione ha registrato con Amazon ECS al momento del primo avvio e il valore di memoria disponibile è quello che non è già stato assegnato 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 Batchutilizza 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 Batchutilizza 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 Batchutilizza il `requests` valore per ridimensionare le decisioni. Se non `requests` viene specificato un valore, il `limits` valore viene copiato nel `requests` valore.

Esempi di definizioni di lavoro

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 sono in esecuzione altri job, al job viene assegnato un massimo di 2 vCPU. 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 del processo AWS Batch su Amazon EKS ha un `request` valore 1 e alloca un massimo di 4 vCPU al processo.

```
{
    "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 Batchsi 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 di prenotazione della CPU del nodo

Il valore di prenotazione della CPU viene calcolato in millicore utilizzando il numero totale di vCPU 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 vCPU è 70 m. Viene calcolato nel modo seguente: (1*60) + (1*10) = 70 m.
- Il valore di prenotazione della CPU per un'`c5.24xlarge`istanza con 96 vCPU 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 di prenotazione della memoria dei nodi

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 * (maximum_number_of_pods)) - 255 - (value. HardEvictionLimit*)).

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 usiDaemonSets, considera 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 Batchinizialmente

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.

Politiche di pianificazione

È possibile utilizzare le politiche di pianificazione per configurare la modalità di allocazione delle risorse di elaborazione in una coda di lavoro tra utenti o carichi di lavoro. Utilizzando i criteri di pianificazione, puoi assegnare diversi identificatori di fair share ai carichi di lavoro o agli utenti. AWS Batchassegna a ciascun identificatore di fair share una percentuale delle risorse totali disponibili in un determinato periodo di tempo.

La percentuale di fair share viene calcolata utilizzando i valori `shareDecaySeconds` and`shareDistribution`. È possibile aggiungere tempo all'analisi del fair share assegnando un periodo di decadimento delle azioni alla polizza. L'aggiunta del tempo dà più peso al tempo e meno al peso definito. Puoi riservare le risorse di calcolo agli identificatori di fair share che non sono attivi specificando una prenotazione di elaborazione. Per ulteriori informazioni, consulta Parametri della politica di pianificazione.

Argomenti

Creazione di 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, si associano uno o più identificatori di fair share o prefissi di fair share ai pesi della coda e, facoltativamente, si assegna un periodo di decadenza 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, inserisci un valore intero per il tempo di decadimento della condivisione della policy di pianificazione. Un tempo di decadimento delle condivisioni più lungo sarà necessario considerare l'utilizzo delle risorse di calcolo per un periodo di tempo

più lungo durante la pianificazione dei lavori. Ciò può consentire ai lavori che utilizzano un identificatore di fair share di utilizzare temporaneamente più risorse di calcolo di quante ne consentirebbe il peso attribuito a quell'identificatore di fair share se tale identificatore non avesse utilizzato recentemente risorse di calcolo.

6. (Facoltativo) Per Compute booking, inserisci un valore intero per la prenotazione di elaborazione della policy di pianificazione. La riserva di elaborazione conterrà alcune vCPU come riserva da utilizzare per identificatori di fair share che non sono attualmente attivi.

   Il rapporto riservato è `(`*`computeReservation`*`/100)^`*`ActiveFairShares`* dove ActiveFairShares è il numero di identificatori di ripartizione equa attivi.

   Ad esempio, un `computeReservation` valore pari a 50 indica che è AWS Batch necessario riservare il 50% della VCPU massima disponibile se esiste un solo identificatore di fair share, il 25% se vi sono due identificatori di fair share e il 12,5% se vi sono tre identificatori di fair share. Un `computeReservation` valore pari a 25 indica che è AWS Batch necessario riservare il 25% della VCPU massima disponibile se esiste un solo identificatore di fair share, il 6,25% se vi sono due identificatori di fair share e l'1,56% se vi sono tre identificatori di fair share.

7. Nella sezione Attributi di condivisione, è possibile specificare l'identificatore di fair share e il peso per ogni identificativo di fair share da associare alla politica di programmazione.

   a. Scegli Aggiungi identificatore di condivisione.

   b. Per l'identificatore Share, specifica l'identificatore Fair Share. Se la stringa termina con '*', diventa un prefisso identificativo di fair share utilizzato per abbinare gli identificatori di fair share per le offerte di lavoro. Tutti gli identificatori di fair share e i prefissi degli identificatori di fair share in una politica di pianificazione devono essere unici e non possono sovrapporsi. Ad esempio, non è possibile inserire gli identificatori di fair share come prefisso 'userA*' e l'identificatore di fair share 'userA1' nella stessa politica di pianificazione.

   c. Per il fattore di peso, specifica il peso relativo per l'identificatore del fair share. 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 fair share, i lavori con identificatori di fair share che iniziano con il prefisso condivideranno il fattore di ponderazione. Ciò aumenta efficacemente il fattore di ponderazione per tali lavori, abbassando la priorità individuale ma mantenendo lo stesso fattore di peso per il prefisso identificativo della quota equa.

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 Tagging delle risorse AWS Batch.

9. Scegli Invia per completare e creare la tua politica di pianificazione.

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 BatchAPI Reference.

```
{
    "name": "",
    "fairsharePolicy": {
        "shareDecaySeconds": 0,
        "computeReservation": 0,
        "shareDistribution": [
            {
                "shareIdentifier": "",
                "weightFactor": 0.0
            }
        ]
    },
    "tags": {
        "KeyName": ""
    }
}
```

> **Note**
>
> È possibile generare il modello di coda dei lavori precedente con il seguente AWS CLI comando.
>
> ```
> $ aws batch create-scheduling-policy --generate-cli-skeleton
> ```

Parametri della politica di pianificazione

Le politiche di pianificazione sono suddivise in tre componenti di base: il nome, la politica di condivisione equa e i tag della politica di pianificazione.

Argomenti

Nome della politica di pianificazione

`name`

Il nome della politica di pianificazione. Il nome può contenere un massimo di 128 lettere (maiuscole e minuscole), numeri, trattini e caratteri di sottolineatura.

Tipo: stringa

Campo obbligatorio: sì

Politica di condivisione equa

`fairsharePolicy`

La policy di ripartizione equa della policy di pianificazione.

```
"fairsharePolicy": {
   "computeReservation": number,
   "shareDecaySeconds": number,
   "shareDistribution": [
      {
         "shareIdentifier": "string",
         "weightFactor": number
      }
   ]
}
```

Tipo: oggetto

Campo obbligatorio: no

`computeReservation`

Un valore utilizzato per riservare parte della VCPU massima disponibile agli identificatori di fair share che non sono ancora stati utilizzati.

Il rapporto riservato è *computeReservation*/100)^*ActiveFairShares* dove ActiveFairShares è il numero di identificatori di ripartizione equa 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 fair share attivo, il 25% se vi sono due identificatori di fair share attivi e il 12,5% se vi sono tre identificatori di fair share attivi. 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 fair share attivo, il 6,25% se vi sono due identificatori di fair share attivi e l'1,56% se vi sono tre identificatori di fair share attivi.

Tipo: integer

Intervallo valido: valore minimo pari a 0. Valore massimo di 99.

Campo obbligatorio: no

`shareDecaySeconds`

Il periodo di tempo da utilizzare per calcolare la percentuale di fair share per ogni identificatore di fair share utilizzato. Un valore pari a zero (0) indica che è necessario misurare solo l'uso corrente. Con il decadimento, i lavori eseguiti di recente presentano un peso maggiore rispetto a quelli eseguiti in precedenza.

Tipo: integer

Intervallo valido: valore minimo pari a 0. Valore massimo di 604800 (1 settimana).

Campo obbligatorio: no

`shareDistribution`

Serie di oggetti che contengono i pesi degli identificatori di fair share per la politica di fair share. Gli identificatori di fair share non inclusi hanno un peso predefinito di. `1.0`

```
"shareDistribution": [
   {
      "shareIdentifier": "string",
      "weightFactor": number
   }
]
```

Tipo: Array

Campo obbligatorio: no

`shareIdentifier`

Un identificatore di ripartizione equa o un suo prefisso. Se la stringa termina con '*', questa stringa specifica un prefisso identificativo di fair share per gli identificatori di fair share che iniziano con quel prefisso. Ad esempio, se il valore è `UserA*` e il `weightFactor` è 1 e ci sono due identificatori di fair share che iniziano con`UserA`, ognuno di questi identificatori di fair share avrà un peso pari a 2; se ci sono cinque identificatori di fair share, ognuno avrà un peso di 5.

L'elenco degli identificativi di fair share e dei prefissi degli identificativi di fair share in una politica di fair share non può sovrapporsi. Ad esempio, non è possibile avere un prefisso identificativo di fair share pari a `UserA*` e un identificativo di fair share nella stessa politica di fair share. `UserA-1`

Tipo: stringa

Campo obbligatorio: sì

`weightFactor`

Il fattore di ponderazione per l'identificatore di ripartizione equa. Il valore predefinito è 1.0. Un valore inferiore ha una priorità più alta per le risorse di calcolo. I processi che utilizzano un identificatore di ripartizione con un fattore di ponderazione pari a 0,125 (1/8), ad esempio, ottengono 8 volte le risorse di calcolo rispetto agli identificatori di ripartizione con fattore di ponderazione 1.

Il valore più piccolo supportato è 0,0001 e il valore massimo supportato è 999,9999.

Tipo: Float

Campo obbligatorio: no

Tag

`tags`

Tag di coppia chiave-valore da associare alla politica di pianificazione. Per ulteriori informazioni, consulta Tagging delle risorse AWS Batch.

Tipo: mappatura stringa a stringa

Campo obbligatorio: no

Orchestra i AWS Batch lavori con le macchine a stati Step Functions nella console AWS Batch

È possibile utilizzare la AWS Batch console per visualizzare i dettagli sulle macchine a stati Step Functions e sulle funzioni che utilizzano.

Sections

Visualizzazione dei 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 console AWS Batch apre la pagina Dettagli.
   </result>

Per ulteriori informazioni, consulta Step Functions nella la Guida per sviluppatori di AWS Step Functions.

Modifica di 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.

Esecuzione di 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. Scegliere 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 Batchsu AWS Fargate

AWSFargate è una tecnologia che puoi utilizzare AWS Batch per eseguire container senza dover gestire server o cluster di istanze Amazon EC2. Con AWS Fargate, non è più necessario effettuare il provisioning, configurare o dimensionare i cluster di macchine virtuali per eseguire i container. Viene anche eliminata la necessità di scegliere i tipi di server, di decidere quando dimensionare i cluster o ottimizzarne il packing.

Quando 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.

Indice

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 istanza Amazon EC2. 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 vCPU
- 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'infrastruttura Amazon EC2. Ad esempio, se il numero di lavori eseguiti contemporaneamente supera i limiti di limitazione di Fargate. Questo perché, con EC2, i lavori possono essere assegnati a una velocità maggiore alle risorse EC2 rispetto alle risorse di Fargate. Inoltre, è possibile eseguire più lavori contemporaneamente quando si utilizza EC2. Per ulteriori informazioni, consulta le quote dei servizi AWS Fargate nella Amazon Elastic Container Service Developer Guide.

Definizioni di lavoro su Fargate

AWS Batchi lavori su Fargate non supportano tutti i parametri di definizione dei processi disponibili. Alcuni parametri non sono affatto supportati e altri si comportano diversamente 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 Utilizzo del driver di log 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 esecuzione (ruolo IAM).

Job in coda a Fargate

AWS Batchle code di lavoro su Fargate sono sostanzialmente invariate. L'unica restrizione è che gli ambienti di elaborazione elencati `computeEnvironmentOrder` devono essere tutti ambienti di calcolo Fargate (o). `FARGATE` `FARGATE_SPOT` Gli ambienti di calcolo EC2 e Fargate non possono essere combinati.

Ambienti di calcolo su Fargate

AWS Batchgli ambienti di calcolo su 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`

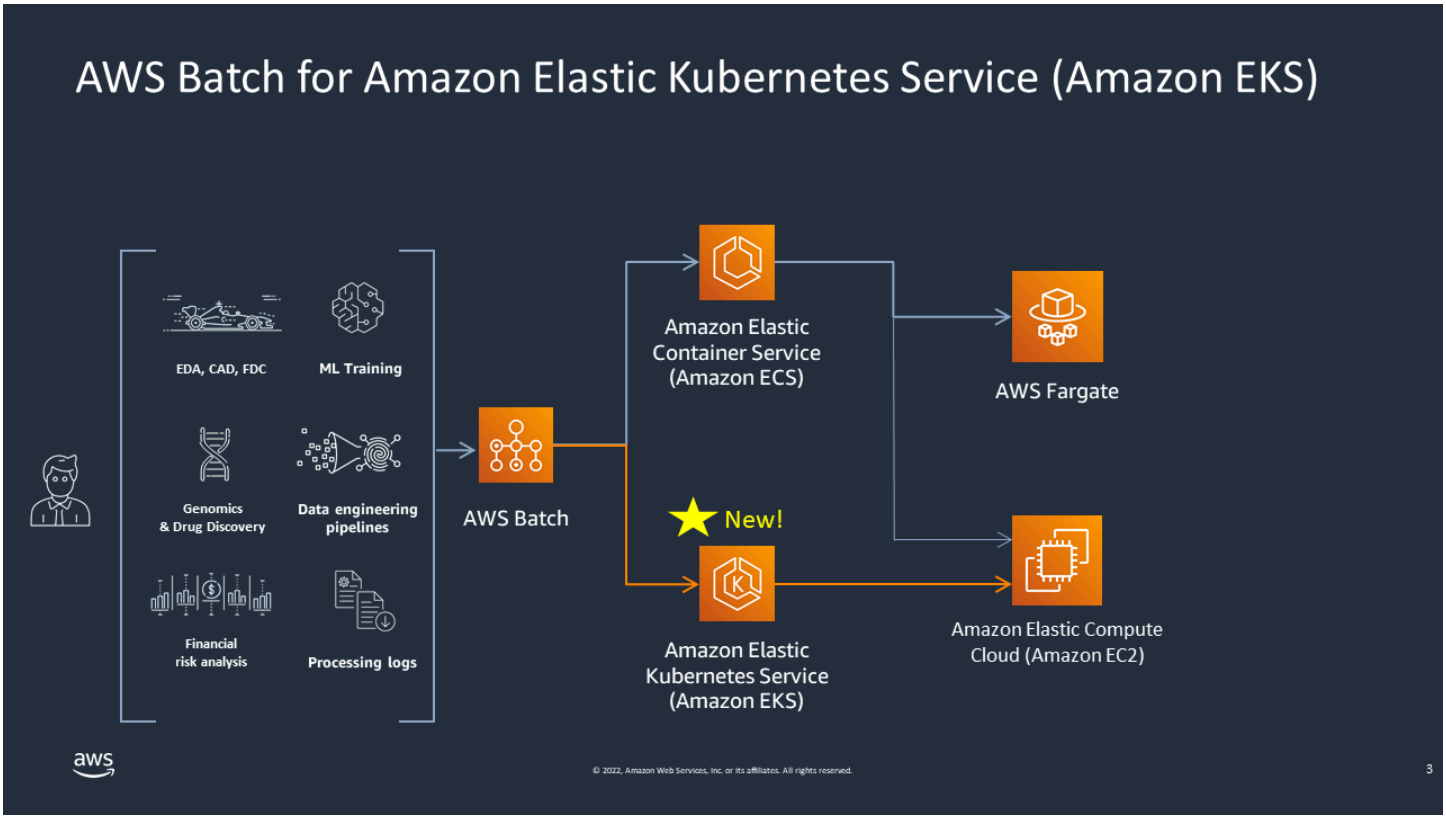
Questo non è applicabile agli ambienti di calcolo Fargate e non può essere fornito. Per specificare i tag per gli ambienti di calcolo Fargate, utilizzate il `tags` parametro che non si trova nell'oggetto. `computeResources`

`type`

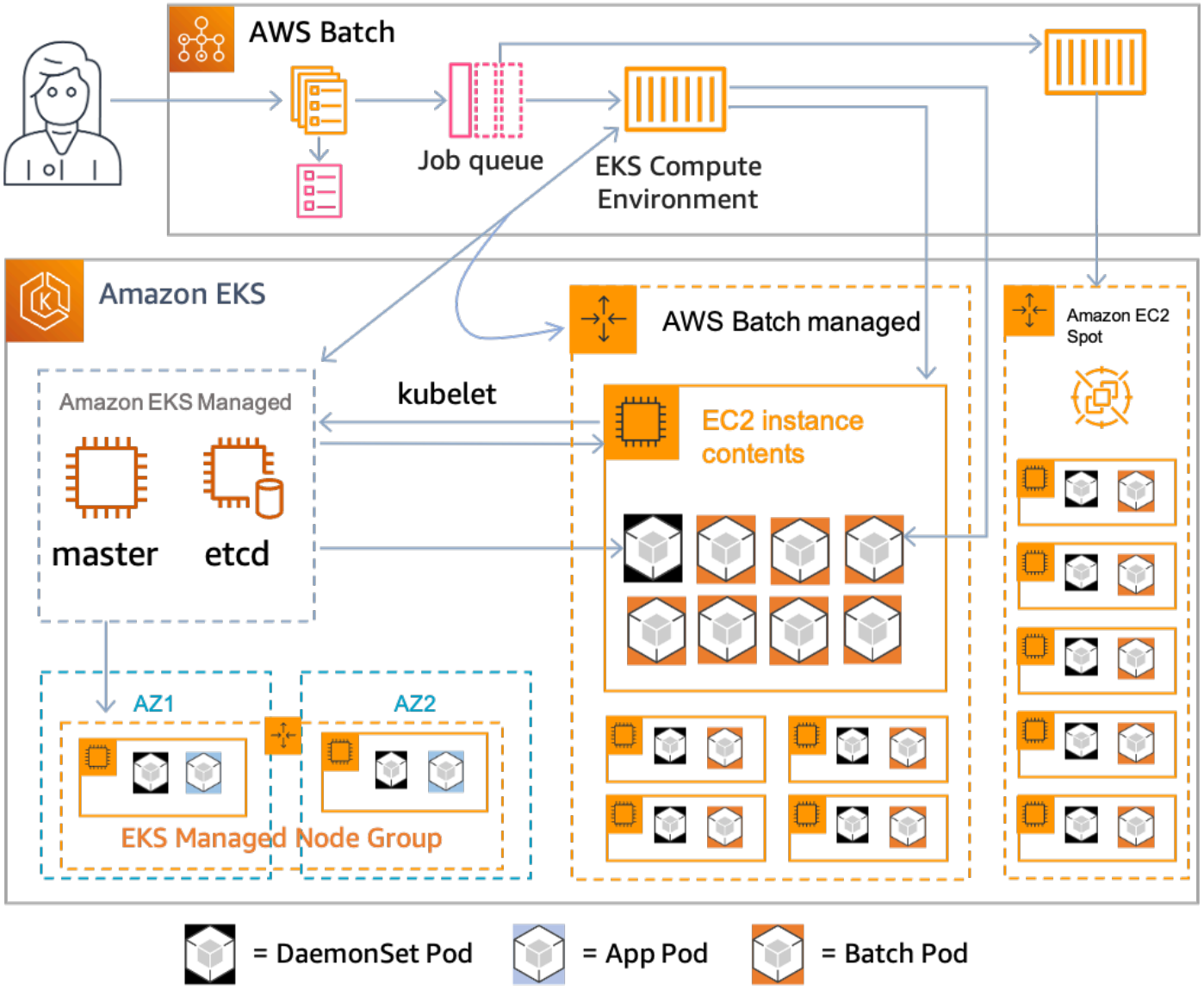
Questo deve essere `FARGATE` o `FARGATE_SPOT`.

```
"type": "FARGATE_SPOT"
```

AWS Batch su 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 istanze Amazon EC2. AWS Batch integra diverse best practice Spot di Amazon EC2 per eseguire i carichi di lavoro con tolleranza ai guasti, riducendo al minimo le 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 chiama le Kubernetes API per creare, monitorare ed eliminare pod e nodi. Kubernetes

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.

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 gli EFA, consulta Tipi di istanze supportati nella Guida per l'utente di Amazon EC2.

  > **Tip**
  >
  > Per visualizzare un elenco di tipi di istanze che supportano gli EFA in un 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` è 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 l'utente di Amazon EC2.

AWS BatchPolitiche, 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 BatchAPI, la AWS Batch console o ilAWS 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 Batcheffettua 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.

Nozioni di base

Una policy IAM deve concedere o negare le autorizzazioni per utilizzare una o più azioni. AWS Batch

Argomenti

- Struttura delle policy
- Autorizzazioni a livello di risorsa supportate per le operazioni API AWS Batch
- Policy di esempio
- Policy gestita di AWS Batch
- Creazione di AWS Batch policy IAM
- Ruolo dell'istanza Amazon ECS
- Ruolo della flotta spot di Amazon EC2
- EventBridge Ruolo IAM

Struttura delle policy

Nei seguenti argomenti viene illustrata la struttura di una policy IAM.

Argomenti

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"
    }
      }
    }
  ]
}
```

Una dichiarazione è costituita da diversi elementi:

- Effect (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. Operazioni 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 a livello di risorsa supportate per le operazioni API AWS Batch e Amazon Resource Name (ARN) 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 Batch Creazione di AWS Batch policy IAM

Operazioni per AWS Batch

In una dichiarazione di policy IAM, è possibile specificare qualsiasi operazione API per qualsiasi servizio che supporta IAM. PerAWS Batch, utilizza il seguente prefisso con il nome dell'azione API: `batch:` (ad esempio, `batch:SubmitJob` and`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.

Amazon Resource Name (ARN) per AWS Batch

Ogni dichiarazione di policy IAM si applica alle risorse specificate utilizzando i rispettivi Amazon Resource Names (ARN).

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'ID dell'Account AWS 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 BatchLe operazioni API attualmente supportano le autorizzazioni a livello di risorsa su diverse operazioni API. Per ulteriori informazioni, consulta Autorizzazioni a livello di risorsa supportate per le operazioni API AWS Batch. Per specificare tutte le risorse o se un'azione API specifica non supporta gli ARN, includi un carattere jolly (*) nell'elemento. `Resource`

```
"Resource": "*"
```

Verifica che gli utenti dispongano delle autorizzazioni necessarie

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 ServiceAPI Reference e decode-authorization-messageil AWS CLICommand Reference.

Autorizzazioni a livello di risorsa supportate per le operazioni 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 Batchsupporta 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 possono 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.

L'elenco seguente descrive le azioni AWS Batch API che attualmente supportano le autorizzazioni a livello di risorsa. L'elenco descrive anche le risorse supportate, gli ARN delle risorse e le chiavi di condizione per ogni azione.

> **Important**
>
> Se un'azione AWS Batch API non è elencata in questo elenco, significa che non supporta le autorizzazioni a livello di risorsa. Se un'azione AWS Batch API non supporta le autorizzazioni a livello di risorsa, puoi concedere agli utenti l'autorizzazione a utilizzare l'azione. Tuttavia, è necessario includere un carattere jolly (*) per l'elemento risorsa della dichiarazione politica.

Azioni

CancelJob, CreateComputeEnvironment, CreateJobQueue, CreateSchedulingPolicy, DeleteComputeEnvironment, DeleteJobQueue, DeleteSchedulingPolicy, DeregisterJobDefinition, ListTagsForResource, RegisterJobDefinition, SubmitJob, TagResource, TerminateJob, UntagResource, UpdateComputeEnvironment, UpdateSchedulingPolicy, UpdateJobQueue

CancelJob

Annulla un lavoro in coda. AWS Batch

Resource (Risorsa)

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

CreateComputeEnvironment

Crea un ambiente di AWS Batch calcolo.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

CreateJobQueue

Crea una coda AWS Batch di lavoro.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

DeleteComputeEnvironment

Elimina un ambiente di AWS Batch calcolo.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

CreateSchedulingPolicy

Crea una politica AWS Batch di pianificazione.

Resource (Risorsa)

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

DeleteJobQueue

Elimina la coda processi specificata. L'eliminazione della coda dei lavori alla fine elimina tutti i lavori in coda. I lavori vengono eliminati a una velocità di circa 16 lavori al secondo.

Resource (Risorsa)

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

aws:ResourceTag/${TagKey} (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

DeleteSchedulingPolicy

Elimina la politica di pianificazione specificata.

Resource (Risorsa)

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

aws:ResourceTag/${TagKey} (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

DeregisterJobDefinition

Annullamenti della registrazione e definizione del lavoroAWS Batch.

Resource (Risorsa)

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/ nome della definizione: revisione

Chiavi relative alle condizioni

aws:ResourceTag/${TagKey} (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

ListTagsForResource

Elenca i tag per la risorsa specificata.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/ nome della definizione: revisione

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

RegisterJobDefinition

Registri e AWS Batch definizioni.

Resource (Risorsa)

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/nome della definizione

Chiavi relative alle condizioni

`batch:AWSLogsCreateGroup`(Booleano)

Quando questo parametro è vero, `awslogs-group` viene creato per i log.

`batch:AWSLogsGroup` (stringa)

Il `awslogs` gruppo in cui si trovano i log.

`batch:AWSLogsRegion` (stringa)

La regione in cui vengono inviati i log.

`batch:AWSLogsStreamPrefix` (stringa)

Il prefisso del flusso di `awslogs` log.

`batch:Image` (stringa)

L'immagine Docker utilizzata per avviare un processo.

`batch:LogDriver` (stringa)

Il driver di registro utilizzato per il lavoro.

`batch:Privileged`(Booleano)

Quando questo parametro è vero, al contenitore del processo vengono concesse autorizzazioni elevate sull'istanza del contenitore host.

`batch:User` (stringa)

Il nome utente o l'uid numerico da utilizzare all'interno del contenitore per il lavoro.

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

SubmitJob

Invia un AWS Batch lavoro da una definizione di lavoro.

Resource (Risorsa)

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/ nome-definizione [: revisione]

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

> **Note**
>
> Questa chiave può essere utilizzata solo quando la definizione del processo Amazon Resource Name (ARN) è nel formato `arn:aws:batch:`*`region`*`:`*`account_number`*`:job-definition/`*`definition-name`*`:`*`revision`*`.`

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

TagResource

Associa i tag alle risorse specificate.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/ nome della definizione: revisione

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

TerminateJob

Termina un lavoro in una coda di AWS Batch lavoro.

Resource (Risorsa)

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

UntagResource

Deseleziona la risorsa specificata.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Processo

arn:aws:batch:regione: account:job/ JoBid

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Definizione del Job

arn:aws:batch: regione: account: definizione del lavoro/ nome della definizione: revisione

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

UpdateComputeEnvironment

Aggiorna un ambiente di AWS Batch calcolo.

Resource (Risorsa)

Ambiente di calcolo

arn:aws:batch: region: account:compute-environment/ compute-environment-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

UpdateJobQueue

Aggiorna una coda di processi.

Resource (Risorsa)

Job Queue

arn:aws:batch: regione: account:job-queue/ nome-coda

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

UpdateSchedulingPolicy

Aggiorna una politica di pianificazione.

Resource (Risorsa)

Politica di pianificazione

arn:aws:batch: region: account:scheduling-policy/ scheduling-policy-name

Chiavi relative alle condizioni

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

Chiavi di condizione per le azioni AWS Batch API

AWS Batchdefinisce le seguenti chiavi di condizione utilizzate nell'`Condition`elemento di una policy IAM. È possibile utilizzare queste chiavi per perfezionare le condizioni a cui si applica la dichiarazione politica. Per visualizzare le chiavi di condizione globali disponibili per tutti i servizi, consulta le chiavi di condizione globali disponibili nella Guida per l'utente IAM.

`batch:AWSLogsCreateGroup`(Booleano)

Quando questo parametro è vero, `awslogs-group` viene creato per i log.

`batch:AWSLogsGroup` (stringa)

Il `awslogs` gruppo in cui si trovano i log.

`batch:AWSLogsRegion` (stringa)

Il Regione AWS luogo in cui vengono inviati i log.

`batch:AWSLogsStreamPrefix` (stringa)

Il prefisso del `awslogs` log stream.

`batch:Image` (stringa)

L'immagine Docker utilizzata per avviare un processo.

`batch:LogDriver` (stringa)

Il driver di registro utilizzato per il lavoro.

`batch:Privileged`(Booleano)

Quando questo parametro è vero, al contenitore del processo vengono concesse autorizzazioni elevate sull'istanza del contenitore host (analogamente all'utente root).

`aws:ResourceTag/${TagKey}` (stringa)

Filtra le azioni in base ai tag associati alla risorsa.

`aws:RequestTag/${TagKey}` (stringa)

Filtra le operazioni in base ai tag passati nella richiesta

`batch:ShareIdentifier` (stringa)

Filtra le azioni in base al `shareIdentifier` parametro inviato a SubmitJob.

`aws:TagKeys` (stringa)

Filtra le operazioni in base alle chiavi di tag passate nella richiesta

`batch:User` (stringa)

Il nome utente o l'ID utente numerico (uid) da utilizzare all'interno del contenitore per il lavoro.

Policy di esempio

Gli esempi seguenti mostrano le dichiarazioni politiche che è possibile utilizzare per controllare le autorizzazioni di cui dispongono gli utenti. AWS Batch

Esempi

- Accesso in sola lettura
- Limitazione all'utente POSIX, all'immagine Docker, al livello di privilegio e al ruolo nell'invio del lavoro
- Limita al prefisso di definizione del lavoro all'invio del lavoro
- Limita alla coda dei lavori
- Nega l'azione quando tutte le condizioni corrispondono alle stringhe
- Nega l'azione quando i tasti condizionali corrispondono alle stringhe
- Usa la chiave batch:ShareIdentifier condizionale

Accesso in sola lettura

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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:Describe*",
                "batch:List*"
            ],
            "Resource": "*"
        }
    ]
}
```

Limitazione 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 A_. JobDef

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 processi 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.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "batch:RegisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
            ],
            "Condition": {
                "StringEquals": {
                    "batch:User": [
                        "nobody"
                    ],
                    "batch:Image": [
                        "<aws_account_id>.dkr.ecr.<aws_region>.amazonaws.com/myImage"
                    ]
                },
                "Bool": {
                    "batch:Privileged": "false"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "batch:DeregisterJobDefinition"
            ],
            "Resource": [
                "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iam:PassRole"
            ],
            "Resource": [
                "arn:aws:iam::<aws_account_id>:role/MyBatchJobRole"
            ]
```

```
        }
    ]
}
```

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 A. JobDef

> **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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/JobDefA_*",
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-queue/*"
      ]
    }
  ]
}
```

Limita alla coda dei lavori

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws:batch:<aws_region>:<aws_account_id>:job-definition/*",
        "arn:aws:batch:<aws_region>:<aws_account_id>: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 delle condizioni 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 perAWS Batch, vedi Condition keys for AWS Batch nel Service Authorization Reference. Ad eccezione dibatch: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.

```
{
  "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"
        }
      }
    }
  ]
}
```

Nega l'azione quando i tasti condizionali corrispondono alle stringhe

La seguente politica nega l'accesso al funzionamento dell'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'Denyeffetto non viene applicato a quel contenitore. Piuttosto, un contenitore diverso nello stesso lavoro potrebbe essere negato.

Per l'elenco delle chiavi di condizione perAWS Batch, vedi Condition keys for AWS Batch nel Service Authorization Reference. Ad eccezione dibatch: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.)

```
{
  "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"
          ]
        }
      }
    }
  ]
}
```

Usa la chiave **batch:ShareIdentifier** condizionale

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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
      ],
      "Resource": [
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-definition/JobDefA",
        "arn:aws::batch:<aws_region>:<aws_account_id>:job-queue/jobqueue1"
      ],
      "Condition": {
        "StringEquals": {
          "batch:ShareIdentifier": [
            "lowCpu"
          ]
        }
      }
    }
  ]
}
```

Policy gestita di AWS Batch

AWS Batchfornisce una policy gestita che è possibile allegare agli utenti che fornisce l'autorizzazione all'uso di 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 BatchAPI Reference.

AWSBatchFullAccess

Questa policy consente l'accesso completo come amministratore a AWS Batch.

```
{
```

```
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
    },
    {
```

```
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
          }
      }
    }
  ]
}
```

Creazione di AWS Batch policy IAM

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.

Ruolo dell'istanza Amazon ECS

AWS Batchgli 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 il ruolo IAM dell'istanza di container Amazon ECS nella Amazon Elastic Container Service Developer Guide.

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.

Come verificare 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 di ruoli per `ecsInstanceRole`. Se il ruolo non esiste, utilizza i seguenti passaggi per creare il ruolo.
   a. Selezionare Crea ruolo.
   b. Per Trusted entity type (Tipo di entità attendibile), scegli Servizio AWS.
   c. Per i casi d'uso comuni, scegli EC2.
   d. Seleziona Avanti.
   e. Per le politiche di autorizzazione, cerca ContainerServiceforAmazonEC2 EC2Role.
   f. Seleziona la casella di controllo accanto a ContainerServiceforAmazonEC2 EC2Role, quindi scegli Avanti.
   g. Per Role Name (Nome ruolo), digita `ecsInstanceRole`, quindi scegli Create Role (Crea ruolo).

   > **Note**
   >
   > Se utilizzi il per AWS Management Console 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.

Come creare un ruolo e il profilo dell'istanza IAM (AWS CLI)

1. Crea la seguente politica di fiducia e salvala in un file di testo denominato`ecsInstanceRole-role-trust-policy.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
```

Di seguito è riportata una risposta di esempio.

```
{
    "Role": {
         "Path": "/",
         "RoleName: "ecsInstanceRole",
         "RoleId": "AROAT46P5RDIY4EXAMPLE",
         "Arn": "arn:aws:iam::123456789012:role/ecsInstanceRole".
        "CreateDate": "2022-12-12T23:46:37.247Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service: "ec2.amazonaws.com"
                    }
                    "Action": "sts:AssumeRole",
                 }
              ]
          }
        }
```

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'AWSAPI 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. Utilizzare il attach-role-policycomando per allegare la politica `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 spot di Amazon EC2

Se crei un ambiente di elaborazione gestito che utilizza istanze Spot Fleet di Amazon EC2, devi creare la policy. `AmazonEC2SpotFleetTaggingRole` 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 AWSServiceRoleForEC2SpotFleetcollegati ai servizi per Amazon EC2 Spot AWSServiceRoleForEC2Spote 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 spot di Amazon EC2 nel AWS Management Console
- Crea ruoli per la flotta Spot di Amazon EC2 con AWS CLI

Crea ruoli per la flotta spot di Amazon EC2 nel AWS Management Console

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, scegli EC2Servizi AWS, quindi scegli EC2 - Spot Fleet Tagging.
6. Seleziona Avanti.
7. Da Politiche di autorizzazione per il nome della politica, verifica. `AmazonEC2SpotFleetTaggingRole`
8. Seleziona Avanti.
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 policy gestite per il ruolo della flotta Spot di Amazon EC2.
>
> - AmazonEC2 SpotFleetRole: questa è la politica gestita originale per il ruolo Spot Fleet. Tuttavia, non è più consigliabile 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.
> - AmazonEC2 SpotFleetTaggingRole: questo ruolo fornisce tutte le autorizzazioni necessarie per etichettare le istanze Spot di Amazon EC2. Utilizzare questo ruolo per consentire il tagging delle istanze Spot negli ambienti di calcolo AWS Batch.

Crea ruoli per la flotta Spot di Amazon EC2 con AWS CLI

Per creare il ruolo SpotFleetTaggingRole IAM di AmazonEC2 per gli ambienti di calcolo 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 IAM SpotFleetTaggingRole gestita da AmazonEC2 al tuo SpotFleetTaggingRole ruolo AmazonEC2, esegui il comando seguente 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 EC2 Spot

> **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 Batchi 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.

```
{
  "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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "batch:SubmitJob"
       ],
      "Resource": "*"
    }
  ]
}
```

AWS Batch Stream 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

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 processo

Ogni volta che un job esistente (inviato in precedenza) cambia stato, viene creato un evento. Per ulteriori informazioni sugli stati AWS Batch lavorativi, vedereStati del processo.

> **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": []
        },
        "tags": {
```

```
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "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 di coda bloccate supportate, consulta esempi di messaggi di coda di lavoro bloccati. Lo stesso motivo è disponibile anche nel `statusReason` campo dell'azione `DescribeJobs` API.

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 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": []
        },
        "tags": {
            "resourceArn": "arn:aws:batch:us-
east-1:123456789012:job/4c7599ae-0a82-49aa-ba5a-4727fcce14a8"
        },
        "propagateTags": false,
        "platformCapabilities": []
    }
}
```

Utilizzo delle 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. È possibile ricevere notifiche per gli eventi tramite più canali, tra cui e-mail, notifiche chat AWS Chatbot o notifiche push AWS Console Mobile Application. 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 Spot di Amazon EC2 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.

Creazione di un lavoro pianificato AWS Batch

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. Seleziona Avanti.
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 Array.

- 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, specifica l'intervallo di tempo per cui vengono conservati gli eventi non elaborati.

    c. Per Riprovare, inserisci 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. Seleziona Avanti.
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. Seleziona Avanti.
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.

Creazione di una regola con uno schema di eventi

La procedura seguente illustra come creare una regola con un pattern 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. Seleziona Avanti.
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. Seleziona Avanti.
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 Array.
    - 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, inserisci 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. Seleziona Avanti.
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. Seleziona Avanti.
24. Per Revisione e creazione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Al termine, 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.

Trasmissione delle informazioni sugli eventi a un AWS Batch Target in base a una pianificazione utilizzando il trasformatore EventBridge di input

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. Seleziona Avanti.
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 Array.
- 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, inserisci 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. Seleziona Avanti.
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. Seleziona Avanti.
30. Per Revisione e creazione, consulta i passaggi di configurazione. Se devi apportare modifiche, seleziona Edit (Modifica). Al termine, scegli Crea regola.

Tutorial: Listening for AWS Batch 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 con AWS Batch 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.

Fase 1: 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 console AWS Lambda 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, modificate il codice di esempio in modo che corrisponda all'esempio seguente:

```
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).

Fase 2: registrazione di una regola di evento

In questa sezione, crei una regola di EventBridge evento che acquisisce gli eventi di lavoro provenienti dalle tue risorse. AWS Batch 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 AWS Management Console 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 ilAWS 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 servizio AWS nell'account emette un evento, passa sempre al bus di eventi predefinito dell'account.
6. Per Rule type (Tipo di regola), scegli Rule with an event pattern (Regola con un modello di eventi).
7. Seleziona Avanti.
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. Seleziona Avanti.
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 Select an Amazon SQS queue in the current AWS account to use as the dead-letter queue (Seleziona una coda Amazon SQS nell'account corrente da utilizzare come coda DLQ), quindi seleziona la coda da utilizzare dal menu a discesa.
    - Scegli Seleziona una coda Amazon SQS in un altro account AWS come coda DLQ e specifica l'ARN della coda da utilizzare. È necessario allegare alla coda una politica basata sulle risorse che conceda l'autorizzazione a inviare messaggi alla coda. EventBridge Per ulteriori informazioni, consulta Concessione delle autorizzazioni alla coda delle lettere non scritte nella Amazon User Guide. EventBridge

15. Seleziona Avanti.
16. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
17. Seleziona Avanti.
18. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Fase 3: testa la configurazione

Ora puoi testare la tua EventBridge configurazione inviando un lavoro alla tua coda lavori. 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 AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Invia un nuovo processo di AWS Batch. Per ulteriori informazioni, consulta Invio di un lavoro.
3. Apri la CloudWatch console all'indirizzo https://console.aws.amazon.com/cloudwatch/.
4. Nel pannello di navigazione, scegli Log, quindi seleziona il gruppo di log della funzione Lambda (ad esempio, /aws/lambda/*my-function*).
5. Seleziona un flusso di log per visualizzare i dati di evento.

Tutorial: invio di avvisi Amazon Simple Notification Service per eventi Job non riusciti

In questo tutorial, configuri una regola di EventBridge evento che acquisisce solo gli eventi del processo in cui il lavoro ha raggiunto 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 con AWS Batch

Fase 1: creare e sottoscrivere un argomento 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 all'indirizzo https://console.aws.amazon.com/sns/v3/home.
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).

Fase 2: registrazione di 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. Seleziona Avanti.
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. Seleziona Avanti.
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. Seleziona Avanti.
16. (Facoltativo) Inserire uno o più tag per la regola. Per ulteriori informazioni, consulta i EventBridge tag Amazon nella Amazon EventBridge User Guide.
17. Seleziona Avanti.
18. Rivedi i dettagli della regola e scegli Create rule (Crea regola).

Fase 3: Test del tuo articolo

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 con il testo dell'evento entro pochi minuti.

Verifica di una regola

1. Apri la AWS Batch console all'indirizzo https://console.aws.amazon.com/batch/.
2. Invia una nuova AWS Batch offerta di lavoro. Per ulteriori informazioni, consulta Invio di 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: Batch Job Queue Bloccata

Per creare una regola evento che controlli Batch Job Queue Blocked, ripeti i passaggi di questo tutorial con le seguenti modifiche:

1. Nel passaggio 1, utilizzare *BlockedJobQueue*come nome dell'argomento.
2. Nel passaggio 2, usa il seguente schema nell'editor JSON:

   ```
   {
      "detail-type": [
   ```

```
      "Batch Job Queue Blocked"
    ],
    "source": [
      "aws.batch"
    ]
}
```

Utilizzo dei CloudWatch registri con AWS Batch

Puoi configurare i tuoi AWS Batch lavori sulle risorse EC2 per inviare informazioni e metriche dettagliate sui log ai log. CloudWatch In questo modo, puoi visualizzare diversi log 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 su risorse EC2, aggiungi una policy IAM al ruolo. `ecsInstanceRole` Per ulteriori informazioni, consulta the section called “Aggiungi una policy CloudWatch Logs IAM”.
- Crea un modello di lancio di Amazon EC2 che includa un CloudWatch monitoraggio dettagliato, quindi specifica il modello quando crei il tuo ambiente di AWS Batch calcolo. 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. Puoi aggiungere parametri che modificano il comportamento predefinito sulle risorse EC2 e Fargate. Per ulteriori informazioni, consulta the section called “Utilizzo del driver di log awslogs”.

Aggiungi una policy CloudWatch Logs IAM

Prima che i tuoi job possano inviare dati di log e metriche dettagliate a CloudWatch Logs, devi creare una policy IAM che utilizzi le CloudWatch API Logs. 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 risorse Amazon EC2. 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 pannello di navigazione, selezionare Policies (Policy).
3. Scegli Create Policy (Crea policy).
4. Scegli JSON, quindi inserisci la seguente politica:

```
{
    "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 CloudWatchECS-Logs, quindi scegli Allega policy.

Installa e configura l'agente CloudWatch

Puoi creare un modello di lancio di Amazon EC2 che CloudWatch includa 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'AMI Amazon EC2 esistente e quindi specificare l'immagine nella procedura guidata di AWS Batch prima esecuzione. Per ulteriori informazioni, consulta Installazione dell' CloudWatch agente e Guida introduttiva. AWS Batch

> **Note**
>
> I modelli di avvio non sono supportati nelle AWS Fargate risorse.

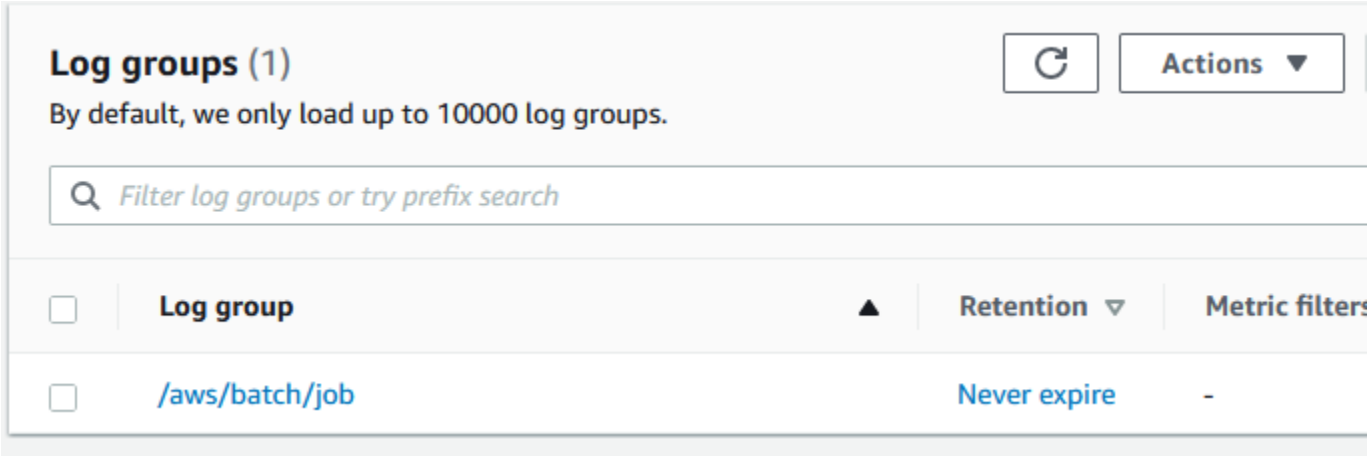
Visualizza i CloudWatch registri

È possibile visualizzare e cercare CloudWatch i registri dei registri in. AWS Management Console

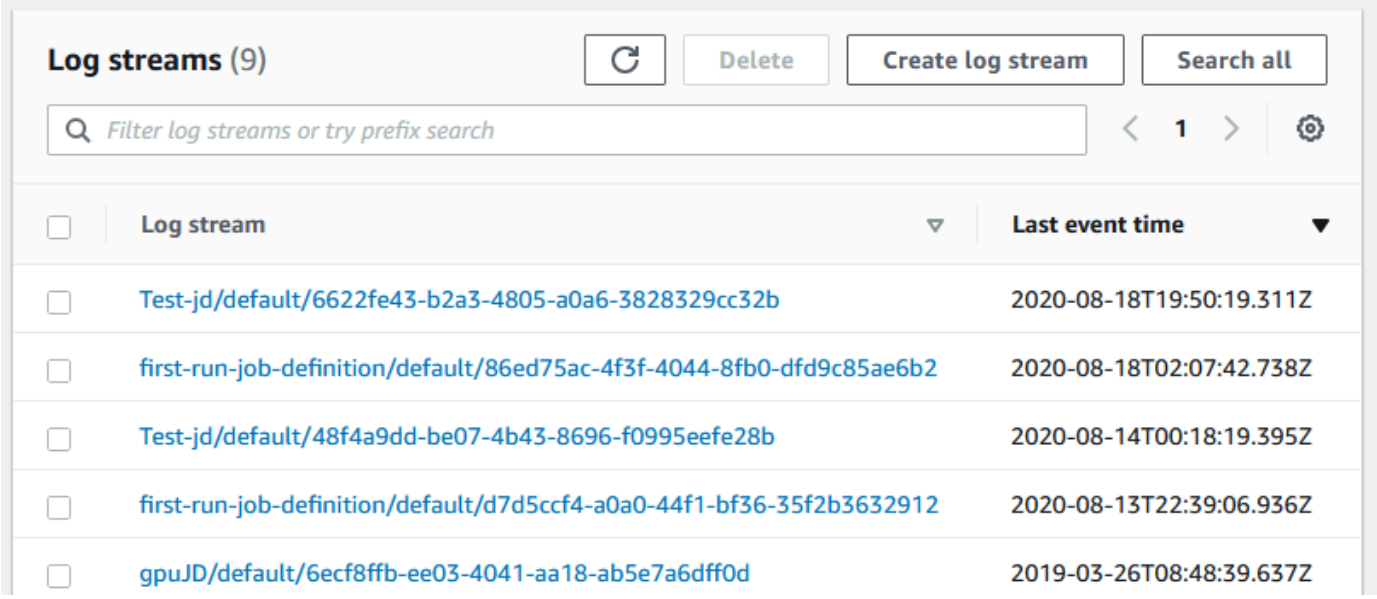
> **Note**
>
> La visualizzazione dei dati nei registri potrebbe richiedere alcuni minuti. CloudWatch

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.

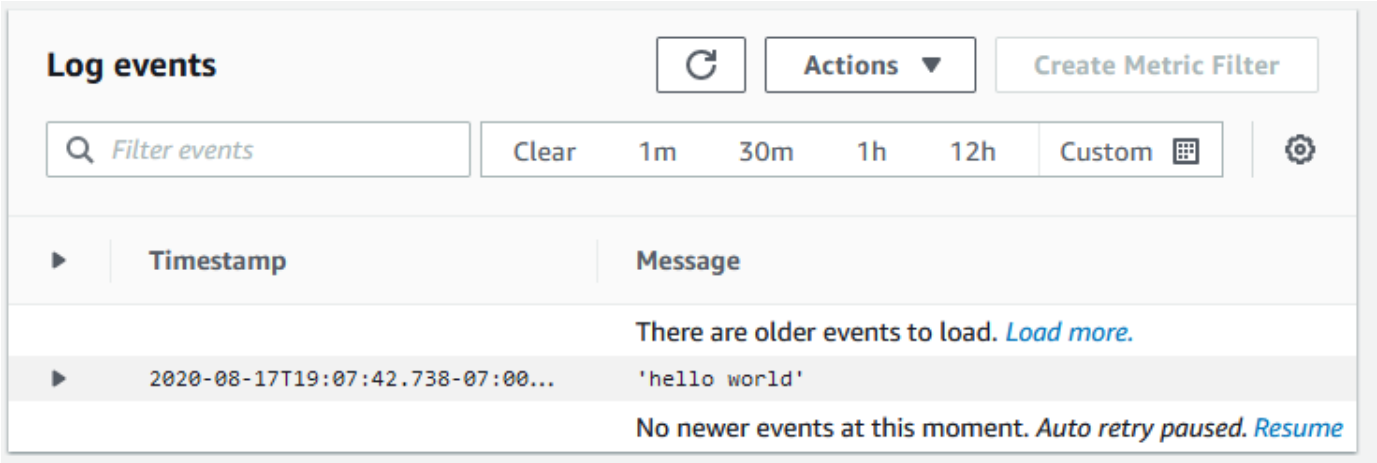


3. Scegli un gruppo di log da visualizzare.



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.



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.

Puoi scaricare un'Fluent Bitimmagine AWS for che include un plug-in per il monitoraggio dei lavori AWS Batch Amazon EKS in CloudWatch Logs. Fluent Bitè un elaboratore di log e forwarder open source che è sia Docker che compatibile. Kubernetes Ti consigliamo di utilizzarlo Fluent Bit come router di registro perché richiede meno risorse rispetto a. Fluentd Per ulteriori informazioni, consulta Uso dell'immagine AWS for Fluent Bit.

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 AWS per Fluent Bit

Per istruzioni su come installare AWS Fluent Bit e creare i CloudWatch gruppi, vedi Configurazione Fluent Bit o Avvio rapido con l' CloudWatchagente e Fluent Bit.

> **Tip**
>
> Ricorda che Fluent Bit utilizza 1,5 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.

Attiva Fluent Bit per i nodi AWS Batch

Per garantire che la Fluent Bit registrazione venga DaemonSet eseguita sui nodi AWS Batch gestiti, modifica le Fluent Bit DaemonSet tolleranze:

```
tolerations:
```

```
- key: "batch.amazonaws.com/batch-node"
  operator: "Exists"
```

AWS Batch CloudWatch Informazioni approfondite sui container

CloudWatch Container Insights raccoglie, aggrega e riepiloga metriche e log dagli ambienti di elaborazione e dai lavori. 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

Attiva Container Insights

Puoi attivare Container Insights per gli ambienti di AWS Batch elaborazione.

1. Apri la AWS Batch console.
2. Scegli Ambienti di calcolo.
3. Scegli l'ambiente di elaborazione che desideri.
4. Per Container Insights, attiva Container Insights per l'elaborazione ambiente.

> **Tip**
>
> Puoi selezionare un intervallo predefinito per aggregare le metriche o crearne uno personalizzato intervallo.

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 processi in 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.

Registrazione di log delle chiamate API di AWS Batch con AWS CloudTrail

AWS Batchè integrato conAWS CloudTrail, un servizio che fornisce un registro delle azioni intraprese da un utente, un ruolo o un AWS servizio inAWS Batch. CloudTrail acquisisce tutte le chiamate API AWS Batch come eventi. Le chiamate acquisite includono le chiamate dalla console di AWS Batch e le chiamate di codice alle operazioni delle API AWS Batch. 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, puoi determinare a quale richiesta è stata inviataAWS Batch, l'indirizzo IP da cui è stata effettuata, chi ha effettuato la richiesta, quando è stata effettuata e dettagli aggiuntivi.

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

Informazioni di AWS Batch in CloudTrail

CloudTrail è abilitato sul tuo AWS account al momento della creazione dell'account. Quando si verifica un'attività inAWS Batch, tale attività viene registrata in un CloudTrail evento insieme ad altri eventi AWS di servizio nella cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, vedere Visualizzazione degli eventi con la cronologia degli CloudTrail eventi.

Per una registrazione continua degli eventi nell'account AWS che includa gli eventi per AWS Batch, crea un trail. Un trail consente di CloudTrail inviare file di log a un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni AWS. Il trail registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri AWS servizi per analizzare ulteriormente e agire in base ai dati sugli eventi raccolti nei CloudTrail log. 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 AWS Batch le azioni vengono registrate CloudTrail e documentate nel https://docs.aws.amazon.com/batch/latest/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.

Informazioni sulle voci dei file di log di AWS Batch

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"
}
```

Creazione di 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 seguente tabella per determinare quali opzioni selezionare.

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

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

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 calcolo gestito di AWS Batch che avvia risorse di calcolo nel nuovo VPC. Per ulteriori informazioni, consulta Creazione di 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 Creazione di una coda di lavoro.
- Crea una definizione per l'esecuzione dei processi. Per ulteriori informazioni, consulta Creazione di 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 Invio di un lavoro.

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 e te. AWS Il modello di responsabilità condivisa descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- Sicurezza del cloud: AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori 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. I seguenti argomenti mostrano 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

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

Il modo in cui usi AWS Identity and Access Management (IAM) varia a seconda del lavoro che AWS Batch svolgi.

Utente del servizio: se utilizzi il AWS Batch servizio per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Man mano che utilizzi più AWS Batch funzionalità per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS Batch, consulta Risoluzione dei problemi di AWS Batch identità e accesso.

Amministratore del servizio: se sei responsabile delle AWS Batch risorse della tua azienda, probabilmente hai pieno accesso a AWS Batch. È tuo compito determinare a quali AWS Batch funzionalità e risorse devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS Batch, consultaCome AWS Batch funziona con IAM.

Amministratore IAM: un amministratore IAM potrebbe essere interessato a ottenere dei dettagli su come scrivere policy per gestire l'accesso a AWS Batch. Per visualizzare esempi di policy AWS Batch basate sull'identità che puoi utilizzare in IAM, 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 essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente.

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta Signing AWS API request nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta Autenticazione a più fattori nella Guida per l'utente di AWS IAM Identity Center e Utilizzo dell'autenticazione a più fattori (MFA) in AWSnella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per

creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione Attività che richiedono le credenziali dell'utente root nella Guida per l'utente IAM.

Identità federata

Come procedura consigliata, richiedi agli utenti umani, compresi gli utenti che richiedono l'accesso come amministratore, di utilizzare la federazione con un provider di identità per accedere Servizi AWS utilizzando credenziali temporanee.

Un'identità federata è un utente dell'elenco utenti aziendale, di un provider di identità Web AWS Directory Service, della directory Identity Center o di qualsiasi utente che accede utilizzando le Servizi AWS credenziali fornite tramite un'origine di identità. Quando le identità federate accedono Account AWS, assumono ruoli e i ruoli forniscono credenziali temporanee.

Per la gestione centralizzata degli accessi, consigliamo di utilizzare AWS IAM Identity Center. Puoi creare utenti e gruppi in IAM Identity Center oppure puoi connetterti e sincronizzarti con un set di utenti e gruppi nella tua fonte di identità per utilizzarli su tutte le tue applicazioni. Account AWS Per ulteriori informazioni su IAM Identity Center, consulta Cos'è IAM Identity Center? nella Guida per l'utente di AWS IAM Identity Center .

Utenti e gruppi IAM

Un utente IAM è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine nella Guida per l'utente IAM.

Un gruppo IAM è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli

utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta Quando creare un utente IAM (invece di un ruolo) nella Guida per l'utente IAM.

Ruoli IAM

Un ruolo IAM è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console cambiando ruolo. Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta Utilizzo di ruoli IAM nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- Accesso utente federato: per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta Creazione di un ruolo per un provider di identità di terza parte nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta Set di autorizzazioni nella Guida per l'utente di AWS IAM Identity Center .
- Autorizzazioni utente IAM temporanee: un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- Accesso multi-account: è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente IAM.
- Accesso a più servizi: alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.

- Sessioni di accesso diretto (FAS): quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.
- Ruolo di servizio: un ruolo di servizio è un ruolo IAM che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella Guida per l'utente IAM.
- Ruolo collegato al servizio: un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un. Servizio AWS Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

- Applicazioni in esecuzione su Amazon EC2: puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un AWS ruolo a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2 nella Guida per l'utente IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta Quando creare un ruolo IAM (invece di un utente) nella Guida per l'utente IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La

maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta Panoramica delle policy JSON nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall'o dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta Scelta fra policy gestite e policy inline nella Guida per l'utente IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario specificare un principale in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o. Servizi AWS

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta Panoramica delle liste di controllo degli accessi (ACL) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- Limiti delle autorizzazioni: un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal`sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta Limiti delle autorizzazioni per le entità IAM nella Guida per l'utente IAM.
- Politiche di controllo dei servizi (SCP): le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in. AWS Organizations AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle Policy di controllo dei servizi nella Guida per l'utente di AWS Organizations .
- Policy di sessione: le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire

da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta Policy di sessione nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta Logica di valutazione delle policy nella IAM User Guide.

Come AWS Batch funziona con IAM

Prima di utilizzare IAM per gestire l'accesso a AWS Batch, scopri con quali funzionalità IAM è disponibile l'uso AWS Batch.

Funzionalità IAM che puoi utilizzare con AWS Batch

Funzionalità IAM	AWS Batch supporto
Policy basate su identità	Sì
Policy basate su risorse	No
Azioni di policy	Sì
Risorse relative alle policy	Sì
Chiavi di condizione delle policy	Sì
Liste di controllo degli accessi (ACL)	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 su identità	Sì

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta Creazione di policy IAM nella Guida per l'utente IAM.

Con le policy basate su identità di IAM, è possibile specificare quali operazioni e risorse sono consentite o respinte, nonché le condizioni in base alle quali le operazioni sono consentite o respinte. Non è possibile specificare l'entità principale in una policy basata sull'identità perché si applica all'utente o al ruolo a cui è associato. Per informazioni su tutti gli elementi utilizzabili in una policy JSON, consulta Guida di riferimento agli elementi delle policy JSON IAM nella Guida per l'utente di IAM.

Esempi di policy basate su identità per 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	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action`di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Per visualizzare un elenco di 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 di policy	Sì

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource`della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource`o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo nome della risorsa Amazon (ARN). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

```
"Resource": "*"
```

Per visualizzare un elenco dei tipi di AWS Batch risorse e dei relativi ARN, consulta 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.

Per visualizzare esempi di politiche AWS Batch basate sull'identità, vedere. Esempi di policy basate sull'identità per 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. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Condition`(o blocco `Condition`) consente di specificare le condizioni in cui un'istruzione è in vigore. L'elemento `Condition`è facoltativo. Puoi compilare espressioni condizionali che utilizzano operatori di condizione, ad esempio uguale a o minore di, per soddisfare la condizione nella policy con i valori nella richiesta.

Se specifichi più elementi `Condition`in un'istruzione o più chiavi in un singolo elemento `Condition`, questi vengono valutati da AWS utilizzando un'operazione `AND`logica. Se si specificano più valori per una singola chiave di condizione, AWS valuta la condizione utilizzando un'operazione logica. `OR` Tutte le condizioni devono essere soddisfatte prima che le autorizzazioni dell'istruzione vengano concesse.

Puoi anche utilizzare variabili segnaposto quando specifichi le condizioni. Ad esempio, puoi autorizzare un utente IAM ad accedere a una risorsa solo se è stata taggata con il relativo nome utente IAM. Per ulteriori informazioni, consulta Elementi delle policy IAM: variabili e tag nella Guida per l'utente di IAM.

AWS supporta chiavi di condizione globali e chiavi di condizione specifiche del servizio. Per visualizzare tutte le chiavi di condizione AWS globali, consulta le chiavi di contesto delle condizioni AWS globali nella Guida per l'utente IAM.

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.

Per visualizzare esempi di politiche AWS Batch basate sull'identità, vedere. Esempi di policy basate sull'identità per AWS Batch

Controllo degli accessi basato sugli attributi (ABAC) con AWS Batch

Supporta ABAC (tag nelle policy)	Sì

Il controllo dell'accesso basato su attributi (ABAC) è una strategia di autorizzazione che definisce le autorizzazioni in base agli attributi. In AWS, questi attributi sono chiamati tag. Puoi allegare tag a entità IAM (utenti o ruoli) e a molte AWS risorse. L'assegnazione di tag alle entità e alle risorse è il primo passaggio di ABAC. In seguito, vengono progettate policy ABAC per consentire operazioni quando il tag dell'entità principale corrisponde al tag sulla risorsa a cui si sta provando ad accedere.

La strategia ABAC è utile in ambienti soggetti a una rapida crescita e aiuta in situazioni in cui la gestione delle policy diventa impegnativa.

Per controllare l'accesso basato su tag, fornisci informazioni sui tag nell'elemento condizione di una policy utilizzando le chiavi di condizione `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`o `aws:TagKeys`.

Se un servizio supporta tutte e tre le chiavi di condizione per ogni tipo di risorsa, il valore per il servizio è Yes (Sì). Se un servizio supporta tutte e tre le chiavi di condizione solo per alcuni tipi di risorsa, allora il valore sarà Parziale.

Per ulteriori informazioni su ABAC, consulta Che cos'è ABAC? nella Guida per l'utente IAM. Per visualizzare un tutorial con i passaggi per l'impostazione di ABAC, consulta Utilizzo del controllo degli accessi basato su attributi (ABAC) nella Guida per l'utente di IAM.

Utilizzo di credenziali temporanee con AWS Batch

Supporta le credenziali temporanee	Sì

Alcune Servizi AWS non funzionano quando si accede utilizzando credenziali temporanee. Per ulteriori informazioni, incluse quelle che Servizi AWS funzionano con credenziali temporanee, consulta la sezione relativa alla Servizi AWS compatibilità con IAM nella IAM User Guide.

Stai utilizzando credenziali temporanee se accedi AWS Management Console utilizzando qualsiasi metodo tranne nome utente e password. Ad esempio, quando accedete AWS utilizzando il link Single Sign-On (SSO) della vostra azienda, tale processo crea automaticamente credenziali temporanee. Le credenziali temporanee vengono create in automatico anche quando accedi alla console come utente e poi cambi ruolo. Per ulteriori informazioni sullo scambio dei ruoli, consulta Cambio di un ruolo (console) nella Guida per l'utente IAM.

È possibile creare manualmente credenziali temporanee utilizzando l'API or. AWS CLI AWS È quindi possibile utilizzare tali credenziali temporanee per accedere. AWS AWS consiglia di generare dinamicamente credenziali temporanee anziché utilizzare chiavi di accesso a lungo termine. Per ulteriori informazioni, consulta Credenziali di sicurezza provvisorie in IAM.

Autorizzazioni del principale tra servizi per AWS Batch

Supporta l'inoltro delle sessioni di accesso (FAS)	Sì

Quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un principale. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama an Servizio AWS, in combinazione con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina Forward access sessions.

Ruoli di servizio per 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 la sezione Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWSnella Guida per l'utente IAM.

> **Warning**
>
> La modifica delle autorizzazioni per un ruolo di servizio potrebbe compromettere 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'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati in Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.

Per 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.

AWS Batch esecuzione (ruolo 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 IAM di esecuzione 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.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ecr:GetAuthorizationToken",
        "ecr:BatchCheckLayerAvailability",
        "ecr:GetDownloadUrlForLayer",
        "ecr:BatchGetImage",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "*"
    }
  ]
}
```

Puoi utilizzare la seguente procedura per verificare che il tuo account abbia già il ruolo di esecuzione e per allegare la policy IAM gestita, se necessario.

Come verificare la presenza di **`ecsTaskExecutionRole`** 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 di ruoli per `ecsTaskExecutionRole`. Se non riesci a trovare il ruolo, consultaCreazione del ruolo IAM di esecuzione. Se hai trovato il ruolo, scegli il ruolo per visualizzare le politiche allegate.

4. Nella scheda Autorizzazioni, verifica che la policy gestita da TaskExecutionRolePolicyAmazonECS 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 AmazonECS TaskExecution RolePolicy.
   c. Seleziona la casella a sinistra della politica di TaskExecutionRolePolicyAmazonecS 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 riportata di seguito, il ruolo è configurato correttamente. Se la relazione di fiducia non corrisponde, scegli Modifica politica di fiducia, inserisci quanto segue e scegli Aggiorna politica.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "",
      "Effect": "Allow",
      "Principal": {
        "Service": "ecs-tasks.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Creazione del ruolo IAM di esecuzione

Se il tuo account non ha già un ruolo di esecuzione, utilizza i seguenti passaggi per creare il ruolo.

Per creare il ruolo **`ecsTaskExecutionRole`** IAM

1. Apri 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 EC2. Quindi scegli nuovamente EC2.
6. Seleziona Successivo.
7. Per le politiche di autorizzazione, cerca TaskExecutionRolePolicyAmazonecs.
8. Seleziona la casella di controllo a sinistra della politica di TaskExecutionRolePolicyAmazonecS, quindi scegli Avanti.
9. Per Nome ruolo, inserisci **`ecsTaskExecutionRole`** e quindi scegli Crea ruolo.

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 . Inoltre, non possono eseguire attività utilizzando AWS Management Console, AWS Command Line Interface (AWS CLI) o l'API. AWS Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta Creazione di policy IAM nella Guida per l'utente di IAM.

Per informazioni dettagliate sulle azioni e sui tipi di risorse definiti da AWS Batch, incluso il formato degli ARN per ciascun tipo di risorsa, consulta 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 crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche 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 ulteriori informazioni, consulta Policy gestite da AWSo Policy gestite da AWS per le funzioni dei processi nella Guida per l'utente IAM.

- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta Policy e autorizzazioni in IAM nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per ulteriori informazioni, consulta la sezione Elementi delle policy JSON di IAM: condizione nella Guida per l'utente IAM.
- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta Convalida delle policy per IAM Access Analyzer nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta Configurazione dell'accesso alle API protetto con MFA nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta Best practice di sicurezza in IAM nella Guida per l'utente di IAM.

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, concedi l'accesso solo alle operazioni 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 ulteriori informazioni, consulta Aggiunta di autorizzazioni a un utente nella Guida per l'utente 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 cpllegate 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": "*"
        }
    ]
}
```

Prevenzione del confused deputy tra servizi

Con "confused deputy" si intende un problema di sicurezza in cui un'entità che non dispone dell'autorizzazione per eseguire una certa operazione può costringere un'entità con più privilegi a eseguire tale operazione. Nel frattempo AWS, l'impersonificazione tra servizi 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 del tuo 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. Utilizza `aws:SourceAccount` se desideri 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 usare la chiave di contesto della condizione globale `aws:SourceArn`con l'ARN completo della risorsa. Se non conosci l'ARN completo della risorsa o scegli più risorse, utilizza 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 1: Ruolo per l'accesso a un solo ambiente di elaborazione

Il seguente ruolo 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.

```
{
  "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 2: 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.

```
{
  "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/*"
          ]
        }
      }
    }
  ]
}
```

Risoluzione dei problemi di AWS Batch identità e accesso

Utilizza le seguenti informazioni per aiutarti a diagnosticare e risolvere i problemi più comuni che potresti riscontrare quando lavori con un AWS Batch IAM.

Argomenti

- Non sono autorizzato a eseguire un'operazione in AWS Batch
- Non sono autorizzato a eseguire iam: PassRole
- Voglio consentire a persone esterne al mio AWS account di accedere alle mie AWS Batch risorse

Non sono autorizzato a eseguire un'operazione in AWS Batch

Se ti AWS Management Console 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 passare il ruolo al servizio.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

In questo caso, le policy di Mary devono essere aggiornate per poter eseguire l'operazione `iam:PassRole`.

Se hai bisogno di aiuto, contatta il tuo AWS amministratore. L'amministratore è la persona che ti ha fornito le credenziali di accesso.

Voglio consentire a persone esterne 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 servizi che supportano policy basate su risorse o liste di controllo degli accessi (ACL), utilizza tali policy per concedere alle persone l'accesso alle tue risorse.

Per ulteriori informazioni, consulta gli argomenti seguenti:

- Per sapere se AWS Batch supporta queste funzionalità, consultaCome 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 tra l'utilizzo di ruoli e policy basate su risorse per l'accesso multi-account, consultare Differenza tra i ruoli IAM e le policy basate su risorse nella Guida per l'utente di IAM.

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

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 collegata a nessun'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 the section called “Nome ruolo o ARN errati” la Guida AWS Batch per l'utente.

È possibile eliminare un ruolo collegato ai servizi solo dopo aver eliminato le risorse correlate. Questa procedura protegge le risorse di AWS Batch perché impedisce la rimozione involontaria delle autorizzazioni di accesso alle risorse.

Per informazioni sugli altri servizi che supportano i ruoli collegati ai servizi, consultare Servizi AWS che funzionano con IAM e cercare i servizi che riportano Sì nella colonna Ruolo collegato ai servizi. Scegli Sì in corrispondenza di un link per visualizzare la documentazione relativa al ruolo collegato ai servizi per tale servizio.

Autorizzazioni di ruolo collegate al servizio per AWS Batch

AWS Batch utilizza il ruolo collegato al servizio denominato. AWSServiceRoleForBatch Il AWSServiceRoleForBatchruolo consente di AWS Batch creare e gestire AWS risorse per conto dell'utente.

Il ruolo AWSServiceRoleForBatchcollegato al servizio prevede che il responsabile del `batch.amazonaws.com` servizio assuma il ruolo.

La policy IAM denominata BatchServiceRolePolicyconsente di AWS Batch 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 istanze Amazon EC2, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste di EC2 Spot Fleet per alcuni ambienti di calcolo Spot EC2.

- `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 creare e gestire gruppi AWS Batch di log e flussi di log per i lavori. AWS Batch

Per consentire a un'entità IAM (come un utente, un gruppo o un ruolo) di creare, modificare o eliminare un ruolo collegato ai servizi devi configurare le relative autorizzazioni. Per ulteriori informazioni, consulta Autorizzazioni del ruolo collegato ai servizi nella Guida per l'utente di IAM.

Creazione di un ruolo collegato al servizio per AWS Batch

Non hai bisogno di creare manualmente un ruolo collegato ai servizi. Quando si CreateComputeEnvironment utilizza la AWS Management Console AWS CLI, o l' AWS API e non si specifica un valore per il `serviceRole` parametro, viene AWS Batch creato 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. Inoltre, 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 AWSServiceRoleForBatch ruolo nel tuo account. Per ulteriori informazioni, consulta Un nuovo ruolo è apparso nel mio account IAM.

Se elimini questo ruolo collegato ai servizi, puoi ricrearlo seguendo lo stesso processo utilizzato per ricreare il ruolo nell'account. Quando lo fai CreateComputeEnvironment, AWS Batch crea nuovamente il ruolo collegato al servizio per te.

Modifica di un ruolo collegato al servizio per AWS Batch

Con AWS Batch, non è possibile modificare il ruolo collegato al AWSServiceRoleForBatch servizio. Dopo aver creato un ruolo collegato al servizio, non puoi modificarne il nome, perché potrebbero farvi riferimento diverse entità. Puoi tuttavia modificarne la descrizione utilizzando IAM. Per ulteriori

informazioni, consulta la sezione Modifica di un ruolo collegato ai servizi nella Guida per l'utente di IAM.

Per consentire a un'entità IAM di modificare la descrizione del ruolo collegato al servizio AWSServiceRoleForBatch

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 al servizio per AWS Batch

Se non è più necessario utilizzare una funzionalità o un servizio che richiede un ruolo collegato al servizio, si consiglia di eliminare tale ruolo. In questo modo, non hai un'entità inutilizzata che non viene monitorata o gestita attivamente. Tuttavia, è necessario effettuare la pulizia delle risorse associate al ruolo collegato al servizio prima di poterlo eliminare manualmente.

Per consentire a un'entità IAM di eliminare il ruolo collegato al AWSServiceRoleForBatch servizio

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 AWSServiceRoleForBatch 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 AWSServiceRoleForBatch 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 AWSServiceRoleForBatch servizio

È necessario eliminare tutti gli ambienti di AWS Batch elaborazione che utilizzano il AWSServiceRoleForBatch ruolo in tutte le AWS regioni prima di poter eliminare il ruolo. AWSServiceRoleForBatch

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 AWS Management Console 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 cancellazione per un ruolo collegato al servizio, chiama. DeleteServiceLinkedRole Nella richiesta, specificate il nome del AWSServiceRoleForBatch 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. Per verificare lo stato dell'eliminazione, chiama GetServiceLinkedRoleDeletionStatus. 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 AWS Batch collegati ai servizi

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.

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, consulta 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 istanze Amazon EC2, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste di EC2 Spot Fleet per alcuni ambienti di calcolo Spot EC2.
- `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 creare e gestire gruppi AWS Batch di log e flussi di log per i lavori. AWS Batch

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

```
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:RequestSpotFleet",
                "autoscaling:DescribeAccountLimits",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "eks:DescribeCluster",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
                "ecs:DeregisterTaskDefinition",
                "ecs:TagResource",
                "ecs:ListAccountSettings",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
```

```
            "Resource": "*"
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogGroup",
                "logs:CreateLogStream"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*"
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": [
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/batch/job*:log-stream:*"
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateOrUpdateTags"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
```

```
                    "ecs-tasks.amazonaws.com"
                ]
            }
        }
    },
    {
        "Sid": "AWSBatchPolicyStatement6",
        "Effect": "Allow",
        "Action": "iam:CreateServiceLinkedRole",
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:AWSServiceName": [
                    "spot.amazonaws.com",
                    "spotfleet.amazonaws.com",
                    "autoscaling.amazonaws.com",
                    "ecs.amazonaws.com"
                ]
            }
        }
    },
    {
        "Sid": "AWSBatchPolicyStatement7",
        "Effect": "Allow",
        "Action": [
            "ec2:CreateLaunchTemplate"
        ],
        "Resource": "*",
        "Condition": {
            "Null": {
                "aws:RequestTag/AWSBatchServiceTag": "false"
            }
        }
    },
    {
        "Sid": "AWSBatchPolicyStatement8",
        "Effect": "Allow",
        "Action": [
            "ec2:TerminateInstances",
            "ec2:CancelSpotFleetRequests",
            "ec2:ModifySpotFleetRequest",
            "ec2:DeleteLaunchTemplate"
        ],
        "Resource": "*",
```

```
            "Condition": {
                "Null": {
                    "aws:ResourceTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement9",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:DeleteLaunchConfiguration"
            ],
            "Resource":
"arn:aws:autoscaling:*:*:launchConfiguration:*:launchConfigurationName/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement10",
            "Effect": "Allow",
            "Action": [
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup"
            ],
            "Resource":
"arn:aws:autoscaling:*:*:autoScalingGroup:*:autoScalingGroupName/AWSBatch*"
        },
        {
            "Sid": "AWSBatchPolicyStatement11",
            "Effect": "Allow",
            "Action": [
                "ecs:DeleteCluster",
                "ecs:DeregisterContainerInstance",
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:cluster/AWSBatch*"
        },
        {
```

```
            "Sid": "AWSBatchPolicyStatement12",
            "Effect": "Allow",
            "Action": [
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task-definition/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement13",
            "Effect": "Allow",
            "Action": [
                "ecs:StopTask"
            ],
            "Resource": "arn:aws:ecs:*:*:task/*/*"
        },
        {
            "Sid": "AWSBatchPolicyStatement14",
            "Effect": "Allow",
            "Action": [
                "ecs:CreateCluster",
                "ecs:RegisterTaskDefinition"
            ],
            "Resource": "*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement15",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": [
                "arn:aws:ec2:*::image/*",
                "arn:aws:ec2:*::snapshot/*",
                "arn:aws:ec2:*:*:subnet/*",
                "arn:aws:ec2:*:*:network-interface/*",
                "arn:aws:ec2:*:*:security-group/*",
                "arn:aws:ec2:*:*:volume/*",
                "arn:aws:ec2:*:*:key-pair/*",
                "arn:aws:ec2:*:*:launch-template/*",
```

```
                "arn:aws:ec2:*:*:placement-group/*",
                "arn:aws:ec2:*:*:capacity-reservation/*",
                "arn:aws:ec2:*:*:elastic-gpu/*",
                "arn:aws:elastic-inference:*:*:elastic-inference-accelerator/*",
                "arn:aws:resource-groups:*:*:group/*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement16",
            "Effect": "Allow",
            "Action": "ec2:RunInstances",
            "Resource": "arn:aws:ec2:*:*:instance/*",
            "Condition": {
                "Null": {
                    "aws:RequestTag/AWSBatchServiceTag": "false"
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement17",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": [
                        "RunInstances",
                        "CreateLaunchTemplate",
                        "RequestSpotFleet"
                    ]
                }
            }
        }
    ]
}
```

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 che prevedono la concessione del privilegio minimo, la policy AWSBatchServiceRolegestita può essere utilizzata come guida. Se una qualsiasi delle autorizzazioni concesse nella policy gestita non è necessaria per il caso d'uso, crea una policy personalizzata e aggiungi solo le autorizzazioni richieste. Questa politica e questo ruolo AWS Batch gestiti possono essere utilizzati con la maggior parte dei tipi di ambienti di elaborazione, ma l'utilizzo di ruoli collegati ai servizi è preferibile per un'esperienza gestita e con less-error-prone obiettivi più ampi.

- `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 istanze Amazon EC2, nonché di creare e gestire modelli e tag di avvio. AWS Batch crea e gestisce le richieste di EC2 Spot Fleet per alcuni ambienti di calcolo Spot EC2.
- `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 creare e gestire gruppi AWS Batch di log e flussi di log per i lavori. AWS Batch

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AWSBatchPolicyStatement1",
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeAccountAttributes",
                "ec2:DescribeInstances",
                "ec2:DescribeInstanceStatus",
                "ec2:DescribeInstanceAttribute",
                "ec2:DescribeSubnets",
```

```
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeKeyPairs",
                "ec2:DescribeImages",
                "ec2:DescribeImageAttribute",
                "ec2:DescribeSpotInstanceRequests",
                "ec2:DescribeSpotFleetInstances",
                "ec2:DescribeSpotFleetRequests",
                "ec2:DescribeSpotPriceHistory",
                "ec2:DescribeSpotFleetRequestHistory",
                "ec2:DescribeVpcClassicLink",
                "ec2:DescribeLaunchTemplateVersions",
                "ec2:CreateLaunchTemplate",
                "ec2:DeleteLaunchTemplate",
                "ec2:RequestSpotFleet",
                "ec2:CancelSpotFleetRequests",
                "ec2:ModifySpotFleetRequest",
                "ec2:TerminateInstances",
                "ec2:RunInstances",
                "autoscaling:DescribeAccountLimits",
                "autoscaling:DescribeAutoScalingGroups",
                "autoscaling:DescribeLaunchConfigurations",
                "autoscaling:DescribeAutoScalingInstances",
                "autoscaling:DescribeScalingActivities",
                "autoscaling:CreateLaunchConfiguration",
                "autoscaling:CreateAutoScalingGroup",
                "autoscaling:UpdateAutoScalingGroup",
                "autoscaling:SetDesiredCapacity",
                "autoscaling:DeleteLaunchConfiguration",
                "autoscaling:DeleteAutoScalingGroup",
                "autoscaling:CreateOrUpdateTags",
                "autoscaling:SuspendProcesses",
                "autoscaling:PutNotificationConfiguration",
                "autoscaling:TerminateInstanceInAutoScalingGroup",
                "ecs:DescribeClusters",
                "ecs:DescribeContainerInstances",
                "ecs:DescribeTaskDefinition",
                "ecs:DescribeTasks",
                "ecs:ListAccountSettings",
                "ecs:ListClusters",
                "ecs:ListContainerInstances",
                "ecs:ListTaskDefinitionFamilies",
                "ecs:ListTaskDefinitions",
                "ecs:ListTasks",
                "ecs:CreateCluster",
```

```
                "ecs:DeleteCluster",
                "ecs:RegisterTaskDefinition",
                "ecs:DeregisterTaskDefinition",
                "ecs:RunTask",
                "ecs:StartTask",
                "ecs:StopTask",
                "ecs:UpdateContainerAgent",
                "ecs:DeregisterContainerInstance",
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:PutLogEvents",
                "logs:DescribeLogGroups",
                "iam:GetInstanceProfile",
                "iam:GetRole"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AWSBatchPolicyStatement2",
            "Effect": "Allow",
            "Action": "ecs:TagResource",
            "Resource": [
                "arn:aws:ecs:*:*:task/*_Batch_*"
            ]
        },
        {
            "Sid": "AWSBatchPolicyStatement3",
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": [
                        "ec2.amazonaws.com",
                        "ec2.amazonaws.com.cn",
                        "ecs-tasks.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement4",
```

```
            "Effect": "Allow",
            "Action": "iam:CreateServiceLinkedRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:AWSServiceName": [
                        "spot.amazonaws.com",
                        "spotfleet.amazonaws.com",
                        "autoscaling.amazonaws.com",
                        "ecs.amazonaws.com"
                    ]
                }
            }
        },
        {
            "Sid": "AWSBatchPolicyStatement5",
            "Effect": "Allow",
            "Action": [
                "ec2:CreateTags"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "StringEquals": {
                    "ec2:CreateAction": "RunInstances"
                }
            }
        }
    ]
}
```

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, Amazon CloudWatch EKS e IAM. In questo modo le identità IAM, utenti o ruoli, possono visualizzare le risorse AWS Batch gestite che sono state 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.

```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Effect":"Allow",
      "Action":[
        "batch:*",
        "cloudwatch:GetMetricStatistics",
        "ec2:DescribeSubnets",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeVpcs",
        "ec2:DescribeImages",
        "ec2:DescribeLaunchTemplates",
        "ec2:DescribeLaunchTemplateVersions",
        "ecs:DescribeClusters",
        "ecs:Describe*",
        "ecs:List*",
        "eks:DescribeCluster",
        "eks:ListClusters",
        "logs:Describe*",
        "logs:Get*",
        "logs:TestMetricFilter",
        "logs:FilterLogEvents",
        "iam:ListInstanceProfiles",
        "iam:ListRoles"
      ],
      "Resource":"*"
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:PassRole"
      ],
      "Resource":[
        "arn:aws:iam::*:role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/service-role/AWSBatchServiceRole",
        "arn:aws:iam::*:role/ecsInstanceRole",
        "arn:aws:iam::*:instance-profile/ecsInstanceRole",
        "arn:aws:iam::*:role/iaws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/aws-ec2-spot-fleet-role",
        "arn:aws:iam::*:role/AWSBatchJobRole*"
      ]
```

```
    },
    {
      "Effect":"Allow",
      "Action":[
        "iam:CreateServiceLinkedRole"
      ],
      "Resource":"arn:aws:iam::*:role/*Batch*",
      "Condition": {
        "StringEquals": {
          "iam:AWSServiceName": "batch.amazonaws.com"
          }
      }
    }
  ]
}
```

AWS Batch aggiornamenti alle politiche AWS gestite

Visualizza i dettagli sugli aggiornamenti delle 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
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 ID di dichiarazione, concedere AWS Batch autorizzazioni a `ec2:DescribeSpotFleetRequestHistory` `eautoscaling:DescribeScalingActivities` .	5 dicembre 2023

Modifica	Descrizione	Data
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 di prenotazione della capacità di Amazon EC2 gestiti da. AWS Resource Groups Per ulteriori informazioni, consulta Work with Capacity Reservation groups in 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 EC2 in modo da sostituire le istanze non integre.	6 dicembre 2021
BatchServiceRolePolicypolitica aggiornata	Aggiornato per aggiungere il supporto per il gruppo di posizionamento, 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

Modifica	Descrizione	Data
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

Accesso AWS Batch tramite un endpoint di interfaccia

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. AWS 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 AWS Secrets Manager .
  - 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 risorse EC2 richiedono che le istanze di container 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 il seguente nome di servizio:

```
com.amazonaws.region.batch
```

Per esempio:

```
com.amazonaws.us-east-2.batch
```

Nella `aws-cn` partizione, il formato è diverso:

```
cn.com.amazonaws.region.batch
```

Per esempio:

```
cn.com.amazonaws.cn-northwest-1.batch
```

Se abiliti il DNS privato per l'endpoint dell'interfaccia, puoi effettuare richieste API AWS Batch utilizzando il nome DNS regionale predefinito. Ad esempio, `batch.us-east-1.amazonaws.com`.

Per ulteriori 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 AWS Batch dal tuo VPC, collega una policy 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 .

È possibile scaricare report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta Scaricamento dei report in AWS Artifact .

La vostra responsabilità di conformità durante l'utilizzo Servizi AWS è determinata dalla sensibilità dei dati, dagli obiettivi di conformità dell'azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- Guide introduttive su sicurezza e conformità: queste guide all'implementazione illustrano considerazioni sull'architettura e forniscono passaggi per implementare ambienti di base incentrati sulla AWS sicurezza e la conformità.
- Progettazione per la sicurezza e la conformità HIPAA su Amazon Web Services: questo white paper descrive in che modo le aziende possono utilizzare AWS per creare applicazioni idonee all'HIPAA.

> **Note**
>
> Non Servizi AWS tutte sono idonee all'HIPAA. Per ulteriori informazioni, consulta la sezione Riferimenti sui servizi conformi ai requisiti HIPAA.

- AWS Risorse per la per la conformità: questa raccolta di cartelle di lavoro e guide potrebbe essere valida per il tuo settore e la tua località.
- AWS Guide alla conformità dei clienti: comprendi il modello di responsabilità condivisa attraverso la lente della conformità. Le guide riassumono le migliori pratiche per la protezione Servizi AWS e mappano le linee guida per i controlli di sicurezza su più framework (tra cui il National Institute of Standards and Technology (NIST), il Payment Card Industry Security Standards Council (PCI) e l'International Organization for Standardization (ISO)).
- Valutazione delle risorse con regole nella Guida per gli AWS Config sviluppatori: il AWS Config servizio valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida e alle normative del settore.
- AWS Security Hub— Ciò Servizio AWS fornisce una visione completa dello stato di sicurezza interno. AWS La Centrale di sicurezza utilizza i controlli di sicurezza per valutare le risorse AWS e verificare la conformità agli standard e alle best practice del settore della sicurezza. Per un elenco dei servizi e dei controlli supportati, consulta la pagina Documentazione di riferimento sui controlli della Centrale di sicurezza.
- Amazon GuardDuty: Servizio AWS rileva potenziali minacce ai tuoi carichi di lavoro Account AWS, ai contenitori e ai dati monitorando l'ambiente alla ricerca di attività sospette e dannose. GuardDuty può aiutarti a soddisfare vari requisiti di conformità, come lo standard PCI DSS, soddisfacendo i requisiti di rilevamento delle intrusioni imposti da determinati framework di conformità.
- AWS Audit Manager— Ciò Servizio AWS consente di verificare continuamente l' AWS utilizzo per semplificare la gestione del rischio e la conformità alle normative e agli standard di settore.

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.

Inoltre, le richieste devono essere firmate utilizzando un ID chiave di accesso e una chiave di accesso segreta associata a un principale IAM. O puoi utilizzare AWS Security Token Service (AWS STS) per generare credenziali di sicurezza temporanee per sottoscrivere le richieste.

Puoi 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 VPC specifici. In effetti, questo isola l'accesso alla rete a una determinata AWS Batch risorsa solo dal VPC specifico all'interno AWS della rete.

Tagging delle risorse AWS Batch

Per semplificare la gestione delle risorse AWS Batch, è possibile assegnare metadati personalizzati a ciascuna risorsa sotto forma di tag. Questo argomento descrive i tag e mostra come crearli.

Indice

Nozioni di base sui tag

Un tag è un'etichetta che assegni a una risorsa AWS. Ogni tag è composto da una chiave e da un valore opzionale, entrambi personalizzabili.

I tag consentono di categorizzare le risorse AWS per scopo, proprietario o ambiente. In presenza di un numero elevato di risorse, è possibile individuare rapidamente una risorsa specifica in base ai tag assegnati. Ad esempio, puoi definire un set di tag per i servizi AWS Batch per monitorare il proprietario di ogni servizio e il livello di stack. 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 per 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 la AWS Management Console, l'AWS CLI e l'API AWS Batch.

Se utilizzi AWS Identity and Access Management (IAM), puoi controllare quali utenti nel tuo account AWS dispongono dell'autorizzazione per creare, modificare o eliminare i tag.

Tagging delle risorse

È possibile etichettare ambienti di AWS Batch calcolo, processi, definizioni di lavoro, code di lavoro e politiche di pianificazione nuovi o esistenti.

Se utilizzi la console AWS Batch, puoi applicare tag alle nuove risorse quando vengono create o alle risorse esistenti utilizzando la scheda Tags (Tag) nella pagina della risorsa interessata in qualsiasi momento.

Se utilizzi l'API AWS Batch, l'AWS CLI o un SDK AWS, puoi applicare i tag alle nuove risorse mediante il parametro `tags` nell'operazione API rilevante oppure alle risorse esistenti mediante l'operazione API `TagResource`. 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.

Nella seguente tabella sono descritte le risorse AWS Batch a cui puoi associare i tag, nonché le risorse che possono essere associate a tag in fase di creazione.

Supporto del tagging per le risorse AWS Batch

Risorsa	Supporta tag	Supporta la propagazione di tag	Supporto del tagging in fase di creazione (API AWS Batch, AWS CLI, SDK AWS)
AWS Batchambienti 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	Supporto del tagging in fase di creazione (API AWS Batch, AWS CLI, SDK AWS)
		nell'operazione API. CreateComputeEnvironment	
AWS Batch processi	Sì	Sì	Sì
AWS Batchdefinizioni delle mansioni	Sì	No	Sì
Code di processi AWS Batch	Sì	No	Sì
AWS Batchpolitiche 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 lo schema di assegnazione dei tag viene utilizzato in più servizi e risorse AWS, tieni presente che altri servizi potrebbero prevedere limitazioni 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 `aws:`, `AWS:` o qualsiasi combinazione di maiuscole o minuscole di un tale prefisso per chiavi o valori poiché tali stringhe sono riservate per l'utilizzo esclusivo da parte di 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 tags-per-resource limite.

Utilizzo di tag tramite la console

Utilizzando la AWS Batch console, puoi gestire i tag associati agli ambienti di calcolo, ai lavori, alle definizioni dei processi e alle code di lavoro nuovi o esistenti.

Aggiunta di tag a una singola risorsa alla creazione

È possibile aggiungere tag agli ambienti di AWS Batch calcolo, ai processi, 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 i 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.

Utilizzo di tag tramite la CLI o l'API

Utilizza i seguenti comandi AWS CLI o operazioni API AWS Batch per aggiungere, aggiornare, elencare ed eliminare i tag per le risorse.

Supporto del tagging per le risorse AWS Batch

Attività	Azione API	AWS CLI	AWS Tools for Windows PowerShell
Aggiungere sovrascrivere uno o più tag.	TagResource	tag-resource	Aggiungi barra ResourceTag
Eliminare uno o più tag.	UntagResource	untag-resource	Rimuovi-bat ResourceTag
Elencazione dei tag associati a una risorsa	ListTagsForResource	list-tags-for-resource	Prendi pipistrello ResourceTag

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	New-BAT ComputeEnvironment
Crea una coda di lavoro	CreateJobQueue	create-job-queue	Nuova BAT JobQueue
Crea una politica di pianificazione	CreateSchedulingPolicy	create-scheduling-policy	Nuova BAT SchedulingPolicy
Registra una definizione di lavoro	RegisterJobDefinition	register-job-definition	Registrati - BAT JobDefinition
Invio di un processo	SubmitJob	invia un lavoro	Invia - batjob

Quote di servizio di AWS Batch

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 Ambiente di elaborazione.	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

A seconda dell'utilizzoAWS Batch, potrebbero essere applicate quote aggiuntive. Per ulteriori informazioni sulle quote Amazon EC2, 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

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 viene eseguito sull'infrastruttura 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 alle istanze EC2 nella Amazon EC2 User Guide

Indice

    - Il profilo dell'istanza non esiste
    - Namespace non valido Kubernetes
    - Ambiente di elaborazione eliminato
    - I nodi non entrano a far parte del cluster Amazon EKS
- AWS Batch su Amazon EKS il lavoro è bloccato RUNNABLE
- Verifica che aws-auth ConfigMap sia configurato correttamente
- Le autorizzazioni o le associazioni RBAC non sono configurate correttamente

AWS Batch

INVALIDambiente 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.

Nome ruolo o ARN errati

La causa più comune per cui un ambiente di calcolo entra in uno `INVALID` stato è che il ruolo di AWS Batch servizio o il ruolo Amazon EC2 Spot Fleet ha un nome o un Amazon Resource Name (ARN) errati. Questo è più comune negli ambienti di elaborazione creati utilizzando o gli SDK. AWS CLI AWS Quando crei un ambiente di elaborazione in AWS Management Console, 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. Quando usi gli SDK AWS CLI o gli AWS SDK, specifichi solo il nome di un ruolo IAM, 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 specifichi solo il nome di un ruolo IAM quando usi AWS CLI o gli AWS SDK, AWS Batch presuppone che l'ARN non utilizzi il prefisso del percorso. `aws-service-role` 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 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.
6. Seleziona Save (Salva) per aggiornare l'ambiente di calcolo.

Lavori bloccati in uno status **`RUNNABLE`**

Supponiamo che l'ambiente di elaborazione contenga risorse di elaborazione, ma che i lavori non vadano oltre tale 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 evento di coda lavori bloccati da Amazon CloudWatch Events con il motivo. Lo stesso motivo viene aggiornato anche nel `statusReason` campo come parte delle chiamate API`ListJobs`. `DescribeJobs`

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 istanze Amazon EC2 che presentano errori di capacità insufficiente. L'annullamento del processo, manualmente o impostando il `jobStateTimeLimitActions` parametro on`statusReason`, consente al lavoro successivo di passare in testa alla coda.
   - **`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 `AWSServiceRoleForBatch`collegato al servizio (SLR) o la policy `AWSBatchServiceRolePolicy`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 calcolo Amazon EC2 ad alta priorità collegato presenta `insufficient capacity` errori, blocca la coda anche se in un ambiente di calcolo con priorità inferiore si verifica 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 dei ruoli di servizio con le autorizzazioni dei ruoli di servizio AWS Batch gestiti e colma eventuali lacune.

   È consigliabile utilizzare la AWS Batch reflex digitale per ambienti di elaborazione per evitare errori simili.

   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.`
   - **`reason`**usato per**`jobStateTimeLimitActions`**: `MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`
   - **`statusReason`**messaggio dopo l'annullamento del lavoro: `Canceled by JobStateTimeLimit action due to reason: MISCONFIGURATION:SERVICE_ROLE_PERMISSIONS`

5. Motivo: tutti gli ambienti di elaborazione non sono validi. Per ulteriori informazioni, consulta Ambiente di `INVALID`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.`

6. 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 con il 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, consulta Specifiche AMI delle risorse di calcolo e Creazione di una risorsa di calcolo AMI.

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 Risorsa di calcoloGestione della memoria.

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 istanze Amazon EC2 raggiunto

Il numero di istanze Amazon EC2 in cui il tuo account può avviare Regione AWS è determinato dalla quota di istanze EC2. Alcuni tipi di istanze hanno anche una quota. per-instance-type Per ulteriori informazioni sulla quota di istanze Amazon EC2 del tuo account e su 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 container Amazon ECS è installato per impostazione predefinita sulle AMI ottimizzate per Amazon ECS. 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 elaborazione 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 in 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 in Amazon EC2 Spot Fleet

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli i ruoli e scegli il tuo ruolo nella flotta Spot di Amazon EC2.
3. Scegli Collega policy.
4. Seleziona AmazonEC2 SpotFleet TaggingRole e scegli Allega policy.
5. Scegli nuovamente il tuo ruolo in Amazon EC2 Spot Fleet per rimuovere la politica precedente.
6. Seleziona la x a destra della politica del SpotFleetruolo di AmazonEC2 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 Spot EC2, ma che il ruolo Spot utilizzato non includa la policy gestita di AmazonEC2. SpotFleet TaggingRole 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

- Allega la politica SpotFleet TaggingRole gestita di AmazonEC2 al tuo ruolo di Spot Fleet nel AWS Management Console
- Associa la politica SpotFleet TaggingRole gestita di AmazonEC2 al tuo ruolo nella flotta Spot con il AWS CLI

Allega la politica SpotFleet TaggingRole gestita di AmazonEC2 al tuo ruolo di Spot Fleet nel AWS Management Console

Per applicare l'attuale policy gestita da IAM al tuo ruolo in Amazon EC2 Spot Fleet

1. Aprire la console IAM all'indirizzo https://console.aws.amazon.com/iam/.
2. Scegli i ruoli e scegli il tuo ruolo nella flotta Spot di Amazon EC2.

3. Scegli Collega policy.
4. Seleziona AmazonEC2 SpotFleet TaggingRole e scegli Allega policy.
5. Scegli nuovamente il tuo ruolo in Amazon EC2 Spot Fleet per rimuovere la politica precedente.
6. Seleziona la x a destra della politica del SpotFleetruolo di AmazonEC2 e scegli Detach.

Associa la politica SpotFleet TaggingRole gestita di AmazonEC2 al tuo ruolo nella flotta Spot con il AWS CLI

I comandi di esempio presuppongono che il ruolo Spot Fleet di Amazon EC2 sia denominato *SpotFleetAmazonEC2* Role. Se il tuo ruolo utilizza un nome diverso, modifica i comandi in modo che corrispondano.

Per allegare la politica SpotFleet TaggingRole gestita da AmazonEC2 al tuo ruolo Spot Fleet

1. Per collegare la politica IAM SpotFleet TaggingRole gestita da AmazonEC2 al tuo ruolo di *SpotFleetruolo AmazonEC2*, esegui il seguente comando 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 IAM gestita dal ruolo AmazonEC2 dal SpotFleet ruolo AmazonEC2 Role, esegui il comando seguente *utilizzando SpotFleet*. 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 come segue:

```
"containerProperties": {
   "resourceRequirements": [
      {
         "type": "MEMORY",
         "value": "4096"
      },
      {
         "type": "VCPU",
         "value": "2"
      }
   ],
}
```

Messaggio di errore quando si aggiorna l'**`desiredvCpus`**impostazione

Viene visualizzato il seguente messaggio di errore quando si utilizza l' AWS Batch API per aggiornare l'impostazione `desiredvCpus` vCPUS () desiderata.

`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 valore correnteDesiredvCpus. 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

INVALIDambiente 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:

- `1.29`
- `1.28`
- `1.27`
- `1.26`
- `1.25`
- `1.24`
- `1.23`

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 CreateCompute Ambiente.

- 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 Verifica 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**

Un `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 `kubectl` riapplicare il manifesto. Per ulteriori informazioni, consulta Fase 1: Preparazione del 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.

Verifica 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 `kubectl` riapplicare il manifesto. Per ulteriori informazioni, consulta Fase 1: Preparazione del 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 allo spazio dei nomi: `aws-batch` Kubernetes 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]
```

```
$ 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 1: Preparazione del cluster Amazon EKS per AWS Batch.

Best practice 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 Batchi 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 Batchesegue 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, raggruppa binpack le 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. Valuta la possibilità di raggruppare le attività in modo che le attività vengano eseguite da 3 a 5 minuti ciascuna. Dopo aver completato binpack i lavori, ripercorri i gruppi di attività all'interno del tuo AWS Batch lavoro.
- I lavori che devono essere eseguiti immediatamente AWS Batch possono essere elaborati rapidamente. Tuttavia, AWS Batch è uno strumento di pianificazione e ottimizza i costi, le

prestazioni, la priorità dei lavori e la produttività. AWS Batchl'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 50mila o più vCPU, considera la seguente lista di controllo.

> **Note**
>
> Se hai intenzione di eseguire un carico di lavoro di grandi dimensioni su un milione o più di vCPU o hai bisogno di assistenza per l'esecuzione su larga scala, contatta il tuo team. AWS

- Controlla le tue quote Amazon EC2: controlla le tue quote Amazon EC2 (note anche come limiti) nel pannello Service Quotas del. AWS Management Console Se necessario, richiedi un aumento della quota per il numero massimo di istanze Amazon EC2. 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 volume GP2 o GP3 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 i punti deboli: per un lavoro eseguito su un milione o più di VCPUS, è consigliabile iniziare con un valore inferiore e aumentare gradualmente, in modo da poter identificare tempestivamente i punti deboli. Ad esempio, inizia eseguendo 50 mila vCPU. Quindi, aumenta il conteggio a 200 mila vCPU, quindi a 500 mila vCPU e così via. In altre parole, continuate ad aumentare gradualmente il numero di vCPU fino a raggiungere il numero desiderato di vCPU.

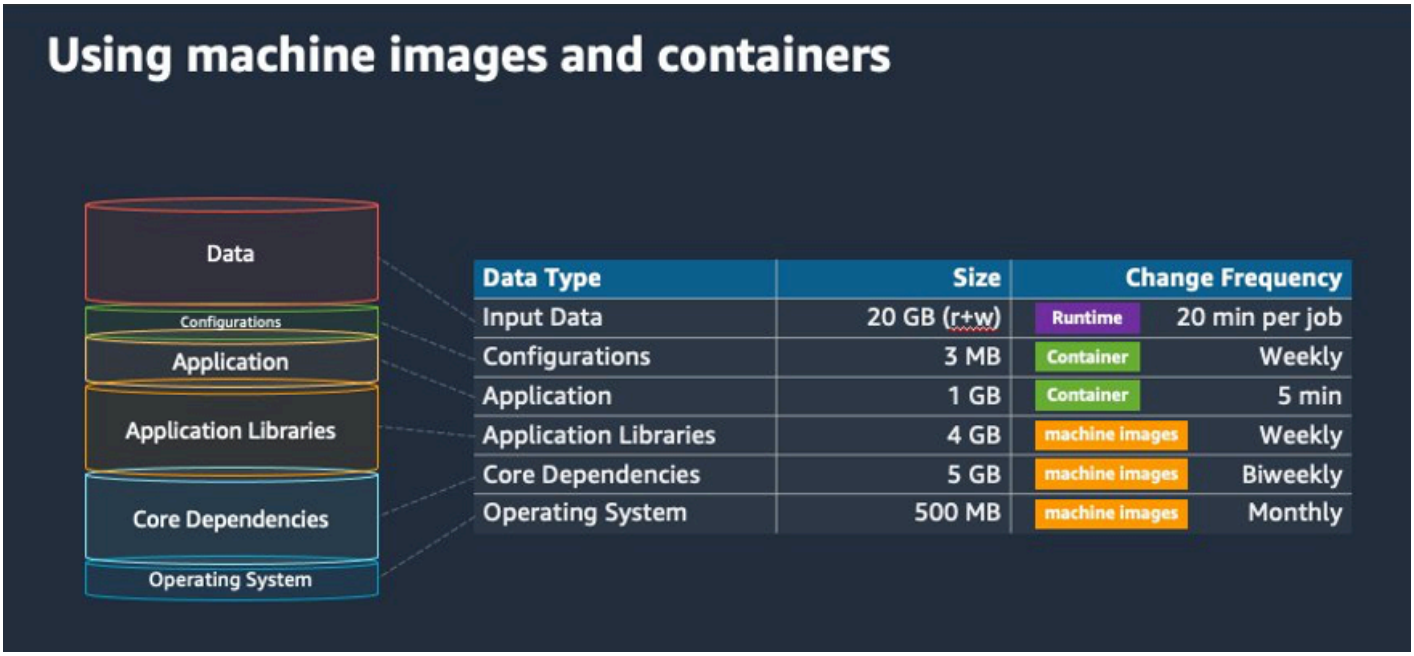
- Monitora per identificare tempestivamente potenziali problemi: per evitare potenziali interruzioni e problemi durante l'esecuzione su larga scala, assicurati di monitorare sia l'applicazione che l'architettura. Potrebbero verificarsi interruzioni anche in caso di scalabilità da 1.000 a 5.000 vCPU. 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 contenitori e AMI

Le dimensioni e la struttura dei contenitori sono importanti per la prima serie di processi 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 con oltre un milione di vCPU.



Scegli la risorsa giusta per l'ambiente di elaborazione

AWS Fargaterichiede 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 lavori sono 16 vCPU o meno, nessuna GPU e 120 GiB di memoria o meno.

Per ulteriori informazioni, consulta Quando usare Fargate.

Se hai i seguenti requisiti, ti consigliamo di utilizzare le istanze Amazon EC2:

- 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 essere fornite, come GPU, più memoria, un'AMI personalizzata o Amazon Elastic Fabric Adapter.
- È necessario un elevato livello di velocità effettiva o di concorrenza.

- È necessario personalizzare l'AMI, il modello di avvio di Amazon EC2 o l'accesso 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 su richiesta o Amazon EC2 Spot

La maggior parte dei AWS Batch clienti utilizza le istanze Spot di Amazon EC2 grazie ai risparmi rispetto alle istanze On-Demand. Tuttavia, se il 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, usa le istanze On-Demand di Amazon EC2:

- 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, usa le istanze Spot di Amazon EC2:

- 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 calcolo in esecuzione su istanze Spot di Amazon EC2. 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 Amazon Simple Notification Service per eventi Job non riusciti, consulta le best practice per la gestione delle interruzioni delle istanze Spot di Amazon EC2 e Manage with Step AWS Batch 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 Spot di Amazon EC2 e ridurre il tasso di interruzione.

Utilizza le best practice Spot di Amazon EC2 per AWS Batch

Scegliendo 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 for Amazon EC2 Spot.

Per ottimizzare il flusso di lavoro e ridurre i costi, prendi in considerazione le seguenti best practice Spot di Amazon EC2 per: AWS Batch

- Scegli la strategia di **`SPOT_CAPACITY_OPTIMIZED`** allocazione: AWS Batch sceglie le istanze Amazon EC2 dai pool di capacità Spot di Amazon EC2 più profondi. Se sei preoccupato per le interruzioni, questa è la scelta giusta. Per ulteriori informazioni, consulta Strategie di allocazione.
- 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`,, `c5nc5d`, `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 Spot di Amazon EC2 per evitare interruzioni. Se dividi o controlli i tuoi lavori in parti più piccole che durano 30 minuti o meno, puoi ridurre in modo significativo 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 Spot di Amazon EC2, consulta Spot Interruption Dashboard.

  InfattiAWS Batch, se si imposta 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 inAWS 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`

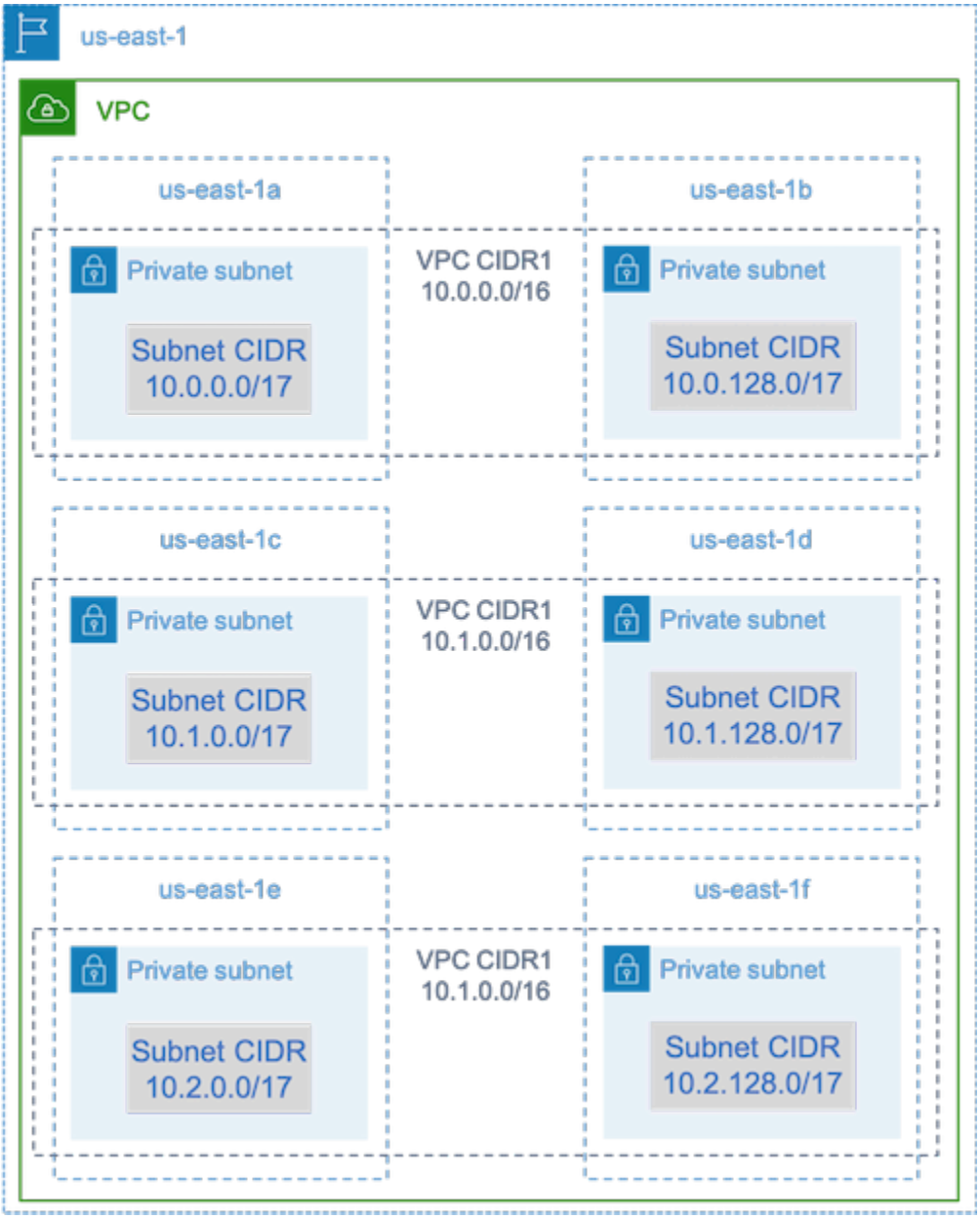
```
"retryStrategy": {
    "attempts": 5,
    "evaluateOnExit":
    [{
        "onStatusReason" :"Host EC2*",
        "action": "RETRY"
    },{
      "onReason" : "*"
        "action": "EXIT"
    }]
}
```

- 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 Spot di Amazon EC2 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 status `RUNNABLE`. Per informazioni sulla risoluzione dei problemi degli ambienti di calcolo in uno `INVALID` stato, consulta`INVALID`ambiente di calcolo.

- Controlla le quote di vCPU Spot di Amazon EC2: verifica che le quote di servizio attuali soddisfino i requisiti del lavoro. Ad esempio, si supponga che la quota di servizio corrente sia di 256 vCPU e che il job richieda 10.000 vCPU. 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 servizio di Amazon EC2 e Come posso aumentare la quota di servizio delle mie risorse Amazon EC2? .
- 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 "" inAWS 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 (CIDR) 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, è possibile creare un VPC con più CIDR `10.x.0.0/16` e una sottorete 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 con Amazon EC2: se vedi le tue istanze nella console Amazon EC2, ma nessuna istanza del contenitore Amazon Elastic Container Service nel cluster Amazon ECS, l'agente Amazon ECS potrebbe non essere installato su un'Amazon Machine Image (AMI). Inoltre, l'agente Amazon ECS, i dati Amazon EC2 nell'AMI o il modello di avvio potrebbero non essere configurati correttamente. Per isolare la causa principale, crea un'istanza Amazon EC2 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. AMI per risorse di calcolo

- Esamina la AWS dashboard: esamina la AWS dashboard per verificare lo stato del processo previsto 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 console Amazon EC2, 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, i dati Amazon EC2 nell'AMI o il modello di avvio potrebbero non essere configurati correttamente. Per isolare la causa principale, crea un'istanza Amazon EC2 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. AMI per risorse di calcolo
- 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 AWS Support 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 job e istanze. AWS Batch Per ulteriori informazioni, consulta AWS BatchRuntime Monitoring Dashboards Solution.

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
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 supportano l'inclusione della versione 1.29.	29 febbraio 2024
Nuovi tentativi di lavoro automatizzati	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		5 dicembre 2023

	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 ID di dichiarazione, concedere AWS Batch autorizzazioni a `ec2:DescribeSpotFleetRequestHistory` e. `autoscaling:DescribeScalingActivities`	
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. AWS 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 equa della condivisione	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

Supporto per Amazon Linux 2	AWS Batch aggiunge il supporto per la selezione automatica delle AMI Amazon Linux 2 nell'ambiente di calcolo utilizzando i parametri di configurazione EC2.	24 novembre 2020
Strategia di riprova migliorata	AWS Batch migliora la strategia di nuovi tentativi di lavoro. Ora i lavori possono essere ritentati o interromp ere ulteriori tentativi associand o lo o `ExitCode` il `Reason` lavoro a `StatusReason` degli schemi.	20 ottobre 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 la scelta dei 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, è possibile specificare il numero di GPU richieste da ciascun 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 istanze Amazon EC2.	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 Spot di Amazon EC2 lanciate all'inter no 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 automatizzati	AWS Batch aggiunge il supporto per i nuovi tentativi di lavoro. Con questo aggiornamento, 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.